Improving the Hopfield Network through Beam Search

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 this paper we propose a beam search mechanism to improve the performance of the Hopfield network for solving optimization problems. The beam search readjusts the top M ($M > 1$) activated neurons to more similar activation levels in the early phase of relaxation, so that the network has the opportunity to explore more alternative, potentially better solutions. We evaluated this approach using a large number of simulations ($20,000$ *for each parameter setting), based on 200 randomly generated city distributions of the 10-city traveling salesman problem. The results show that the beam search has the capability of significantly improving the network performance over the original Hopfield network, increasing the percentage of valid tours by 17.0% and reducing error rate by 24.3%.*

1 Introduction

Hopfield and Tank [1] proposed a neural network approach to find approximate solutions to combinatorial optimization problems, such as the *traveling salesman problem* (*TSP*). This approach has since received intensive attention from various research fields, due to its advantages over other approaches, such as fast convergence, parallel processing, and potential hardware implementation.

However, there have been some issues about the performance of the Hopfield network, which have been active research topics in the past decade. In their experiments with the Hopfield network, Wilson and Pawley [2] showed that the network often failed to converge to valid solutions; for those converged solutions, their qualities were often far from optimal. To address these issues, much research has focused on analyzing and improving the original model to obtain more valid high-quality solutions. Brandt et al. [3] and Aiyer et al. [4] modified the energy function to enforce stricter constraints for valid solutions to improve the convergence. Li [5] combined the Hopfield network with the "augmented Lagrange multipliers" algorithm from optimization theory, capable of improving convergence and quality but requiring many more iterations. Catania et al. [6] applied a fuzzy approach to tune the parameters of the network. Although these approaches have demonstrated improvement of performance, most of them were empirically tested on only a single or a small number of TSP city distributions, without reporting general performance on a large number of city distributions.

Despite the improvement of performance over the past decade, there are still unsolved problems for the Hopfield network [7, 8]. One problem is that the performance is inconsistent – good for some TSP city distributions but poor for others. The performance is usually better for distributions with a simple topology, but poor for those with complex topology or multiple clusters. Another problem is that the performance is sensitive to the choice of parameters in the energy function: a different parameter setting can lead to a significant difference in performance. In previous work, we addressed these problems by proposing a rescaling scheme to reduce clustering effects [9], and proposing a new relaxation procedure to reach a smoother relaxation process [10], which have been shown capable of significantly improving the performance based on a large number of simulations.

In this work we propose a beam search mechanism in the network which has the capability of searching more and better solutions across a wider range. In this approach, multiple neurons are kept at a similar activation level to allow the network the opportunity to explore a wide spectrum of potential solutions. We have tested this approach through a large number of simulations ($20,000$ for each parameter setting) based on 200 randomly generated city distributions of 10-city *TSP*. The results show that the beam search is capable of increasing the percentage of valid tours by 17.0% and decreasing the error rate by 24.3%.

2 Hopfield Network Basics

For an N-city *TSP*, the Hopfield network [1] has $N \times N$ fully connected neurons in the network, in which the row index represents the city and the column index represents the order of city in a tour. The constraints and the cost of the *TSP* are 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]:

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

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 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:

$$W_{Xi,Yj} = -A\delta_{XY}(1-\delta_{ij}) - B\delta_{ij}(1-\delta_{XY})$$
$$-C - Dd_{XY}(\delta_{j,i+1} + \delta_{j,i-1}) \qquad (2)$$

where δ_{ij} is equal to 1 if $i = j$, and equal to 0 otherwise. Each neuron (X,i) is also connected to an external input current: $I_{Xi} = CN_0$. 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 small random noise 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. The input value $U_{Xi}^{(n+1)}$ at iteration step *(n+1)* is given by:

$$U_{Xi}^{(n+1)} = U_{Xi}^{(n)} + \Delta U_{Xi} \qquad (3)$$

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

$$\Delta U_{Xi} = (-\frac{U_{Xi}}{\tau} + \sum_{Y=1}^{N}\sum_{j=1}^{N} W_{Xi,Yj}V_{Yj} + I_{Xi})^{(n)}\Delta t \qquad (4)$$

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 hyperbolic tangent function:

$$V_{Xi}^{(n+1)} = \frac{1}{2}(1 + tanh(\frac{U_{Xi}^{(n+1)}}{u_0})) \qquad (5)$$

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 ($W_{Xi,Yj} = W_{Yj,Xi}$) weights.

3 Beam Search Mechanism

The proposed beam search mechanism keeps the top M ($M > 1$) activated neurons (M is called *beam size*) at a similar activation level in the early phase of relaxation to give the network opportunity to explore alternative (potentially better) solutions. The mechanism to achieve this effect is to readjust the top M neurons with the highest activation in each column after each iteration, so that all the M neurons have a similar opportunity of being activated in the next iteration.

Let $V_{X_1 i}, V_{X_2 i} ... V_{X_M i}$ be top M activated neurons in column i. Then their average $\overline{V}_i$ is given by

$$\overline{V}_i = \frac{\sum_{s=1}^{M} V_{X_s i}}{M} \qquad (6)$$

The beam search algorithm readjusts activation $V_{X_s i}$ to a new value $V'_{X_s i}$ ($s = 1, 2...M$) using the formula:

$$V'_{X_s i} = \overline{V}_i + (1.0 - \alpha)(V_{X_s i} - \overline{V}_i) \qquad (7)$$

where α is a parameter, called *beam intensity*, which controls the amount of readjustment and thus determines how similar the M activations are after readjustment. The value for α is in the range from 0.0 to 1.0. When $\alpha = 0.0$, $V'_{X_s i} = V_{X_s i}$, i.e., there is no readjustment. When $\alpha = 1.0$, $V'_{X_s i} = \overline{V}_i$, which is total readjustment (*i.e.*, equalizing all the M activations). When $0.0 < \alpha < 1.0$, $V'_{X_s i}$ is between $V_{X_s i}$ and $\overline{V}_i$, which is a partial readjustment. Using a *continuous* parameter α as a way to control the search can be considered as a *graded* beam search. The purpose of applying beam intensity α is to study the network's performance across the transition spectrum from non-beam-search to full beam-search.

The beam search stops after L iterations to allow the network to converge to a final solution. The values of beam size M, beam intensity α, and beam search iteration L are determined empirically through a large number of simulations, as will be shown in the next section. Besides replacing $V_{X_s i}$ by $V'_{X_s i}$, all the other parameters, as well as all the formulas and procedures, are kept exactly the same as described in the last section.

4 Simulation Results

We have evaluated the performance of the beam search through a large number of simulations based on 200 randomly generated 10-city *TSP* city distributions, including a wide variety of topologies.

Many previous studies used only one (e.g., [1]) or a small number of city distributions (e.g., 10 distributions in [2]) in their simulations. This may lead to unreliable conclusions when comparing two algorithms, because the performance of an algorithm usually depends on the topology in a city distribution, and different algorithms may favor different topology types. Using a large number of city distributions can reduce this effect and allows a better evaluation.

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

For a fixed set of parameters (dt, u_0, *etc.*), the quantities to be evaluated 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 for each parameter setting to obtain simulation results. We experimented with different sets of $20,000$ runs for a fixed set of parameters. The results show that the variations are small enough to give stable evaluations on estimated quantities.

The original energy function of the Hopfield network was used in the simulation, and the parameters in the energy function are those used by Hopfield and Tank [1]: $A = B = D = 500$, $C = 200$, $N_0 = 15$. The value of dt in Eq. (4) is set to be 10^{-5}, and the value of u_0 in Eq. (5) is set at 0.02.

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 result using 0.001 is slightly better than the others.

We evaluated the performance of the beam search by comparing it with the original Hopfield network (without using the beam search), and studied how its performance depends on the three parameters: beam size M, beam intensity α, and beam search iteration L. In the following figures, we will show how the performance varies with one parameter while keeping the other two fixed. We have experimented with a large number of different combinations of parameter settings and found that the setting, $M = 4$, $\alpha = 0.1$, and $L = 80$, gave overall best results in terms of both quality and convergence. In each figure, two of these parameters are fixed when we show how the performance varies with the third parameter.

Fig. 1 and Fig. 2 show how the percentage of valid tours and error rate (%) vary with different beam sizes M when using the beam search technique (with fixed beam intensity $\alpha = 0.1$ and beam search iteration $L = 80$), and compare them with those using the original Hopfield network (without the beam search).

The percentage of valid tours displayed in Fig. 1 is defined as the weighted average percentage of valid tours over $N_{CityDist}$ (= 200) city distributions:

$$Valid = \frac{\sum_{i=1}^{N_{CityDist}} Valid_i}{N_{CityDist}} \tag{8}$$

where $Valid_i$ is the percentage of valid tours for city distribution i and is defined as follows. 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. Let $N_{valid,i}$ be the number of valid tours among the total of $N_{total,i}$ runs, and $Valid_i$ be defined by

$$Valid_i = \frac{N_{valid,i}}{N_{total,i}} \tag{9}$$

The error rate (%) shown in Fig. 2 is calculated as follows. For city distribution i, the error of a valid tour j is defined by

$$Error_{i,j} = \frac{d_{i,j} - d_{i,optimal}}{d_{i,optimal}} \tag{10}$$

where $d_{i,j}$ is the tour length of a valid tour j and $d_{i,optimal}$ is the optimal (shortest) tour length of the city distribution i. The error for city distribution i is the averaged error over all valid tours:

$$Error_i = \frac{\sum_{j=1}^{N_{valid,i}} Error_{i,j}}{N_{valid,i}} \tag{11}$$

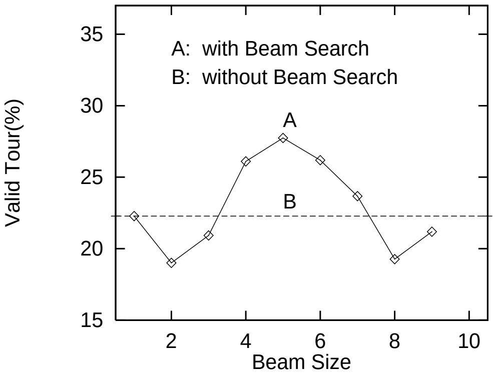

Figure 1: Percentage of valid tours vs beam size M using the beam search technique ($\alpha = 0.1$, $L = 80$), compared to that using the original network.

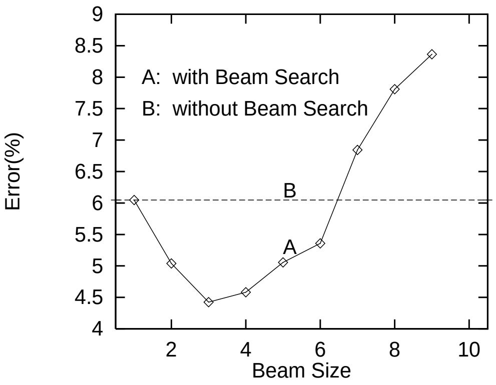

Figure 2: Error (%) vs beam size using the beam search technique ($\alpha = 0.1$, $L = 80$), compared to that using the original network.

The error shown in the figure is the averaged error of valid tours in all city distributions and is weighted by the percentage of valid tours for each city distribution:

$$Error = \frac{\sum_{i=1}^{N_{CityDist}} (Valid_i Error_i)}{\sum_{i=1}^{N_{CityDist}} Valid_i} \tag{12}$$

From the result in Fig. 1 and Fig. 2, we can see that the beam search is capable of improving the performance of the Hopfield network. The improvement is most significant when the beam size M is in the middle range between 3 to 7. The largest improvement for the percentage of valid tours is 24.2% (27.7% vs 22.3%) when $M = 5$, and for the error rate is 26.9% (4.42% vs 6.05%) when $M = 3$.

Fig. 3 and Fig. 4 show how the percentage of valid tours and error rate (%) vary with different beam intensities α when using the beam search technique (with fixed beam size $M = 4$ and beam search iteration $L = 80$), and compare them with those using the original Hopfield network. We can see that the beam search is able to improve the performance when the beam intensity α is between 0 and 0.2. The largest improvement for the percentage of valid tours is 17.0% (26.1% vs 22.3%) when $\alpha = 0.1$, and for the error rate is 26.0% (4.48% vs 6.05%) when $\alpha = 0.15$. The performance of the beam search starts decreasing as the beam intensity becomes larger.

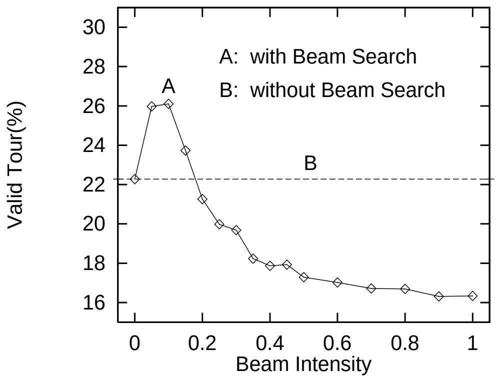

Figure 3: Percentage of valid tours vs beam intensity using the beam search technique ($M = 4$, $L = 80$), compared to that using the original network.

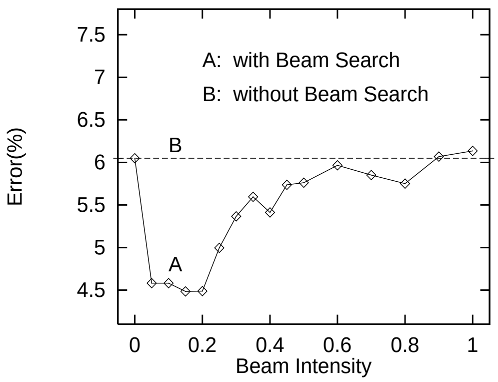

Figure 4: Error (%) vs beam intensity using the beam search technique ($M = 4$, $L = 80$), compared to that using the original network.

Fig. 5 and Fig. 6 show how the percentage of valid tours and error rate (%) vary with different beam iterations L when using the beam search technique (with fixed beam size $M = 4$ and beam intensity $\alpha = 0.1$), and compare them with those using the original Hopfield network. The beam search improves the performance when the beam iteration L is larger than 40. The optimal range for L is between 50 and 100 when considering both $Valid$ and $Error$. The largest improvement for the percentage of valid tours is 17.0% (26.1% vs 22.3%) when $L = 80$, and for the error rate is 37.0% (3.81% vs 6.05%) when $M = 120$. As L becomes too large, $Valid$ becomes smaller while $Error$ keeps approximately unchanged.

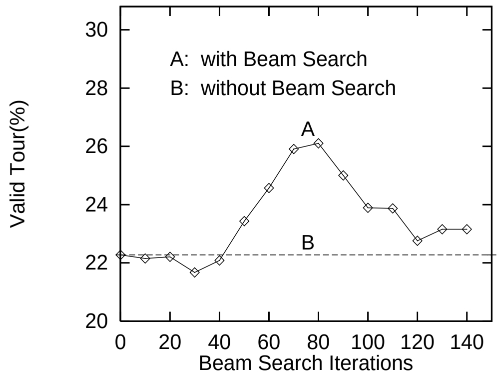

Figure 5: Percentage of valid tours vs beam search iteration using the beam search technique ($M = 4$, $\alpha = 0.1$), compared to that using the original network.

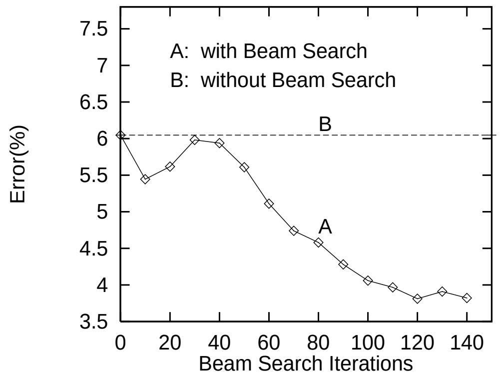

Figure 6: Error (%) vs beam search iteration using the beam search technique ($M = 4$, $\alpha = 0.1$), compared to that using the original network.

The above results (Fig. 1 through Fig. 6) show significant performance improvement by the beam search technique within a reasonable range of parameter settings. Also note that the amount of improvement depends on the choice of parameter setting – if they are not properly set (too small or too large), the performance can be negatively affected. For example, when beam intensity α becomes too large, $Valid$ starts decreasing. One reason is that if α is too large, the activations of multiple neurons becomes too close so that it could reach a tour that still keeps multiple neurons activated after the iteration timeout (which is counted as an invalid tour). We observed a 14.6% increase in the average converging iterations (156.2 to 179.0) for valid tours when α is increased from 0.1 to 0.25. The decreasing pattern on $Valid$ (due to increased α) matches approximately the increasing pattern on the average converging iterations. Another effect of a large α value is that it can distort much of the Hopfield network's dynamics itself, which could affect the quality of converged solutions. When the parameter settings are within a reasonable range, the network's performance can improve because it takes advantage of the better searching capability offered by using the beam search, while keeping negative effects to a minimum.

The results also show that some parameter settings favor improvement on $Valid$ while others favor improvement on $Error$. The choice of the setting needs to balance these two goals. We can give a quantified estimation on the improvement of the beam search by choosing an approximately balanced setting: $M = 4$, $\alpha = 0.1$, $L = 80$. It gives overall improvement of 17.0% (26.1% vs 22.3%) for $Valid$ and 24.3% (4.58% vs 6.05%) for $Error$. These results demonstrate that the beam-search is capable of improving performance over a reasonable range of parameter settings.

5 Summary

In this paper, we have proposed a beam search technique for the Hopfield network to improve its relaxation process. The beam search readjusts neuron activations to keep multiple neurons at a similar level. This provides a chance for the network to have a wider searching range for more and potentially better solutions. We have evaluated the performance of the beam search and its behavior in terms of the effects of different beam size, beam intensity, and beam search iterations. The evaluation was based on a large number of simulations (20,000 for each parameter setting) using 200 randomly generated city distributions of the 10-city TSP. The result demonstrates that the beam search has the capability of achieving a 17.6% increase in the percentage of valid tours and a 24.3% decrease in error rate.

In future we plan to extend this research in the following directions: (i) apply an adaptive mechanism to adjust the beam size, beam intensity, and beam search iterations during the relaxation process (instead of using fixed values as in this work); (ii) evaluate its performance on other optimization problems to obtain a better understanding of its behavior and its generality in applications.

6 Acknowledgments

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

7 References

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

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

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

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

[5] S. Z. Li, "Improving Convergence and Solution Quality of Hopfield-Type Neural Networks with Augmented Lagrange Multipliers", *IEEE Transactions On Neural Networks*, 1996, vol. 7, no. 6, pp. 1507-1516.

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

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

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

[9] X. Zeng and T. R. Martinez, "A New Activation Function in the Hopfield Network for Solving Optimization Problems", *Proceedings of International Conference on Artificial Neural Networks and Genetic Algorithms*, 1999, pp. 67-72.

[10] X. Zeng and T. R. Martinez, "A New Relaxation Procedure in the Hopfield Network for Solving Optimization Problems" *Neural Processing Letters*, 1999, vol. 10, pp. 1-12.