IMPROVING THE PERFORMANCE OF THE HOPFIELD NETWORK FOR SOLVING OPTIMIZATION PROBLEMS

by

Xinchuan Zeng

A thesis submitted to the faculty of

Brigham Young University

in partial fulfillment of the requirements for the degree of

Master of Science

Department of Computer Science

Brigham Young University

December 1998

BRIGHAM YOUNG UNIVERSITY

GRADUATE COMMITTEE APPROVAL

of the thesis submitted by

Xinchuan Zeng

This thesis has been read by each member of the following graduate committee and by majority vote has been found to be satisfactory.

Date	Tony R. Martinez, Committee Chair
Date	Theodore A. Norman, Committee Member
Date	Quinn O. Snell, Committee Member
Date	Scott N. Woodfield, Graduate Coordinator

ABSTRACT

IMPROVING THE PERFORMANCE OF THE HOPFIELD NETWORK FOR SOLVING OPTIMIZATION PROBLEMS

Xinchuan Zeng

Department of Computer Science

Master of Science

Three approaches that significantly improve the performance of the *Hopfield network* for solving optimization problems have been developed and evaluated in this work. The first approach uses a new activation function to reduce the effects of noise in the network. The second approach includes a parameter to control the relaxation process so that the network can be relaxed at a proper rate. The third approach applies a new relaxation procedure to achieve a smoother relaxation process. These approaches have been tested through a large number of simulations based on 200 randomly generated city distributions of the 10-city *traveling salesman problem*. The results show that all the three approaches are capable of improving the performance of the network. Among them, the new relaxation procedure achieved the highest improvement: an increase of 185.7% in the percentage of valid tours and a decrease of 33.1% in the error rate. The performance of combining the new activation function with the other two approaches has also been evaluated, and the results reveal that the combining can further improve

the performance. Combining the new activation function with the new relaxation procedure resulted in a 245.7% increase in the percentage of valid tours and a 64.1% decrease in the error rate.

CONTENTS

1. Introduction

Hopfield and Tank [1] proposed an approach of using a neural network to find a suboptimal solution of the traveling salesman problem (*TSP*). In their approach, a *TSP* city distribution is represented by an *energy function* including the cost and constraint terms that reflect the objective of a solution. The objective of the constraint term is to find a valid tour, which requires that each city must be visited once and only once. The objective of the cost term is to find the shortest valid tour.

The energy function can be implemented by a neural network. For an N-city *TSP* problem, the network consists of $N \times N$ neurons and the links connecting the neurons. The weights of the links are encoded to represent the cost and the constraints for the *TSP* city distribution to be solved.

To find a solution, the network starts from a random initial state and then relaxes until reaching an equilibrium state. During the relaxation, each neuron updates its input value based on information received from other neurons. Hopfield and Tank showed that a neural network configured with symmetric weights always converges to a local minimum which may represent a valid solution. Their simulation on a 10-city *TSP* city distribution showed that the neural network was able to obtain a high percentage of valid tours with high quality (short tours). This network is commonly referred to as a *Hopfield network*.

Since Hopfield and Tank's work [1], there has been growing interest in the

Hopfield network because of its advantages over other approaches for solving optimization problems. The advantages include massive parallelism, convenient hardware implementation of the neural network architecture, and a common approach for solving various optimization problems. Much research has focused on analyzing and improving the original model in order to obtain a higher percentage of valid solutions and solutions with better quality.

The work by Wilson and Pawley [2] showed that the Hopfield network often failed to converge to valid solutions. When it converged, the obtained solution was often far from the optimal solution. Since their report, various modifications have been proposed to improve the convergence of the Hopfield network. Brandt et al. [3] and Aiyer et al. [4] showed that the convergence of the Hopfield network could be improved by modifying the energy function. Other modified energy functions [5, 6, 7] were also attempted to improve the performance of the Hopfield network. The modified energy functions have a higher degree of constraints for the network to relax to valid solutions. Protzel et al. [8] studied Brandt et al.'s formulation with different parameters and were able to improve the performance. Li combined the Hopfield network with the "augmented Lagrange multipliers" algorithm from optimization theory [9]. Although both the convergence and the quality of the solutions can be improved by this approach, many more iterations are required to converge to a solution. Catania et al. applied a fuzzy approach to automatically tune the parameters in the Hopfield network [10]. They reported that the automatic tuning of the parameters performed better than heuristic and manual tuning. Although the above approaches have demonstrated improvement, most of them were empirically tested only on a single or a small number of TSP city distributions, and general performances on a large number of city distributions were not reported.

The Hopfield network has also been applied to solve various practical optimization problems, such as routing in packet-switched communication network [11], satellite broadcasting scheduling [12], channel allocation for mobile communication [13], polygonal approximation [14], stereo correspondence [15], and medical image segmentation [16]. In many of these applications, the Hopfield network procedure can obtain better or comparable results as compared with traditional procedures in each individual field.

Despite the improvement of the performance of the Hopfield network over the past decade, this model still has some basic problems [17, 18]. One of the problems is that the performance of the Hopfield network is inconsistent. The performance is good for some city distributions of *TSP*, but poor for other city distributions with the same size. The performance is usually better for city distributions with simple topology, but poor for those with complex topology, in which case there are large number of similar competent fixed points and the solutions are often trapped in poor local minima. Another problem is that the performance is sensitive to the choice of the parameters in the energy function: a different parameter setting can lead to a significant difference in the performance.

The work reported in this paper focuses on modifying the activation function and the relaxation process to improve the Hopfield model, in contrast to the previous work which mainly emphasizes modifying the energy function. More specifically, we have proposed the following three approaches to improve the performance of the Hopfield network: (1) using a *new activation function*, (2) including a *relaxation controlling parameter*, and (3) applying a *new relaxation procedure*.

The first approach [19] focuses on improving the activation function, which maps the input of a neuron to its output (activation). The original activation function in the Hopfield network is a *sigmoid* function. This function however has a drawback: it is highly sensitive in the noise-dominant range and thus would relatively weaken useful signals between neurons. To reduce this effect, we proposed a noise-robust activation function. The new function uses an adjustable parameter to control the sensitivity threshold so that the function is much less sensitive to noise while remaining responsive to useful signals.

The second approach [20] includes a relaxation controlling parameter to control the relaxation process, which is an important factor for performance. The goal of introducing a relaxation controlling parameter is to adjust the relaxation pace so that the network is relaxed at a more optimal rate.

The third approach [21] applies a new relaxation procedure into the network dynamics. Unlike the original relaxation procedure, which first updates the input of a neuron and then updates its output through the activation function, the new relaxation procedure updates the output directly without using the update of the input as an intermediate step. The analysis shows that the new relaxation procedure is able to support a smoother relaxation.

The above proposed approaches have been tested on a large number of simulations based on 200 randomly generated city distributions of the 10-city *TSP*. The results demonstrate that all of them are capable of significantly improving the performance of the network. The best improvement was achieved by the new relaxation procedure, increasing the percentage of valid tours by 185.7% and decreasing the error rate by 33.1% as compared to the original Hopfield network.

This paper also evaluates the performance of combining the new activation function with the other two approaches, and the results reveal that combining can further improve the performance. The combining of the new activation function with the new relaxation procedure leads to an increase of 245.7% in the percentage of valid tours and a decrease of 64.1% in the error rate.

The rest of paper is organized as follows. Section 2 briefly reviews the basic architecture and procedure of the Hopfield network for solving optimization problems. Section 3, 4 and 5 present the new activation function, the relaxation controlling parameter and the new relaxation procedure respectively, along with simulation results related to these approaches. Section 6 and 7 describe the method and the simulation results for combining the new activation function with the relaxation controlling parameter and the new relaxation procedure respectively. Finally, Section 8 draws conclusions from this work and proposes future research.

2. Basics of the Hopfield Network

The Hopfield network [1] consists of a set of neurons and the connecting links between neurons. To solve an N-city *TSP*, for instance, the network includes $N \times N$ fully connected neurons. Fig. 1 gives an example of the network architecture for solving a 7×7 *TSP*.

In Fig. 1, each row represents a city and each column represents the order of a city in the tour. The connecting strengths (weights) between neurons are determined according to the constraints and the cost in an energy function. *TSP* includes the following constraints for a valid tour: each city must be visited exactly once (each row has exactly one "on" neuron), and only one city can be visited at any time (each column has exactly one "on" neuron). The objective of the cost function is to find a valid tour with minimum tour length.

The above constraints and the cost can be represented by an energy function, which is used to determine the connecting weights between neurons. Hopfield's original energy function for an N-city *TSP* is given by [1]:

$$E = \frac{A}{2}\sum_{X=1}^{N}\sum_{i=1}^{N}\sum_{j=1,j\neq i}^{N} V_{Xi}V_{Xj}$$
$$+\frac{B}{2}\sum_{i=1}^{N}\sum_{X=1}^{N}\sum_{Y=1,Y\neq X}^{N} V_{Xi}V_{Yi} + \frac{C}{2}(\sum_{X=1}^{N}\sum_{i=1}^{N} V_{Xi} - N_0)^2$$
$$+\frac{D}{2}\sum_{X=1}^{N}\sum_{i=1}^{N}\sum_{Y=1,Y\neq X}^{N} d_{XY}V_{Xi}(V_{Y,i+1}+V_{Y,i-1}) \tag{1}$$

where X, Y are row indices, and i, j are column indices, V_{Xi} is the activation for neuron (X,i), and d_{XY} is the distance between cities X and Y. The first three terms enforce the constraints for a valid tour, and the last term represents the

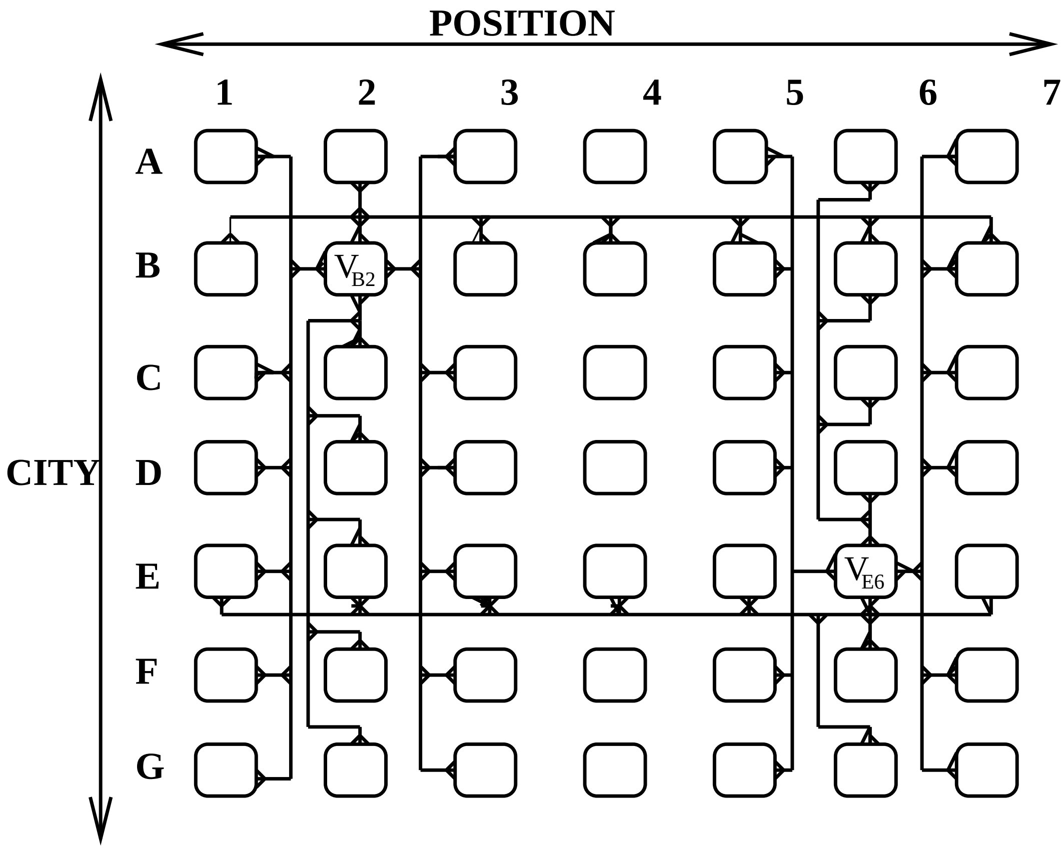

Figure 1: The architecture of the Hopfield network for solving a 7×7 *TSP*

cost function for obtaining a short tour. The value of each parameter (A, B, C, and D) measures the importance of the corresponding term.

Each neuron (X, i) has an input value U_{Xi} and an activation (output) value V_{Xi}. The connecting weight between neuron (X, i) and (Y, j) is set according to:

$$W_{Xi,Yj} = -A\delta_{XY}(1-\delta_{ij}) - B\delta_{ij}(1-\delta_{XY})$$

$$-C - Dd_{XY}(\delta_{j,i+1} + \delta_{j,i-1}) \tag{2}$$

Each neuron (X, i) is also connected to an external input current:

$$I_{Xi} = CN_0 \tag{3}$$

Before the relaxation of the network, the initial value of each U_{Xi} is set to be a constant value (determined by the condition: $\sum_{X=1}^{N}\sum_{i=1}^{N} V_{Xi} = N$) and is then perturbed with a small random noise value (to break the symmetry of the network).

During relaxation, each neuron updates its input and activation value based on the weighted activations of other neurons and its own value. Specifically, the value $U_{Xi}^{(n+1)}$ at iteration step *(n+1)* is given by:

$$U_{Xi}^{(n+1)} = U_{Xi}^{(n)} + \Delta U_{Xi} \tag{4}$$

where $U_{Xi}^{(n)}$ is the input value at iteration step *(n)*. The value of ΔU_{Xi} is given by the following equation:

$$\Delta U_{Xi} = (-\frac{U_{Xi}}{\tau} + \sum_{Y=1}^{N}\sum_{j=1}^{N} W_{Xi,Yj}V_{Yj} + I_{Xi})^{(n)}\Delta t \tag{5}$$

where $\tau(=RC)$ is the time constant of an RC circuit and was set to be 1.0 by Hopfield and Tank [1].

The activation $V_{Xi}^{(n+1)}$ at iteration step *(n+1)* is then determined by $U_{Xi}^{(n+1)}$ through an *activation (output) function.* In the Hopfield network, the activation function is the *sigmoid* function:

$$V_{Xi}^{(n+1)} = \frac{1}{2}(1 + tanh(\frac{U_{Xi}^{(n+1)}}{u_0})) \tag{6}$$

where u_0 is the amplification parameter that reflects the steepness of the activation function.

Hopfield and Tank [1] showed that the network is guaranteed to converge to a local minimum in the case of symmetric $(W_{Xi,Yj} = W_{Yj,Xi})$ connecting weights.

3. A New Activation Function

In this section, we first describe the original activation function and one of its drawbacks, and then present a new activation function [19] that reduces this drawback. Then we display experimental results demonstrating the improvement in performance resulting from using the new activation function.

3.1 Original Activation Function

The original activation (*sigmoid*) function in the Hopfield network is given by Eq. (6). The shape of this function (with $u_0 = 0.02$) is plotted as the curve "A" in Fig. 2. One of main features of this function is that it is sensitive in the range where the input U_{Xi} is small (since its steepest part is around $U_{Xi} = 0$).

However, this feature is undesirable since it puts much emphasis on minor noise perturbation[1] in addition to the signals related to the cost and the constraints encoded in the network. Consequently, this activation function would affect the performance of the network due to the significant amount of irrelevant noise involved in the relaxation process.

This disadvantage of the *sigmoid* function may partially account for the observations in the experiment by Wilson and Pawley [2]. They tested the original Hopfield network on 10 sets of randomly generated 10-city *TSP*. For each city

[1]Adding small noise into neurons at the initial stage is important since without noise, all neurons would maintain the initial values and thus the network would not converge to a valid equilibrium state.

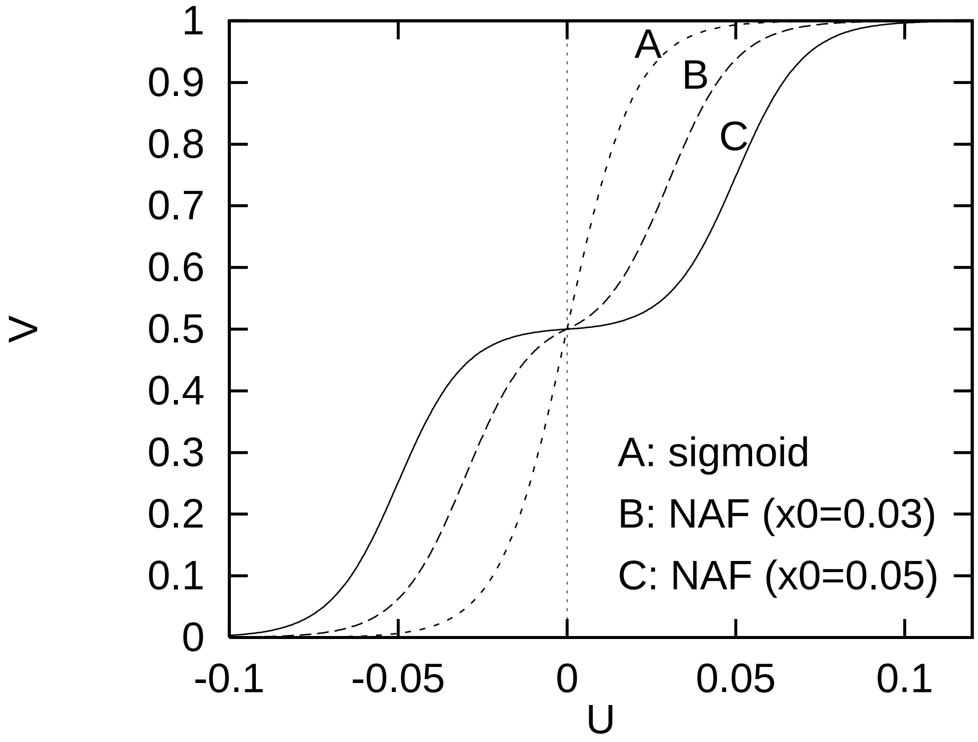

Figure 2: Comparison between the *sigmoid* function and the new activation function (NAF) with different thresholds (x_0).

distribution, 50 runs with different initial noise were simulated in the experiment. They reported that 8% of the runs converged to valid tours, 48% of the runs froze into invalid tours, and 44% of the runs did not converge in 1000 iterations. Different sets of initial noise (with other parameters fixed) yielded solutions with differing quality or invalid solutions. This suggests that the performance of the network may be overly sensitive to the initial noise, which may be caused in part by using the *sigmoid* as the activation function.

3.2 New Activation Function

To reduce the drawback of the original activation function and improve the performance of the network, we propose the following activation function:

$$V_{Xi} = \frac{0.5(1 + tanh(\frac{U_{Xi}+x_0}{u_0}))}{1 + tanh(\frac{x_0}{u_0})} \qquad (U_{Xi} < 0)$$

$$V_{Xi} = \frac{tanh(\frac{x_0}{u_0}) + 0.5(1 + tanh(\frac{U_{Xi}-x_0}{u_0}))}{1 + tanh(\frac{x_0}{u_0})} \qquad (U_{Xi} \geq 0) \tag{7}$$

where x_0 represents the threshold for V_{Xi} to become steep, and u_0 measures the steepness of the activation function.

The shapes of the new activation function with differing values of x_0 are shown in Fig. 2 (u_0 is fixed at 0.02). They becomes steep only when the absolute value of V_{Xi} is larger than a threshold x_0. Thus random noise smaller than the threshold will not be amplified, while the useful signals between neurons (usually having larger magnitude than noise) will still be amplified. Only when substantial positive or negative evidence is summed into a neuron, will the neuron make a significant change in its activation function. This mechanism has the capability of reducing the effects of noise in the formation of a tour.

3.3 Simulation Results

We have evaluated the performance of the new activation function through simulations, and compared it with that of the *sigmoid* function. The evaluation in our experiment was based on 200 randomly generated 10-city *TSP* city distributions, including wide varieties of topologies.

Many previous studies used only one city distribution (the one used by Hopfield and Tank in their original paper [1]) or a small number of city distributions (for example, 10 city distributions in [2]) in their simulations. This may result in an unreliable conclusion when comparing two algorithms. The reason is that the performance of an algorithm often depends on the topology in a city distribution, and different algorithms may favor different types of topologies. Using a large number of city distributions can reduce this effect and allow a better evaluation of the algorithms.

In the simulation, 100 runs are conducted for each of the 200 city distributions. For each of 100 runs, different random noise is added to the initial input values of the neurons.

For a fixed set of parameters (dt, u_0, etc), the quantities to be evaluated (error rate and percentage of valid tours) are first averaged over 100 runs for each city distribution, and then averaged over the entire 200 city distributions.

Thus 20,000 runs are needed to obtain each data point that is shown in the figures of the simulation results.

We experimented with different sets of 20,000 runs for a fixed set of parameters. The result shows that the estimated quantities are fairly stable. Their values vary within a range of about 1% among different sets of 20,000 runs. This demonstrates that the number of runs in our simulation is large enough to make a reasonable estimation of the evaluated quantities.

The original energy function of the Hopfield network is used in the simulation, and the parameters in the energy function are those used by Hopfield and Tank [1]:

$$A = B = D = 500, \quad C = 200, \quad N_0 = 15 \tag{8}$$

The value of dt in Eq. (5) is set to be 10^{-5}, and the value of u_0 is fixed at 0.02 for both Eq. (6) and Eq. (7).

The fraction of random noise in the initial values of neurons is set to be 0.001 in the simulation. We tried several different values for the fraction in the range from 0.0001 to 0.01. The performance of the network is only slightly sensitive to this parameter, and the choice of 0.001 works slightly better than others.

Fig. 3 shows the percentage of valid tours using the new activation function for different thresholds (x_0), compared to that using the *sigmoid* function. Each data point in the figure is the weighted average percentage of valid tours over

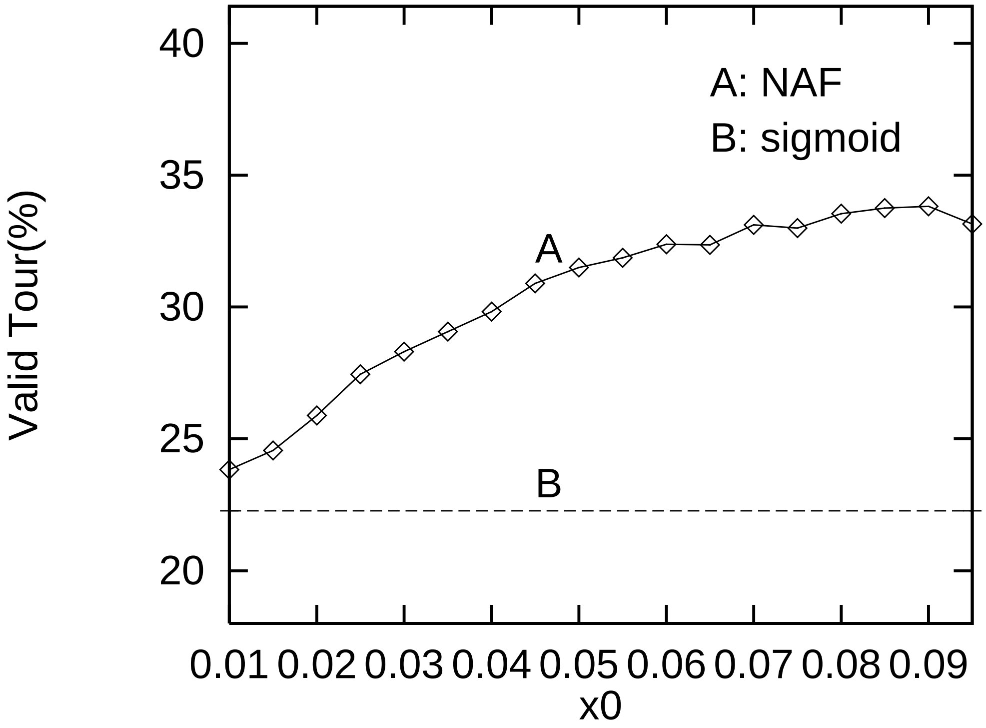

Figure 3: The percentage of valid tours using the new activation function (NAF) for different thresholds (x_0), compared to that using the *sigmoid* function.

$N_{CityDist}$ $(= 200)$ city distributions:

$$Valid = \frac{\sum_{i=1}^{N_{CityDist}} Valid_i}{N_{CityDist}} \tag{9}$$

where $Valid_i$ is the percentage of valid tours for city distribution i and is defined in the following.

For each city distribution i, there are a total of $N_{total,i}$ $(= 100)$ runs with different initial input values. The maximum number of iterations allowed for each run is set to be 1000. If a valid tour can not be reached within 1000 iterations, the network will stop and the tour is counted as invalid.

If $N_{valid,i}$ is the number of valid tours among the total of $N_{total,i}$ runs, then $Valid_i$ is defined by:

$$Valid_i = \frac{N_{valid,i}}{N_{total,i}} \tag{10}$$

Fig. 3 shows that the network using the new activation function is capable of finding more valid tours than that using the *sigmoid* function (for a wide range of the threshold x_0).

Another observation from Fig. 3 is that the improvement depends on the threshold x_0. This dependence can be explained as follows in terms of the effects of noise and signals between neurons.

Recall from Fig. 2 that a different threshold x_0 leads to a different shape of the curve for the new activation function (e.g., curve "B" and "C"). For a smaller threshold x_0, a higher fraction of noise is amplified, which therefore leads to a lower percentage of valid tours. This explains why the percentage of valid tours in Fig. 3 drops when x_0 moves towards 0.

In contrast, when the threshold x_0 increases, a lower fraction of noise but a high fraction of useful signals between neurons are amplified, which explains the trend of the increased improvement in the figure when x_0 moves from 0.01 to 0.07.

However, when x_0 further increases (> 0.07), the percentage of valid tours starts dropping slightly. The reason is that too high a threshold blocks not only noise, but also some useful signals between neurons.

Note that the results are fairly stable across a relatively broad range of x_0, and thus the network with the new activation function is robust with respect to this parameter.

Fig. 4 shows the error rates using the new activation function for different thresholds (x_0), compared to that using the *sigmoid* function. Each data point is averaged over 100 different runs for each city distribution and then averaged over the 200 different city distributions as defined in the following.

For city distribution i, the error of a valid tour j is defined by:

$$Err_{i,j} = \frac{d_{i,j} - d_{i,opt}}{d_{i,opt}} \tag{11}$$

where $d_{i,j}$ is the tour length of a valid tour j and $d_{i,opt}$ is the optimal (shortest) tour length of city distribution i.

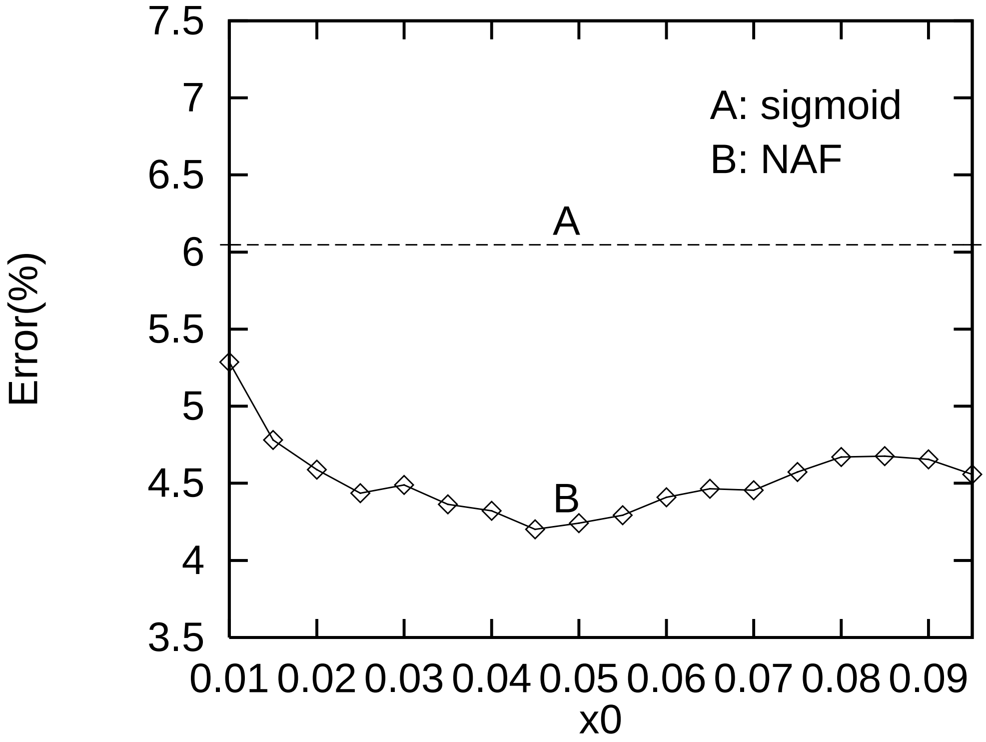

Figure 4: The error rates using the new activation function (NAF) for different thresholds (x_0), compared to that using the *sigmoid* function.

The error for city distribution i is the averaged error over all valid tours:

$$Err_i = \frac{\sum_{j=1}^{N_{valid,i}} Err_{i,j}}{N_{valid,i}} \tag{12}$$

The error shown in Fig. 4 is the average error of valid tours in all city distributions and is weighted by the percentage of valid tours for each city distribution:

$$Err = \frac{\sum_{i=1}^{N_{CityDist}} (Valid_i Err_i)}{\sum_{i=1}^{N_{CityDist}} Valid_i} \tag{13}$$

Fig. 4 shows that the new activation function can also reduce the error rate of the obtained solutions, improving the quality of solutions. The performance in Fig. 4 has a similar trend[2] as that in Fig. 3: the performance increases obviously at the left-side of the curve, becomes relatively flat around the middle, and decreases slightly at the right-side. This trend can be explained by reasons similar to those for Fig. 3.

Thus the new activation function can improve the performance in terms of both percentage of valid tours and error rates. To get a balanced improvement, the parameter x_0 should be chosen with consideration of both terms. From Fig. 3 and Fig. 4, the optimal value of x_0 is about 0.045. Compared with the *sigmoid* function, the new activation function at this x_0 value increases the percentage of valid tours by 38.6% (30.9% vs 22.3%) and reduces the error rate by 30.6% (4.20% vs 6.05%).

[2]Note that we are seeking minimum points in Fig. 4 while seeking maximum points in Fig. 3.

4. Relaxation Controlling Parameter

In this section, we present an approach [20] that includes a parameter (relaxation controlling parameter) to control the relaxation process in order to achieve better performance. We give motivation for using this parameter, describe the way it is included in the relaxation procedure, and show the simulation results.

4.1 Motivation

The process of relaxation has an important impact on the performance of the Hopfield network. To find a valid solution with high quality, the network must be relaxed at an appropriate rate, neither too fast nor too slow.

If the relaxation is too fast, there will be fewer opportunities for the exchange of information between neurons, and consequently a solution formed under this condition is less likely to have high quality. Also too fast a relaxation may cause undesirable instabilities and oscillations of the network.

On the other hand, too slow a relaxation also has a disadvantage. It makes an inefficient use of the network, because it takes many more iterations to reach an equilibrium.

If a parameter is properly applied to control the relaxation process, it would be possible to tune this parameter so the network can be relaxed at an appropriate rate.

4.2 Relaxation Controlling Parameter

In order to avoid the potential problems described above, we introduced a *relaxation controlling parameter (RCP)* to control the relaxation process. This parameter (denoted by P) is included in the dynamics of the network by the following equation:

$$\Delta U_{Xi}^{(new)} = P \Delta U_{Xi} \tag{14}$$

Where ΔU_{Xi} is given by Eq. (5) Thus, in the new formulation, the update of U_{Xi} is based on $\Delta U_{Xi}^{(new)}$, rather than on ΔU_{Xi}. Accordingly, Eq. (4) is modified to be of the form:

$$U_{Xi}^{(n+1)} = U_{Xi}^{(n)} + P \Delta U_{Xi} \tag{15}$$

4.3 Simulation Results

The performance of the network using RCP has been evaluated and compared to that of the original network through simulations. The simulation conditions were the same as those described in the last section for testing the new activation function.

Fig. 5 shows the percentage of valid tours using RCP for different P values and the comparison to that without using RCP. It shows that the percentage of valid tours is sensitive to the P value. The network performs better than that without using RCP when P is in the range from 2 to 3. But when P becomes too large (> 3.5) – that is, when the relaxation is too fast – the performance degrades dramatically.

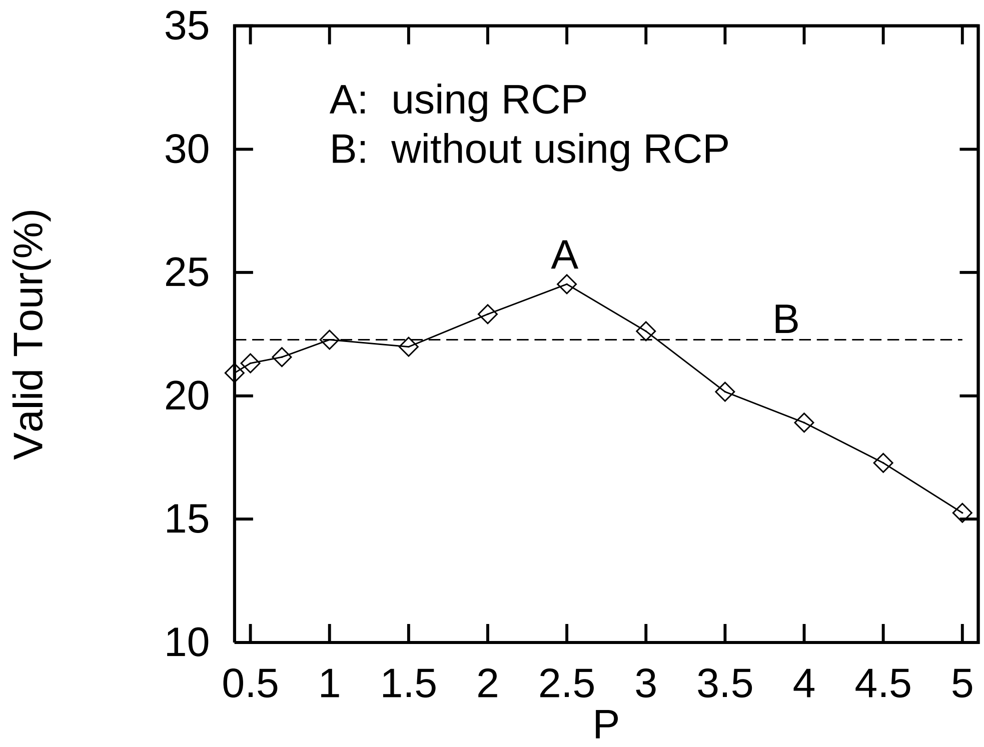

Figure 5: The percentage of valid tours using the relaxation controlling parameter (RCP) for different P values, compared to that without using RCP.

In contrast, when P becomes smaller (< 1) the curve appears flat (with a slight drop[3]) and approaches a limit. As would be expected, more iterations are required to converge to a solution as P becomes smaller. Thus, too small a P value is undesirable because of the inefficient use of the network.

It should be noted that the P value in Fig. 5 does not have an absolute meaning, because this value depends on the choices of other parameters in the simulations, as can be seen from Eq. (5) and Eq. (2). Thus if we choose a different set of parameters, the optimal value for P is usually different from that shown in Fig. 5. In this sense, a simulation result with respect to the value of P is valid only for the given parameter setting. For a different set of parameters, the optimal value for P needs to be reevaluated through simulations.

Fig. 6 shows the error rates using RCP for different P values and the comparison to that without using RCP. The dependence of the performance on the value of P is similar to that in Fig. 5, demonstrating again that a proper rate of relaxation – neither too small nor too large – is able to improve the performance of the network.

With the consideration of both Fig. 5 and Fig. 6, the optimal value for P is about 2.5, at which the percentage of valid tours is increased by 9.87% (24.5% vs 22.3%) and the error rate is reduced by 14.0% (5.20% vs 6.05%) as compared

[3]It is accounted for by reaching the limit of 1000 iterations (the maximum number of iterations allowed for each run).

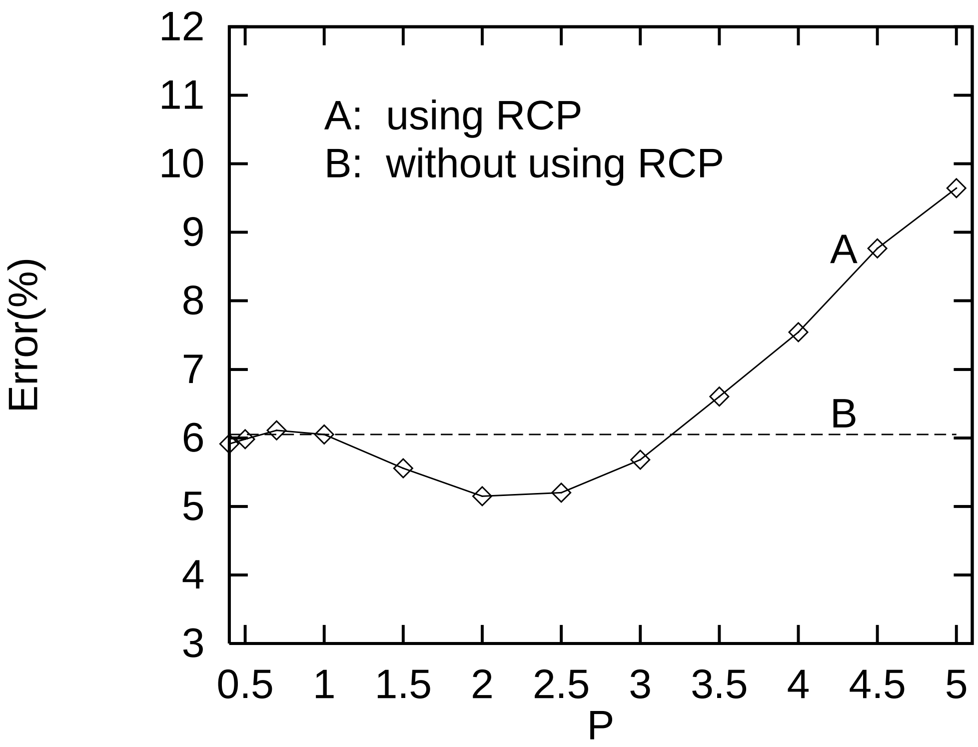

Figure 6: The error rates using the relaxation controlling parameter (RCP) for different P values, compared to that without using RCP.

with that without using RCP.

5. A New Relaxation Procedure

In this section, we present an approach [21] that uses a new relaxation procedure to replace the original one. We describe the new procedure, present the simulation results, and analyze the reasons for the improved performance.

5.1 Description of the New Procedure

Unlike the original procedure that first updates the input and then the activation for each neuron (using Eq. (2) - (6)), the new relaxation procedure updates the activation directly without using the update of the input as an intermediate step.

For convenience of presenting the new procedure, we can reformulate Eq. (5) into the form:

$$\Delta U_{Xi} = (-\frac{U_{Xi}}{\tau} + Net_{Xi})^{(n)}\Delta t \tag{16}$$

where Net_{Xi} is the *net input value* received by the neuron (X, i) from other neurons, and can be written as:

$$Net_{Xi} = \sum_{Y=1}^{N}\sum_{j=1}^{N} W_{Xi,Yj}V_{Yj} + I_{Xi} \tag{17a}$$

For *TSP*, the above equation for Net_{Xi} can be written as (using Eq. (2)):

$$Net_{Xi} = -A \sum_{j=1,j\neq i}^{N} V_{Xj} - B \sum_{Y=1,Y\neq X}^{N} V_{Yi}$$

$$-C(\sum_{Y=1}^{N}\sum_{j=1}^{N} V_{Yj} - N_0) - D \sum_{Y=1,Y\neq X}^{N} d_{XY}(V_{Y,i+1} + V_{Y,i-1}) \tag{17b}$$

In the new procedure, the activation value $V_{Xi}^{(n+1)}$ for neuron (X, i) at iteration

$(n+1)$ is updated according to:

$$V_{Xi}^{(n+1)} = V_{Xi}^{(n)} + R(T_{Xi} - V_{Xi}^{(n)}) \tag{18}$$

where T_{Xi} is the *target value* for the activation V_{Xi}, and $R(0.0 < R < 1.0)$ is a *relaxation rate.* T_{Xi} is derived from Net_{Xi} by using the *sigmoid* function:

$$T_{Xi} = \frac{1}{2}(1 + tanh(\frac{Net_{Xi}}{u_0'})) \tag{19}$$

where u_0' is the steepness parameter (its role is similar to that of u_0 in Eq. (6)).

According to the updating rule specified by Eq. (18), if $V_{Xi}^{(n)} > T_{Xi}$, then $V_{Xi}^{(n+1)} < V_{Xi}^{(n)}$; similarly if $V_{Xi}^{(n)} < T_{Xi}$, then $V_{Xi}^{(n+1)} > V_{Xi}^{(n)}$. Thus after updating, the activation value gets closer to the target value. The relaxation rate R determines how much V_{Xi} should change given the difference between the target value and the current activation value. The role of the relaxation rate here is to control the step size of updating V_{Xi} for each iteration. The value of R should be set to be a small number (< 1.0); otherwise, V_{Xi} may oscillate and cause instability of the network.

The differences between the original and the new relaxation procedure can be summarized as follows:

(a) The original procedure first updates the input value by Eq. (4), and then updates the activation value through the *sigmoid* function given by Eq. (6) (the input and the activation value are both important for specifying a neuron). The new procedure, however, directly updates the activation value using Eq. (18)

without using the update of the input as an intermediate step (the input value does not appear in any formula in the new procedure and is thus unnecessary).

(b) The new procedure has a relaxation rate to gain a better control over the relaxation process, but there is no direct analog in the original procedure.

(c) A parameter Δt appears in the original procedure (in Eq. (5) but not in the new procedure. However, the parameter u_0' (in Eq. (19)) in the new procedure has an effect similar to combining Δt and u_0 in the original procedure.

(d) The steepness parameter u_0' in the new procedure has a very different range than the similar parameter u_0 in the original procedure. u_0' has a much larger scale than u_0 since Δt is in the range about 10^{-5}.

(e) The time constant τ in the original procedure (in Eq. (5)) is not included in the new procedure. The reason is that its effect is negligible compared with the other terms (the τ value (=1.0) is much smaller than the coefficients A, B, C, and D, which are in the order of 10^2).

5.2 Simulation Results

We have evaluated the performance of the new relaxation procedure through simulations, and compared it with that of the original relaxation procedure. The simulation conditions were the same as those in the previous simulation.

The first step is to determine the value of u_0' in Eq. (19). u_0' reflects the steepness of the *sigmoid* function: smaller u_0' leads to a steeper slope around the center of the curve. The value of u_0' should be chosen so that the *sigmoid* function

(19) is in a sensitive range (from -10.0 to 10.0) for typical Net_{Xi} values. For the parameters given by (8), the value of Net_{Xi} given by (17b) is approximately in the range from 10 to 1000. Based on this estimation, we experimented with the following u_0' values: 50, 100, 150, 200, 300, 400 and 500. For each of them, the simulations with different values of relaxation rate R are conducted to evaluate the error rate and the percentage of valid tours. We found that the network has best overall performance when u_0' is 150. The performance drops slightly when u_0' is equal to 100 or 200. But when u_0' further decreases (< 100) or increases (> 200), the performance starts to drop rapidly. For the values of 400 and 500, no valid tour can be found by the network. So the value of u_0' is fixed at 150 for all the simulations reported in the following.

Fig. 7 shows the percentage of valid tours using the new procedure for different relaxation rates, compared to that using the original procedure. The result shows that the new procedure significantly improves the percentage of valid tours over the original procedure when the relaxation rate $R \leq 0.14$.

Another important observation from Fig. 7 is that the percentage of valid tours is sensitive to the relaxation rate in the new procedure: it significantly drops when the relaxation rate R is larger than 0.15 (approaching zero when $R > 0.17$), but its value is relatively stable when $R \leq 0.14$. As R becomes smaller, the curve appears flat and approaches a limit (similar to Fig. 5 and also due to the reason of reaching the limit of 1000 iterations).

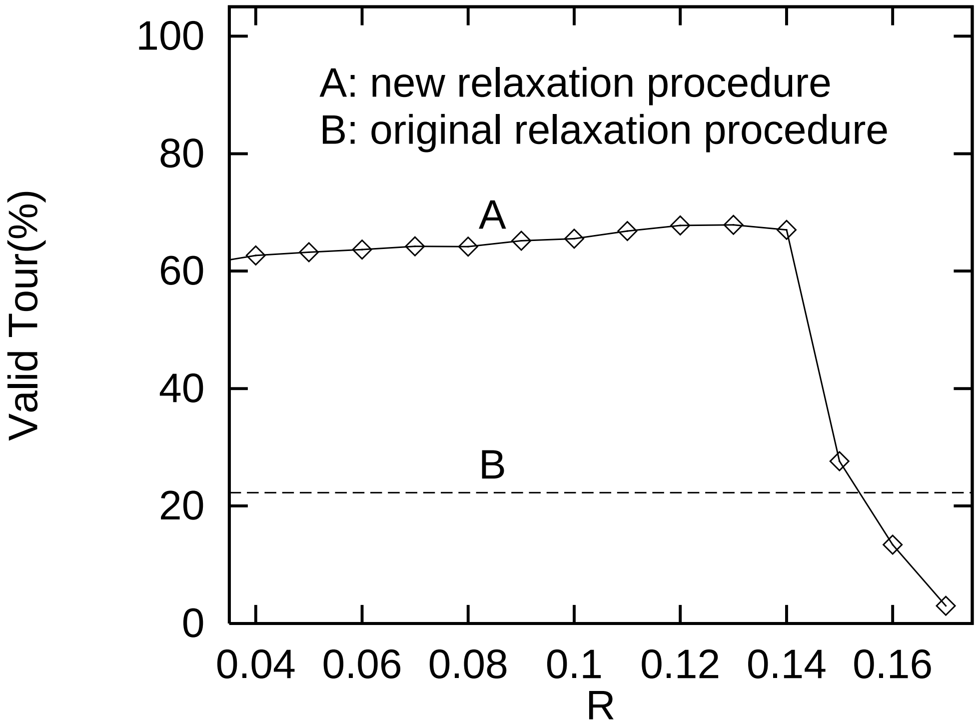

Figure 7: The percentage of valid tours of different relaxation rates using the new relaxation procedure compared to the original relaxation procedure.

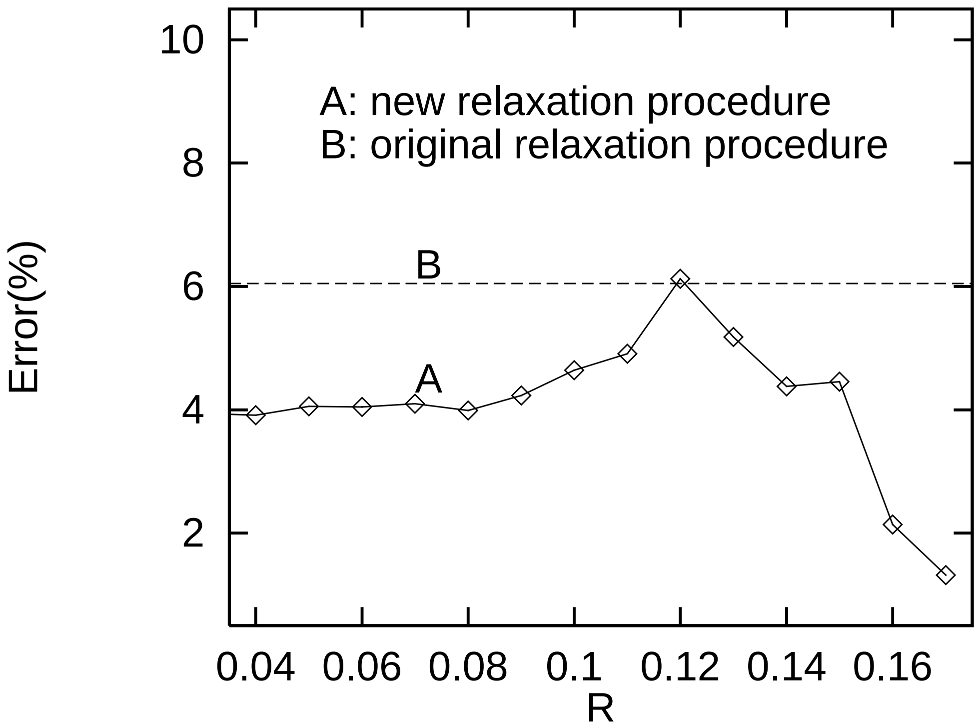

Figure 8: The error rates of different relaxation rates using the new relaxation procedure compared to the original relaxation procedure.

Although the percentage of valid tours is almost the same in a broad range of R (≤ 0.14), the observation shows that the average number of iterations for the network to converge to an equilibrium state differs for different R values. A smaller R requires a larger number of iterations for convergence since the network changes more slowly at each iteration step. So a larger R is preferred for sake of efficiency if two R values give the same percentage of valid tours and the same error rate.

Fig. 8 shows the errors of the solutions obtained by the network using the new relaxation procedure for different relaxation rates, compared to that obtained by using the original relaxation procedure. It has a trend similar to Fig. 7: relatively flat when R is smaller than a threshold (about 0.15) and then rapidly dropping when R becomes larger.

One observation from Fig. 8 is that the error rate is very small when $R > 0.16$. However, looking at Fig. 7, we can see that the percentages of valid tours are extremely low for these values. Hence these values are not good choices for R, because they do not result in a desired balance in the overall performance.

The reason for the dropping error in Fig. 8 when $R > 0.16$ is as follows. Recall from Fig. 7 that there are very few valid tours discovered at these points. When R is at these values, the network can only find valid tours for a very small partion of the 200 city distributions. Those city distributions typically have a very simple topology with an obvious optimal tour. Since only valid tours are used in the error calculation (Eq. (12) and Eq. (13)), the error rates at these points are therefore small.

However, when R is smaller than 0.16, the network can find valid tours for more city distributions. Since many of those city distributions have more complicated topologies, it is more difficult to find optimal tours. As a consequence, the average error starts to grow when R moves from 0.16 to smaller values, which explains the shape of the curve in Fig. 8.

As seen in Fig. 8, the new procedure also significantly improves the network over the original procedure in terms of error rate. The error rates of the new procedure are all lower than that of the original procedure except at one point

$R = 0.12$, as shown in Fig. 8,

To achieve a good overall performance, the value of R should be chosen properly with the consideration of a balanced improvement in terms of both the percentage of valid tours and the error rate. For the parameter setting specified in the simulation, the optimal value of R is about 0.06. At this point, the new procedure increases the percentage of valid tours by 185.7% (63.7% vs 22.3%) and decreases the error rate by 33.1% (4.05% vs 6.05%), as compared to the original procedure.

5.3 Analysis

Why does the new relaxation procedure perform significantly better than the original procedure? We compare and analyze the details of the two procedures and propose the following explanation for the difference in performance.

The main reason suggested to understand this difference is that these two procedures have different variation ranges for the activation value V_{Xi} between two consecutive iterations. The magnitude of the variation for V_{Xi} between iteration (n) and $(n+1)$ is:

$$|\Delta V_{Xi}| = |V_{Xi}^{(n+1)} - V_{Xi}^{(n)}| \tag{20}$$

In the original procedure, $V_{Xi}^{(n+1)}$ is the result of the *sigmoid* function of $U_{Xi}^{(n+1)}$ given by Eq.(6). The range of the *sigmoid* function is $(0.0, 1.0)$. $|\Delta V_{Xi}|$ is the difference of two *sigmoid* functions, and thus its possible range is also $(0.0, 1.0)$. When there is a significant difference between $U_{Xi}^{(n)}$ and $U_{Xi}^{(n+1)}$, this difference will be amplified by the *sigmoid* functions and causes a large value of $|\Delta V_{Xi}|$.

Such a large variation range for $|\Delta V_{Xi}|$ leads to a fast relaxation of the network. If the network is relaxed too fast, there will be less opportunity for exchange of information between neurons, and the solution formed under this condition is less likely to have high quality. Another potential problem with fast relaxation is possible instability and oscillation of the network.

In contrast, $|\Delta V_{Xi}|$ in the new relaxation procedure is given by $|R(T_{Xi} - V_{Xi}^{(n)})|$ according to Eq.(9). T_{Xi} is the *sigmoid* function shown in Eq.(10) and thus has a range of $(0.0, 1.0)$. V_{Xi} is initialized to be around 0.1, and its value will still be in the range from 0.0 and 1.0 after subsequent updates as long as the value of R is in the range from 0.0 to 1.0. Thus $|(T_{Xi} - V_{Xi}^{(n)})|$ is in the range of $(0.0, 1.0)$, and the variation range for $|\Delta V_{Xi}|$ $(= |R(T_{Xi} - V_{Xi}^{(n)})|)$ is $(0.0, R)$.

For a typical value of R chosen in our simulation, for instance, $R = 0.06$, $|\Delta V_{Xi}|$ will be in the range $(0.0, 0.06)$, which is much smaller than $(0.0, 1.0)$ in the original procedure. A smaller variation range for $|\Delta V_{Xi}|$ causes a smoother relaxation process and result in better performance for the network. This is the main reason for the improved performance by the new relaxation procedure.

6. Combining the New Activation Function and the Relaxation Controlling Parameter

In this section, we combine the new activation function (NAF) in Section 3 and the relaxation controlling parameter (RCP) in Section 4, and examine how combining can further improve performance.

6.1 Combining Method

The combining of these two approaches is quite straightforward. We use exactly the same steps as those in Section 4 except that we use the NAF, instead of the *sigmoid* function, as the activation function. Specifically, the combining method in this section uses Eq. (7) as the activation function, rather than Eq. (6).

6.2 Simulation Result

The performance of the combining method is evaluated through simulations. The parameters of NAF in the simulations are chosen based on the results in Section 3. The value of u_0 is still set to be 0.02, as in the previous simulation, and the values of x_0 is chosen to be 0.045. Other parameters in the simulations are the same as those in Section 4.

Fig. 9 shows the percentage of valid tours obtained by using the relaxation controlling parameter (RCP) method combined with the new activation function (NAF), compared to that without using NAF.

The result shows that the combining method (curve "A") can achieve a uni-

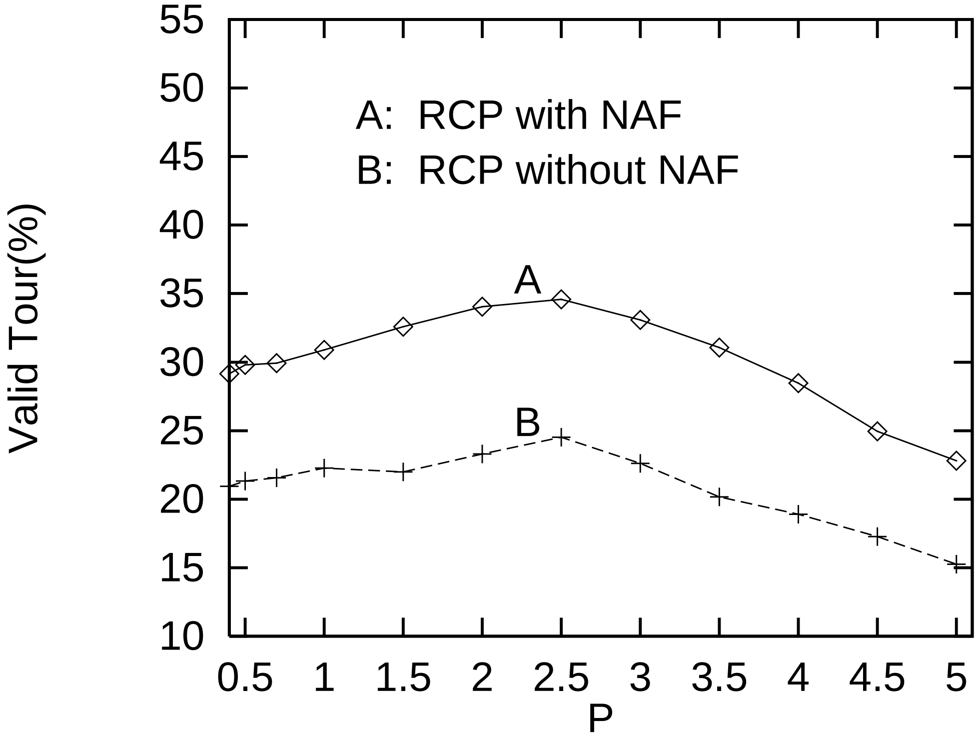

Figure 9: The percentage of valid tours obtained by the relaxation controlling parameter (RCP) method using (combined with) the new activation function (NAF), compared to that without using NAF.

formly higher percentage of valid tours (for all the P values) than the non-combining method (curve "B"). Both curves have a similar shape and the peak points are in the middle range from 2.0 to 3.0.

Similarly, Fig. 10 shows the error rates of the two methods. The combining method (curve "A") thus also outperforms the non-combining method (curve "B") in terms of the error rate.

These results demonstrate the advantages of the combining method over the non-combining method. To make a quantitative estimation of the improvement, we need to choose a P value for the comparison. From the simulation results in Section 4 and in this section, $P = 2.5$ is approximately the optimal value for both methods, and is thus used in the comparison. At this P value, the combining method increases the percentage of valid tours by 41.2% (34.6% vs 24.5%) and reduces the error rate by 10.4% (4.66% vs 5.20%), compared to the non-combining method.

If the combining method is compared to the original method (without using either NAF or RCP), it increases the percentage of valid tours by 55.2% (34.6% vs 22.3%) and reduces the error rate by 23.0% (4.66% vs 6.05%).

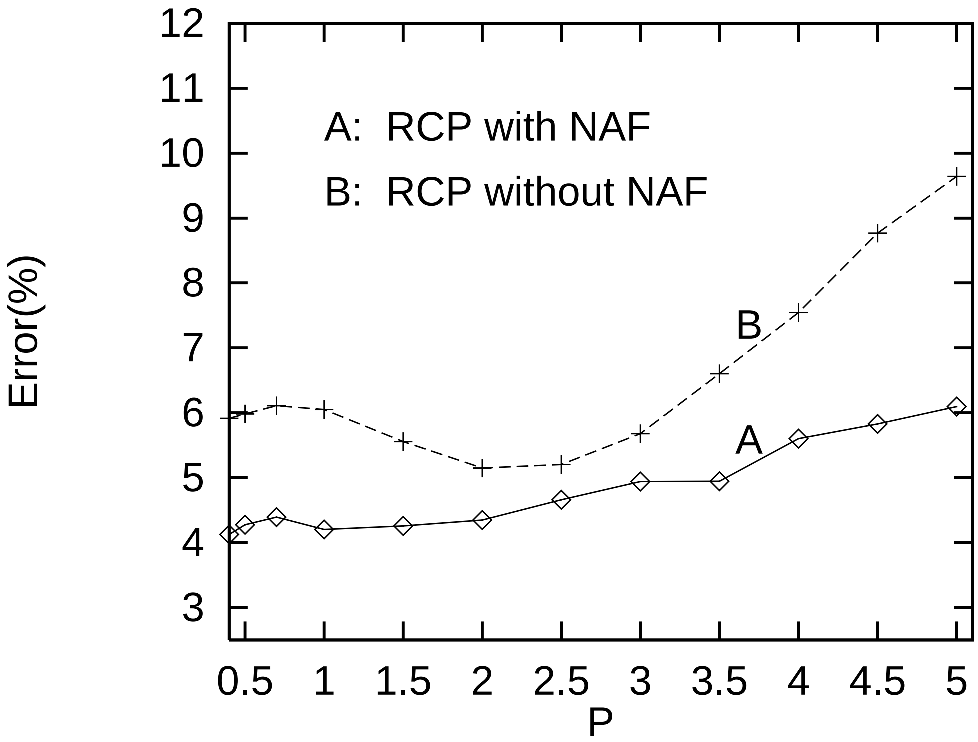

Figure 10: The error rate obtained by the relaxation controlling parameter (RCP) method using (combined with) the new activation function (NAF), compared to that without using NAF.

7. Combining the New Activation Function and the New Relaxation Procedure

In this section, we combine the new activation function (NAF) in Section 3 and the new relaxation procedure (NRP) in Section 5, and show that combining can lead to better performance.

7.1 Combining Method

The method of combining NAF and NRP is similar to that of combining NAF and RCP described in the last section. In the combining method, Eq. (19) (the *sigmoid* function) is replaced by the following function (using NAF in Eq. (7)):

$$T_{Xi} = \frac{0.5(1 + tanh(\frac{Net_{Xi}+x_0'}{u_0'}))}{1 + tanh(\frac{x_0'}{u_0'})} \qquad (U_{Xi} < 0)$$

$$T_{Xi} = \frac{tanh(\frac{x_0'}{u_0'}) + 0.5(1 + tanh(\frac{Net_{Xi}-x_0'}{u_0'}))}{1 + tanh(\frac{x_0'}{u_0'})} \qquad (U_{Xi} \geq 0) \tag{21}$$

which is used as the activation function. All the other steps in this section are the same as in Section 5.

7.2 Simulation Result

The performance of this combining method is evaluated through simulations. The values of parameters in the simulations are the same as those in Section 5 except for those related to the activation function.

The first step is to determine the parameters (u_0', x_0') in Eq. (21). The estimated optimal parameters based on the simulations in Section 3 can not be applied directly to the simulations here, because Section 3 uses the original relaxation procedure while this section uses the new relaxation procedure. Also recall that u_0' in Section 5 and u_0 in Section 3 have a large difference in their ranges. Thus we determine the values for u_0' and x_0' through simulations.

The following procedure is employed to estimate the parameters u_0' and x_0'. To compare the new activation function to the original activation function, we use the same value for the steepness parameter u_0'. Since u_0' is chosen to be 150.0 in Section 5, we also use this value for u_0' in this section.

To determine the value of x_0', we tried different x_0' values (20, 40, ... 220) while keeping the steepness parameter $u_0'(= 150.0)$ and the relaxation rate $R(= 0.06)$ fixed (The reason to choose $R = 0.06$ is because it gives a best overall performance in the simulations of Section 5) Among the x_0' values evaluated, $x_0' = 160$ achieves a best balanced performance (slightly better than the performance when $x_0' =$ 140, 180) with regard to both the percentage of valid tours and the error rate, and is thus used in the simulations.

The simulation results reported in the rest of this section are all based on the parameters determined above ($u_0' = 150.0$, $x_0' = 160$). Thus we can focus on evaluating how the performance of the network depends on the relaxation rate R, and on comparing the obtained results to those in Section 5 in order to determine if the combining method can improve the performance.

Fig 11 shows the percentage of valid tours obtained by combining the new relaxation procedure (NRP) and the new activation function (NAF), compared

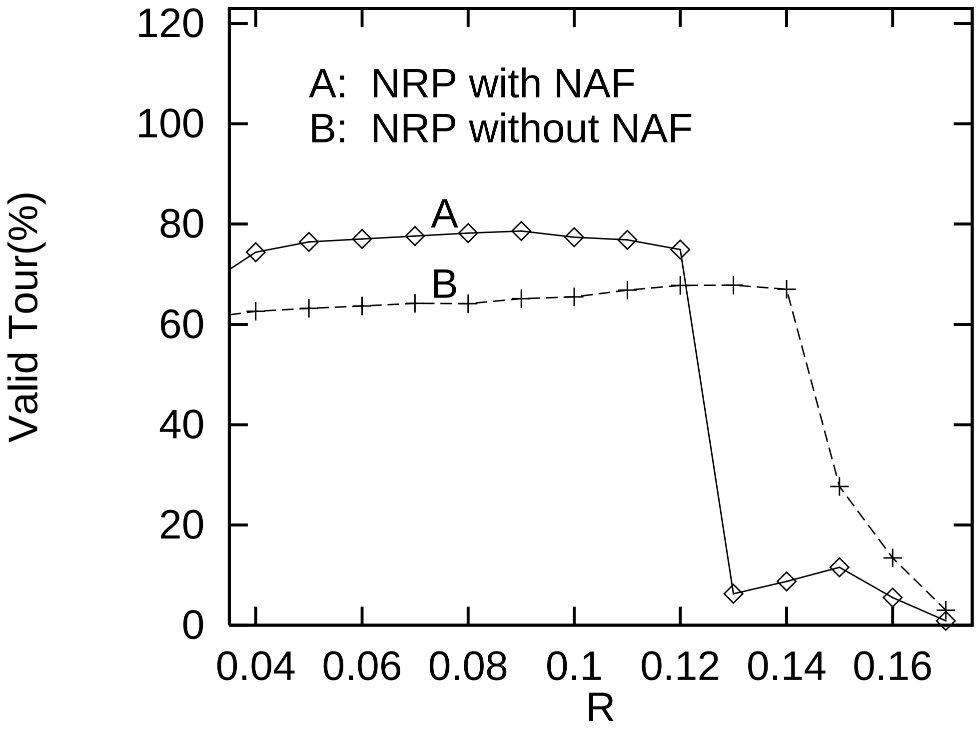

Figure 11: The percentage of valid tours obtained by the new relaxation procedure (NRP) using (combined with) the new activation function (NAF), compared to that without using NAF.

to that without using NAF.

The first observation is that the percentage of valid tours achieved by the combining method (curve "A") is higher than that by the non-combining method (curve "B") when $R \leq 0.12$, and their overall trends in performance are rather similar.

A second observation is that although the combining method performs better than the non-combining method when $R \leq 0.12$, it performs worse when R becomes larger. This shows that there is a certain degree of difference in their response to the relaxation rate R, in addition to a similar overall trend. Despite this difference, the combining method gives substantial improvement over the non-combining method when $R \leq 0.12$.

Fig. 12 shows the error rates using the combining method, compared to that using the non-combining method. All error rates achieved by the combined method (curve "A") are smaller than those by the non-combining method (curve "B").

The results in Fig. 11 and Fig. 12 demonstrates that the combining method performs better than the non-combining method in terms of both the percentage of valid tours and the error rate. We can quantitatively calculate the improvement at $R = 0.06$ (approximately optimal value for both methods) and obtain the following result: the combining method increases the percentage of valid tours by 21.0% (77.1% vs 63.7%) and reduces the error rate by 46.4% (2.17% vs 4.05%)

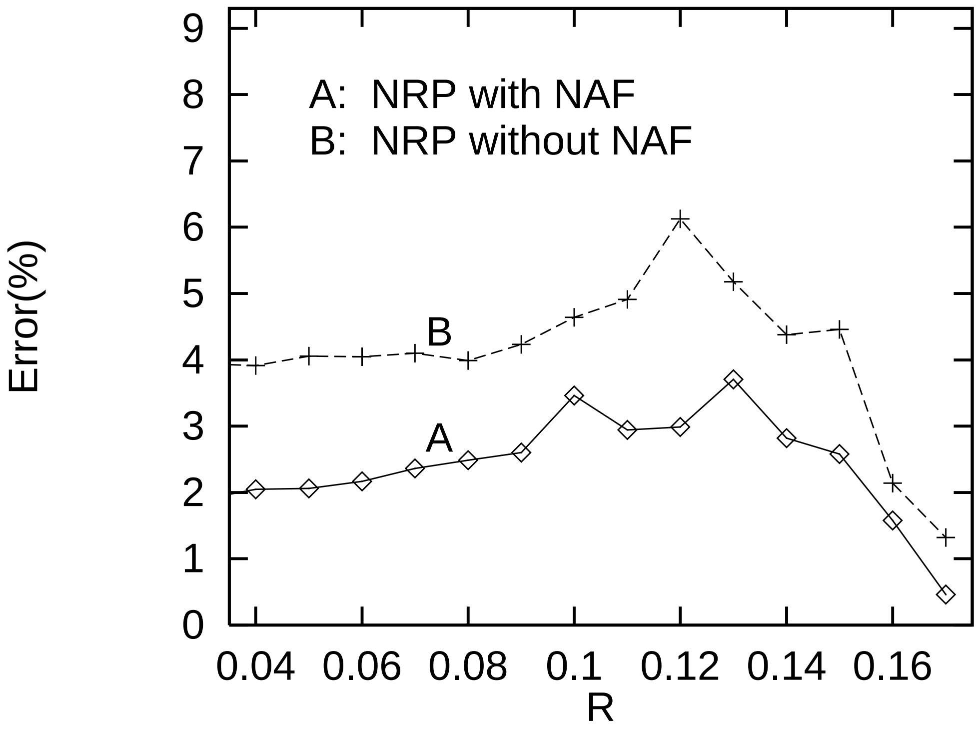

Figure 12: The error rates obtained by the new relaxation procedure (NRP) using (combined with) the new activation function (NAF), compared to that without using NAF

as compared to the non-combining method.

Similarly, if it is compared to the original Hopfield method (without using either NAF or NRP), the combining method increases the percentage of valid tours by 245.7% (77.1% vs 22.3%) and reduces the error rate by 64.1% (2.17% vs 6.05%).

8. Conclusion

In this paper we have presented three basic approaches that improve the performance of the Hopfield network. The first approach introduces a new activation function (NAF) to reduce negative effects of noise in the network. A second approach adopts a relaxation controlling parameter (RCP) to control the relaxation process so that the network can be relaxed at a proper rate. The third approach applies a new relaxation procedure (NRP) to achieve a smoother relaxation of the network.

All three approaches have been tested with a large number of simulations (based on 200 randomly generated city distributions of the 10-city TSP) and compared to the original Hopfield network. The results show that all of them are capable of improving the performance of the network, in terms of the percentage of valid tours and the error rate. Among them, the new relaxation procedure achieved the highest improvement: (185.7%, 33.1%), representing the percentage of increased valid tours, and the percentage of reduced error rate respectively. The improvements by the other two approaches were smaller but still significant: (38.6%, 30.6%) by the new activation function, and (9.87%, 14.0%) by the relaxation controlling parameter.

We have also studied the effects of combining the new activation function with the relaxation controlling parameter and the new relaxation procedure. The simulation results reveal that combining the methods are able to further enhance the performance. The improvement is (55.2%, 23.0%) by combining the new activation function with the relaxation controlling parameter, and the improvement is (245.7%, 64.1%) by combining the new activation function with the new relaxation procedure.

From the work presented in this paper, we draw the following three conclusions about the Hopfield network for solving optimization problems:

- The activation function is an important factor for the performance of the network. However, the *sigmoid* function has a major weakness (sensitive to noise) as the activation function. The performance can be improved by using a more noise-robust activation function, as the one we introduced in this work.

- The rate of the relaxation has an important impact on performance. Choosing a proper relaxation rate is essential for a good performance, because too fast a relaxation rate can cause a dramatic degradation in performance while too slow a relaxation rate reduces the efficiency of the network.

- The relaxation procedure has a significant influence on performance. The original relaxation procedure (updating the input and then the output) has a drawback that may lead to a less smooth relaxation. The procedure presented in this work (updating the output directly) is capable of forming a smoother relaxation and performing significantly better than the original procedure.

We plan to extent this work in the following directions:

- Further analysis to better understand how these methods lead to the observed improvements, with a goal to extend the improvements.
- Using these methods for other optimization applications other than *TSP* in order to better understand their general utility.

Bibliography

[1] Hopfield, J. J. and Tank, D. W. (1985), Neural Computations of Decisions in Optimization Problems. *Biological Cybernetics*, 52, pp. 141-152.

[2] Wilson, G. V. and Pawley, G. S. (1988), On the Stability of the Traveling Salesman Problem Algorithm of Hopfield and Tank. *Biological Cybernetics*, Vol. 58, pp. 63-70.

[3] Brandt, R. D., Wang, Y., Laub, A. J. and Mitra, S. K. (1988), Alternative Networks for Solving the Traveling Salesman Problem and the List-Matching Problem. *Proceedings of IEEE International Conference on Neural Networks*, San Diego, CA. II: 333-340.

[4] Aiyer, S. V. B., Niranjan, M. and Fallside, F. (1990), A Theoretical Investigation into the Performance of the Hopfield Model. *IEEE Transactions on Neural Networks*, Vol. 1, No. 2, pp. 204-215.

[5] Abe, S. (1989), Theories on the Hopfield Neural Networks. *Proceedings of IEEE INNS International Joint Conference on Neural Networks*, Vol. 1, pp. 557-564.

[6] Kamgar-Parsi, B. and Kamgar-Parsi, B. (1987), An Efficient Model of Neural Networks for Optimization. *Proceedings of IEEE First International Conference on Neural Networks*, San Diego, CA. III: 785-790.

[7] Bizzarri, A. R. (1991), Convergence properties of a modified Hopfield-Tank

Model. *Biological Cybernetics*, Vol. 64, pp. 293-300.

[8] Protzel, P. W., Palumbo, D. L. and Arras, M. K. (1993), Performance and Fault-Tolerance of Neural Networks for Optimization, *IEEE Transactions on Neural Networks*, Vol. 4, pp. 600-614.

[9] Li, S. Z. (1996), Improving Convergence and Solution Quality of Hopfield-Type Neural Networks with Augmented Lagrange Multipliers. *IEEE Transactions On Neural Networks*, Vol. 7, No. 6, pp. 1507-1516.

[10] Catania, V., Cavalieri, S. and Russo, M. (1996), Tuning Hopfield Neural Network by a Fuzzy Approach. *Proceedings of IEEE International Conference on Neural Networks*, pp. 1067-1072.

[11] Mehmet Ali, M. K. and Kamoun, F. (1993), Neural Networks for Shortest Path Computation and Routing in Computer Networks. *IEEE Transactions on Neural Networks*, Vol. 4, No. 6, pp. 941-954.

[12] Ansari, N., Hou, S. H. and Yu, Y. (1995), A new Method to Optimize the Satellite Broadcasting Schedules Using the Mean Field Annealing of a Hopfield Neural Network. *IEEE Transactions on Neural Networks*, Vol. 6, No. 2, pp. 470-483.

[13] Del Re, E. (1996), A Dynamic Channel Allocation Technique Based on Hopfield Neural Networks. *IEEE Transactions On Vehicular Technology*, Vol. 45, No. 1, pp. 26-31.

[14] Chung, P., Tsai, C., Chen, E. and Sun, Y. (1994), Polygonal Approximation Using A Competitive Hopfield Neural Network. *Pattern Recognition*, Vol. 27, No. 11, pp. 1505-1512.

[15] Lee, J. J., Shim, J. C. and Ha, Y. H. (1994), Stereo Correspondence Using the Hopfield Neural Network of a New Energy Function. *Pattern Recognition*, Vol. 27, No. 11, pp. 1513-1522.

[16] Lin, J., Cheng, K. and Mao, C. (1996), A Fuzzy Hopfield Neural Network for Medical Image Segmentation. *IEEE Transactions On Nuclear Science*, Vol. 43, No. 4, pp. 2389-2398.

[17] Cooper, B. S. (1995), Higher Order Neural Networks-Can they help us Optimise? in *Proceedings of the Sixth Australian Conference on Neural Networks (ACNN'95)*, pp. 29-32.

[18] Van den Bout, D. E. and Miller, T. K. (1989), Improving the Performance of the Hopfield-Tank Neural Network Through Normalization and Annealing. *Biological Cybernetics*, Vol. 62, pp. 129-139.

[19] Zeng, X. and Martinez, T. R. (1998a), A New Activation Function in the Hopfield Network for Solving Optimization Problems. Submitted to *Fourth International Conference on Artificial Neural Networks and Genetic Algorithms* (*ICANNGA'99*).

[20] Zeng, X. and Martinez, T. R. (1998b), Improving the Performance of the Hopfield Network By Using A Relaxation Rate. Submitted to *Fourth Inter-*

national Conference on Artificial Neural Networks and Genetic Algorithms (ICANNGA'99).

[21] Zeng, X. and Martinez, T. R. (1998c), A New Relaxation Procedure in the Hopfield Network for Solving Optimization Problems. Submitted to *Neural Processing Letters.*