Improving the Performance of the Hopfield Network By Using A Relaxation Rate *

Xinchuan Zeng and Tony R. Martinez
Computer Science Department, Brigham Young University, Provo, Utah 84602
Email: zengx@axon.cs.byu.edu, martinez@cs.byu.edu

Abstract

In the *Hopfield network*, a solution of an optimization problem is obtained after the network is relaxed to an equilibrium state. This paper shows that the performance of the Hopfield network can be improved by using a *relaxation rate* to control the relaxation process. Analysis suggests that the relaxation process has an important impact on the quality of a solution. A *relaxation rate* is then introduced to control the relaxation process in order to achieve solutions with better quality. Two types of relaxation rate (*constant* and *dynamic*) are proposed and evaluated through simulations based on 200 randomly generated city distributions of the 10-city *traveling salesman problem*. The result shows that using a relaxation rate can decrease the error rate by 9.87% and increase the percentage of valid tours by 14.0% as compared to those without using a relaxation rate. Using a dynamic relaxation rate can further decrease the error rate by 4.2% and increase the percentage of valid tours by 0.4% as compared to those using a constant relaxation rate.

1 Introduction

Interest in solving combinatorial optimization problems by neural networks was motivated by the work of Hopfield and Tank [1]. They proposed an approach of finding a suboptimal solution of the traveling salesman problem (*TSP*) by encoding the information of the cost and constraints of a *TSP* city distribution into the links between the neurons in a neural network and relaxing the network until it reaches an equilibrium state. 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). 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 that connect these neurons. The weights are set to encode the information about the constraints and the cost function of a particular *TSP* city distribution. Each neuron updates its input value based on the information received from all other neurons. They showed that a neural network so configured will always converge to a local minimum which may represent a valid solution. 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 of solving various optimization problems. Much research has been 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 there was some difficulty getting the Hopfield model to yield valid tours. For randomly generated sets of the 10-city *TSP*, Wilson and Pawley reported that only 8% of their trials resulted in valid tours. After their report, Brandt et al. [3] and Aiyer et al. [4] showed that better performance can be achieved by modifying the energy function. Li combined the Hopfield network with the "augmented Lagrange multipliers" algorithm from optimization theory [5]. Catania et al. applied a fuzzy approach to tune the parameters in the Hopfield network [6]. Liang added adjusting neurons to the Hopfield network for solving the quadratic assignment problem [7].

The performance of the Hopfield network has

*In *Proceedings of the International Conference on Neural Networks and Genetic Algorithms, 1999.*

been improved over the past decade. However, this model still has some basic problems [8, 9]. 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, and different parameter settings can lead to significant differences in the performance.

In the Hopfield model, a solution is achieved after the network is relaxed and has settled down to an equilibrium point. The process of relaxation is an important step in achieving a solution and have an important impact on the quality of the solution. However, there has not been enough effort to study this process and the underlying mechanism during the process is not well understood.

This paper addresses the issues of the effects of the relaxation process on the quality of the solutions obtained by the network. The analysis shows that the relaxation process is not trivial, but important for the network being able to find a good solution. A relaxation rate is introduced to control the pace of relaxation. The performance of this approach has been tested through simulations based on 200 randomly generated city distributions of the 10-city *TSP*. The simulation reveals that the quality of a tour and the percentage of valid tours depends on the relaxation rate. The error rate can be reduced by 9.87% and the percentage of valid tours can be increased by 14.0% compared to those without using a relaxation rate. A dynamic relaxation rate, with varying value during relaxation process, is also experimented. The error rate drops by a further 4.2% and the percentage of valid tours increases by 0.4% compared to those using a constant relaxation rate. A dynamic amplification parameter has also been experimented and its performance has been compared to those using relaxation rates.

The rest of the paper is organized as follows. A brief outline of the basic structure of the Hopfield network is given in Sec. 2. The motivations and the procedure of including a relaxation rate in the network are described in Sec. 3. The simulation results are presented in Sec. 4. Finally, a summary of this work is given in Sec. 5.

2 Basics of the Hopfield network

The Hopfield network [1] consists of a set of neurons and the links connecting the neurons. For an N-city *TSP*, there are $N \times N$ fully connected neurons in the network. The row index for a neuron represents the city. The column index represents the order of the city in the tour. The weights of the connecting links are determined according to the constraints and the cost function. *TSP* includes the following constraints: The salesman must visit each city exactly once in a tour (each row has exactly one "on" neuron) and must visit exactly one city at any time in a tour (each column has exactly one "on" neuron). The cost function is constructed to reflect the objective that the salesman chooses a valid tour with minimum tour length. Therefore, the Hopfield network is configured to have a better chance to find short tours.

The constraints and the cost function for *TSP* can be represented by an energy function. The energy function is then used to determine the values of all weights in the network. Hopfield's original energy function for an N-city *TSP* is given by [1]:

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

where X and Y are row indices; i and j are column indices; V_{Xi} is the activation for each neuron; and d_{XY} is a measure of the distance between cities X and Y. The first two terms enforce the constraint that no city can be visited more than once. The third term reflects the constraint that each city should be visited. The last term represents the distance cost function. The value of each parameter (A, B, C, and D) is the measure of the importance of the corresponding term.

Each neuron has a current state input U_{Xi} and an activation (output) V_{Xi}. The initial values of U_{Xi} for all neurons are first set to be a small constant value which is determined by the condition: $\sum_{X=1}^{N}\sum_{i=1}^{N} V_{Xi} = N$. To break the symmetry of the network, a small fraction of random noise is added to the initial values of each U_{Xi}. The network is then relaxed until an equilibrium is reached.

After each iteration, U_{Xi} is updated based on the following equation:

$$\frac{dU_{Xi}}{dt} = -\frac{U_{Xi}}{\tau} - \frac{\partial E}{\partial V_{Xi}} \quad (2)$$

where $\tau(= 1.0)$ is the time constant.

V_{Xi} is also updated after each iteration and its value is determined by U_{Xi} through an *activation (output) function*. In the Hopfield network, the activation function is the *sigmoid* function given by:

$$V_{Xi} = \frac{1}{2}(1 + tanh(\frac{U_{Xi}}{u_0})) \quad (3)$$

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 connecting weights. They tested this model for a 10-city *TSP*. For a total of 20 runs, they claimed 16 valid tours, and half of the valid tours were optimal.

3 Relaxation rate

The quality of a solution obtained by the Hopfield network can be affected by certain factors, for example, the parameter setting in the energy function and the random noise in the initial input values of neurons [2].

This work focuses on studying the effects of the relaxation process on the performance of the network. Intuitively, the development of the network should be different when the network is relaxed with a different speed. One of the important factors that could influence the quality of a solution is the difference in the frequency that a neuron receives information from other neurons. This information reflects the cost and constraints encoded in the network, and hence the amount of the information propagated through the network has an important impact on the quality of a solution.

If the network is relaxed too fast, there will be fewer opportunities for exchange of the information between neurons, and therefore a solution formed under this condition is less likely to have a high quality. Another potential problem with a fast relaxation is a possible instability or oscillation of the network.

Relaxing the network at too slow a rate also has a disadvantage: an inefficient use of the network due to an unnecessarily large number of iterations required to form a solution.

In order to avoid the potential problems described above and to have better control of the relaxation process, a *relaxation rate* is introduced in the network dynamics. The function of the relaxation rate is to adjust the speed of the relaxation so that solutions with better quality can be obtained. Specifically, the input of the neuron (X, i) is updated according to the following dynamic equation:

$$\frac{dU_{Xi}^{(new)}}{dt} = R\frac{dU_{Xi}}{dt} \quad (4)$$

where R is the *relaxation rate* and $\frac{dU_{Xi}}{dt}$ is given by Eq. (2). The relaxation rate R reflects how fast the network is relaxed. The value of R is an adjustable parameter and can be determined empirically.

Two types of relaxation rate are studied and evaluated in simulation. One is a *constant* relaxation rate which is invariant through the entire relaxation process. The other is a *dynamic* relaxation rate which depends on the iteration and is different at different stages of relaxation. The motivation for studying the dynamic relaxation rate is to explore the possibility that there are different optimal relaxation rates at different phases of relaxation. Simulation results on both types of relaxation rates are given in the following section.

4 Simulation results

The performance of the Hopfield network including a relaxation rate is evaluated by simulation, and is compared to that of the network without using a relaxation rate. Evaluation is based on 200 randomly generated 10-city *TSP* city distributions, including wide varieties of city distributions. The performance of an algorithm usually depends on the topology of the city locations in a city distribution, and different algorithms may favor different types of distributions. Using a large number of city distributions can reduce this effect and obtain a better evaluation of the algorithms.

For each of the 200 city distributions, there are 100 runs with different random noises added to the initial inputs of neurons. Each evaluated parameter is 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 shown in the following simulation results.

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

$$A = B = D = 500, \quad C = 200, \quad N_0 = 15 \quad (5)$$

The value of dt in Eq. (2) and Eq. (4) is set to be 10^{-5}. The fraction of random noise in the initial values of neurons is set to be 0.001. 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.

4.1 Constant relaxation rate

We first presents the simulation results using a constant relaxation rate that does not change through the entire relaxation process.

Fig. 1 shows the errors of the solutions obtained by the network using different constant relaxation rates compared to that obtained without using a relaxation rate. Throughout the paper, the network "without using a relaxation rate" means that Eq. (2) instead of Eq. (4) is used as the dynamic equation. It is equivalent to use a default value $R = 1.0$ if Eq. (4) is used as the dynamic equation.

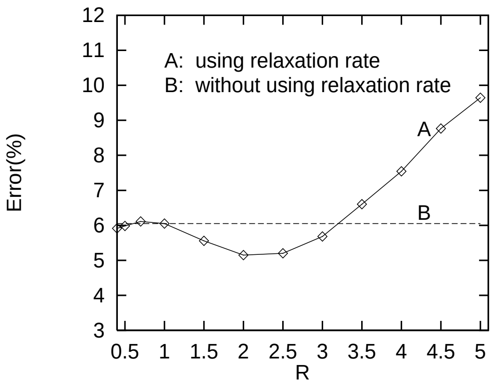

Figure 1: The errors (%) of different relaxation rates and that without using a relaxation rate.

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

$$Err_{i,j} = \frac{d_{i,j} - d_{i,opt}}{d_{i,opt}} \tag{6}$$

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

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

$$Err_i = \frac{\sum_{j=1}^{N_{valid,i}} Err_{i,j}}{N_{valid,i}} \tag{7}$$

where $N_{valid,i}$ is the number of valid tours among the total of $N_{total,i}$ runs with different initial input values for the city distribution i. In the simulation $N_{total,i} = 100$ for all city distributions.

The percentage of valid tours for city distribution i is defined by:

$$Valid_i = \frac{N_{valid,i}}{N_{total,i}} \tag{8}$$

The error shown in Fig. 1 is the average error of valid tours in all city distributions. It is weighted by the percentage of valid tours for each city distribution and is given by:

$$Err = \frac{\sum_{i=1}^{N_{CityDist}} (Valid_i Err_i)}{\sum_{i=1}^{N_{CityDist}} Valid_i} \tag{9}$$

where $N_{CityDist}$ is the total number of city distributions and is equal to 200.

The result in Fig. 1 shows that the error rate depends on the relaxation rate. The error rate can be reduced by using a relaxation rate in a proper range (between 1.0 and 3.0). However, when the relaxation rate becomes too large (> 3.5) the error rate increases rapidly, showing an obvious disadvantage of a fast relaxation. In contrast, when the relaxation rate becomes too small (< 1.0), the error rate approaches a limit without much change. But as we observed in the experiment, more iterations are needed to converge to a solution when the relaxation rate becomes smaller. Thus if two relaxation rates lead to a similar performance, a larger one is preferred in oreder to achieve a higher efficiency.

Fig. 2 shows the percentage of valid tours using different relaxation rates, compared to those without using a relaxation rate.

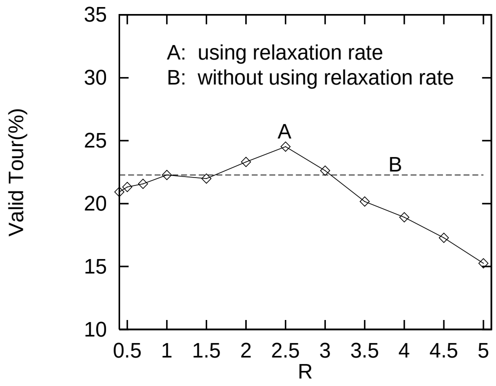

Figure 2: The percentage of valid tours using different relaxation rates and that without using a relaxation rate.

The percentage of valid tours is the weighted average over 200 city distributions:

$$Valid = \frac{\sum_{i=1}^{N_{CityDist}} Valid_i}{N_{CityDist}} \tag{10}$$

The result in Fig. 2 has a similar pattern as in Fig. 1, and the improvement occurs when the relaxation rate is in the middle range. This shows that a relaxation rate has a similar impact on the percentage of valid tours and on the error rate.

With the consideration of both Fig. 1 and Fig. 2, the optimal value for the relaxation rate is about 2.5. At this point, the error rate is reduced by 14.0% (5.20% vs 6.05%) and the percentage of valid tours is increased by 9.87% (24.5% vs 22.3%) as compared to that without using a relaxation rate.

4.2 Dynamic relaxation rate

A *dynamic* relaxation rate with the following form is included in the network:

$$R(M) = R_0 + \frac{(M - M_0)(R_1 - R_0)}{M_1 - M_0} \qquad (M_0 \leq M < M_1)$$

$$R(M) = R_1 \qquad (M \geq M_1) \quad (11)$$

where $R(M)$ is the dynamic relaxation rate which is a function of iteration M. $R(M)$ is equal to R_0 initially (when $M = M_0 = 0$), and then increases linearly with M until reaching R_1 at $M = M_1$.

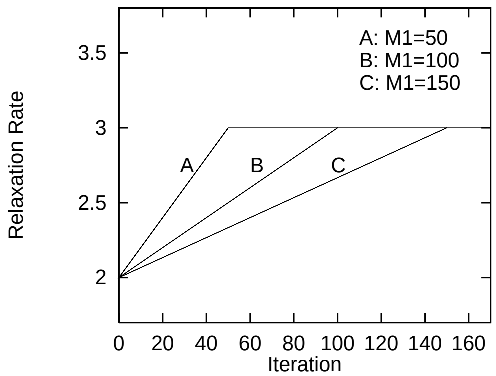

Figure 3: Dynamic relaxation rates with the parameters: $M_0 = 0$, $R_0 = 2.0$, $R_1 = 3.0$, and for different values of M_1.

Fig. 3 shows the shapes of $R(M)$ with the parameters: $M_0 = 0$, $R_0 = 2.0$, $R_1 = 3.0$, and for different values of M_1. The relaxation rate has a lower value during early iterations so that the neurons have sufficient opportunity to exchange information and to form a good prototype. Then its value gradually increases to speed up the convergence.

Fig. 4 and Fig. 5 show the error rate and the percentage of valid tours using dynamic relaxation rate, compared to those using constant relaxation rate. The figures show the results of dynamic relaxation rate using several different M_1 values. The constant relaxation rate for comparison is chosen to be 2.5 because it achieves the best performance as shown in the above discussion.

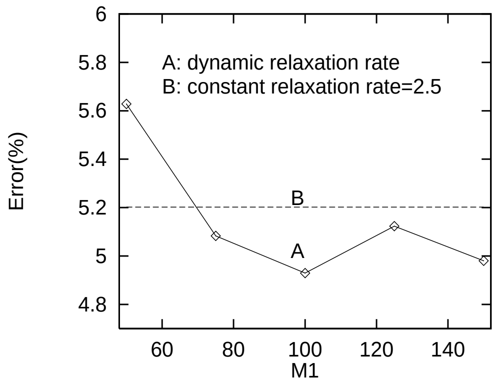

Figure 4: The error rate (%) using dynamic relaxation rate compared to that using constant relaxation rate.

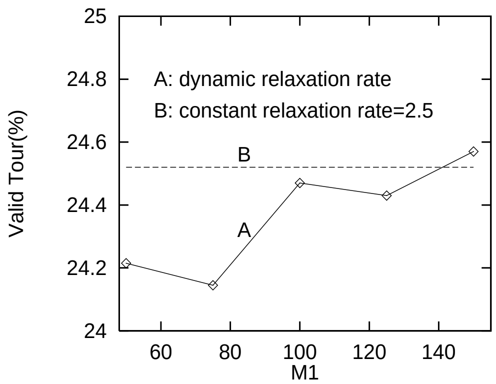

Figure 5: The percentage of valid tours using dynamic relaxation rate compared to that using constant relaxation rate.

We can see that the performance of a dynamic relaxation rate depends on the parameter M_1. Among the M_1 values experimented in the simulation, a larger value (i.e. a slower increase of the relaxation rate from 2.0 to 3.0) can achieve a relatively better performance.

At $M_1 = 150$ (about the optimal value), using a dynamic relaxation rate can reduce the error rate by 4.2% (4.98% vs 5.20%) and increase the percentage of valid tours by 0.4% (24.6% vs 24.5%) as compared to those using the constant relaxation rate ($R = 2.5$). If compared to the result without using relaxation rate, it reduce the error rate by 17.7% (4.98% vs 6.05%) and increase the percentage of valid tours by 10.3% (24.6% vs 22.3%). This result shows that the improvement on the error rate by a dynamic relaxation rate is more obvious than that on the percentage of valid tours.

4.3 Dynamic amplification parameter

We have also experimented with a dynamic amplification parameter $u_0(M)$ which slowly increases with iteration M (similar to Eq. (11)) and comapared its performance with those achieved by using relaxation rates.

In the simulation, $u_0(M_1)$ is increased linearly from 0.02 to 0.03 with different M_1 values (50, 100, 200, 500, 900). The best performance is achieved at $M_1 = 100$, which leads to a decrease of the error rate by 7.1% (5.62% vs 6.05%) and an increase of the percentage of valid tours by 3.1% (23.0% vs 22.3%) as compared to those using the original network.

This result shows that a dynamic amplification parameter can also improve the performance of the network. But the improvement achieved by this approach smaller than that achieved using a constant or dynamic relaxation rate.

5 Summary

In this paper, we have analyzed the relaxation process of the Hopfield network and proposed that the rate of relaxation has an important impact on the performance of the network. A relaxation rate has been introduced into the network dynamics to control the pace of network relaxation. The network using a relaxation rate has been shown to have a better performance than that without using a relaxation rate.

According to the simulation results based on 200 randomly generated city distributions of the 10-city *TSP*, the error rate can be reduced by 9.87% and the percentage of valid tours can be increased by 14.0% as compare to those without using a relaxation rate. A dynamic relaxation rate has also been proposed and compared to the constant relaxation rate, and it can further reduce the error rate by 4.2% and increase the percentage of valid tours by 0.4%. We have also experimented with a dynamic amplification parameter and compared its performance to that using relaxation rates.

In future work we plan to experiment with different forms of dynamic relaxation rate, including exponential form and quadratic form, and compare their performance with the linear form. We also plan to evaluate the performance of the relaxation rate on applications other than *TSP*.

Acknowledgments

This research is funded in part by a grant from *fonix* Corp.

References

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

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

[3] Brandt, R. D., Wang, Y., Laub, A. J. and Mitra, S. K.: 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, 1988.

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

[5] Li, S. Z.: 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, 1996.

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

[7] Liang, Y.:Combinatorial Optimization by Hopfield Networks Using Adjusting Neurons. Information Sciences, vol. 94, pp. 261-276, 1996.

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

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