Optimization by Varied Beam Search in Hopfield Networks

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 - This paper shows that the performance of the *Hopfield network* for solving optimization problems can be improved by a varied beam search algorithm. The algorithm varies the beam search size and beam intensity during the network relaxation process. It consists of two stages: increasing the beam search parameters in the first stage and then decreasing them in the second stage. The purpose of using such a scheme is to provide the network with a better chance to find more and better solutions. A large number of simulation results based on 200 randomly generated city distributions of the 10-city *traveling salesman problem* demonstrated that it is capable of increasing the percentage of valid tours by 28.3% and reducing the error rate by 40.8%, compared to the original Hopfield network.

I. INTRODUCTION

The *Hopfield network* [1] provides a neural network approach for optimization problems. Its architecture consists of a two-dimensional connected neural network in which the linking strengths between neurons are determined based on the constraints and solution criteria of the optimization problem to be solved. A solution is achieved after the network is relaxed and reaches a stable state. This procedure was demonstrated by solving the *traveling salesman problem* (TSP) [1]. Since then there has been increased interest in the Hopfield network from various research fields. Much research work has been devoted to studying the theoretical aspects of the model [2]-[11].

However, there are still some basic problems with the Hopfield network as a tool for solving optimization problems. Wilson and Pawley [2] reported that the Hopfield model often generated invalid or poor solutions when tested on a wider range of TSP instances. In the past decade, many efforts have been made to improve the performance of the Hopfield model in terms of validity and quality of solutions. Brandt et al. [3] and Aiyer et al. [4] modified the energy function to enforce stricter constraints for valid solutions to improve 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 progress, the Hopfield model is still not yet efficient and robust, and it needs more improvements before being successfully applied to a wider spectrum of optimization problems. One problem is the inconsistent performance, which is usually better for simple city distributions, but poor for those with complex topologies and multiple clusters. Another problem is that the performance is sensitive to the choice of parameters in the energy function. 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 performance based on a large number of simulations.

In this work we propose a *varied beam search* procedure which dynamically varies beam search parameters to achieve a smoother searching process so that the network can find more and better solutions. It overcomes the drawbacks of a previously proposed *fixed beam search* [11] while still keeping beam search's advantages that allow the network to explore a wider spectrum of potential solutions. The varied beam search enables a smooth transition between different searching phases by gradually increasing the beam search parameters (size and intensity) in the early stage and then smoothly decreasing them (instead of using a fixed set of parameters through the whole process). This approach has been tested with a large number ($20,000$ per parameter setting) of simulations based on 200 randomly generated city distributions of the 10-city *TSP*. The results indicate that it is capable of improving performance significantly: 28.3% increase in the percentage of valid tours and 40.8% decrease in error rate, compared to the original Hopfield network. It can also significantly outperform the fixed beam search.

II. BASICS OF HOPFIELD NETWORK

The Hopfield network [1] consists of a two-dimensional linked neuron matrix. For an N-city *TSP*, there are

$N \times N$ 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 determines 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 bias: $I_{Xi} = CN_0$. 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.

III. VARIED BEAM SEARCH

The beam search with fixed parameters [11] keeps the top M 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.

If $V_{X_1 i}, V_{X_2 i} ... V_{X_M i}$ denote the top M activated neurons in column i, then their average activation $\overline{V}_i$ can be calculated using

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

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 (*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 $\alpha = 1.0$, $V'_{X_s i} = \overline{V}_i$, there 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 $\overline{V}_i$, there 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 beam search stops after L iterations and the network continues to relax normally until converging.

In the fixed beam search [11], the values of beam size M and beam intensity α are fixed through the whole beam search process. Although it has been shown to improve performance compared to a network without beam search, it has the following drawbacks: (i). It does not allow a smooth transition between beam search and non-beam-search phases. (ii). If M and α are too large, the beam search effect could be strong enough in the early relaxation phase to overshadow the network dynamics itself, and thus affect the formation of good solution prototypes. Also large parameters at the end of relaxation will cause a smaller convergence rate because it can weaken previously formed good solution prototypes. (iii). If M and α are too small, only a small fraction of the positive effects of beam search can be included in the relaxation process.

The purpose of the proposed varied beam search is to overcome the above drawbacks and yet keep the advantages of the beam search by a more flexible parameter setting scheme. It allows beam size and intensity to vary during relaxation. The parameters increase gradually to avoid too much deformation on the network dynamics in the early phase of relaxation. They also decrease smoothly during the transition from beam search to non-beam-search phases in the later beam search phase.

In this approach, beam search phases are characterized by the following beam search iteration parameters: L_i (initial), L_m (middle), L_f (final), with the constraint $L_i < L_m < L_f$. The corresponding values for beam size are M_i, M_m, and M_f; and those values for beam intensity are α_i, α_m, and α_f.

Beam search starts at iteration $L = L_i$, with beam size $M = M_i$ and intensity $\alpha = \alpha_i$. Then as L increases, both M and α increase linearly until $M = M_m$ and $\alpha = \alpha_m$ when L reaches L_m. This is the first phase, or *increasing phase*. From the point $L = L_m$, both M and α start decreasing linearly until $M = M_f$ and $\alpha = \alpha_f$ when $L = L_f$. This is the second phase, or *decreasing phase*. The values of these parameters are determined empirically through a large number of simulations, as will be shown in the next section.

The following is the formula of calculating M for a given L.

$$M = M_i + \frac{(L - L_i)(M_m - M_i)}{(L_m - L_i)} \qquad (L_i \leq L \leq L_m)$$

$$M = M_m + \frac{(L - L_m)(M_f - M_m)}{(L_f - L_m)} \qquad (L_m < L \leq L_f) \quad (8)$$

A similar formula for α is given by

$$\alpha = \alpha_i + \frac{(L - L_i)(\alpha_m - \alpha_i)}{(L_m - L_i)} \qquad (L_i \leq L \leq L_m)$$

$$\alpha = \alpha_m + \frac{(L - L_m)(\alpha_f - \alpha_m)}{(L_f - L_m)} \qquad (L_m < L \leq L_f) \quad (9)$$

During the beam search ($L_i \leq L \leq L_f$), M and α are updated after each iteration using formulas (8) and (9). All neuron activations in the network are then adjusted using formula (7). All the other parameters, formulas and procedures described in section II are kept exactly the same except for replacing $V_{X_s i}$ by $V'_{X_s i}$.

IV. SIMULATION RESULTS

The performance of the proposed varied beam search has been evaluated through a large number of simulations based on 200 randomly generated 10-city *TSP* city distributions (including wide varieties of topologies). Using a large number of city distributions can draw a more reliable conclusion when comparing two algorithms, compared to previous studies (e.g., [1], [2]) using a small number (10 or fewer) of city distributions. The reason is that the performance of an algorithm usually depends on the topology in a city distribution, and different algorithms may favor different topologies. Using a large number of city distributions reduces this effect and allows a better evaluation.

In our simulation, 100 runs were conducted for each of the 200 city distributions. For each of the 100 runs, different random noise was added to the initial neuron activations. For each fixed set of parameters (dt, u_0, L_m, M_m, α_m, *etc.*), the quantities to be evaluated (e.g., percentage of valid tours, error rate) are first averaged over 100 runs for each city distribution, and then averaged over the entire 200 city distributions. Thus for each parameter setting, $20,000$ runs are needed to obtain a simulation result. 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.

We used the original energy function of the Hopfield network in the simulation. All 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. 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.

The performance of varied beam search was evaluated by comparing it to the original Hopfield network (without beam search) and the fixed beam search method [11]. We have experimented with a large number of different combinations of parameter settings to evaluate the impact of each parameter on performance. Due to the large number of parameters (9 parameters in formulas (8) and (9)) and many more possible combinations, we focused on those parameters which have the most impact on performance.

Fig. 1 and Fig. 2 show how the percentage of valid tours and error rate (%) vary with different middle beam sizes M_m, and compare them to the original Hopfield network (no beam search) and the fixed beam search. Other parameters are fixed at $L_i = 0$, $L_m = 100$, $L_f = 200$, $M_i = M_f = 1$, $\alpha_i = 0.0$, $\alpha_m = 0.1$, $\alpha_f = 0.0$ (The reason for choosing these values is because they gave an approximate overall best results in terms of both quality and convergence after trying a large number of combinations).

The beam search starts at the beginning of relaxation ($L_i = 0$), with initial beam size $M_i = 1$ and initial beam intensity $\alpha_i = 0.0$. As iteration L increases, both beam size M and beam intensity α increase linearly until $M = M_m$ and $\alpha = 0.1$ when $L = 100$. After this point, both M and α decrease linearly until $M = 1$ and $\alpha = 0.0$ when $L = 200$.

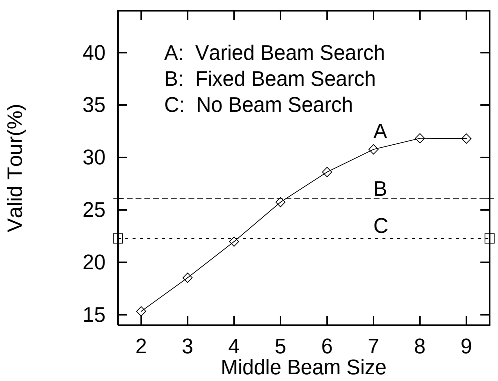

Figure 1: $Valid$ vs middle beam size M_m using varied beam search technique ($M_i = M_f = 1$), compared to that without using beam search and using fixed beam search.

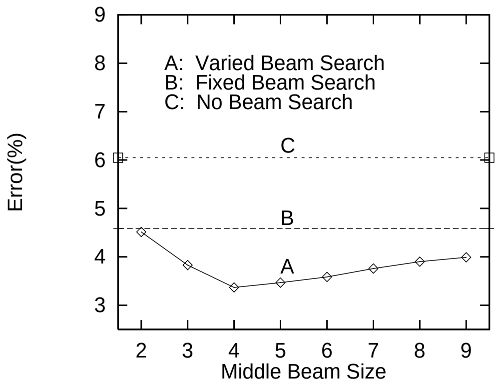

Figure 2: $Error$ vs middle beam size M_m using varied beam search technique ($M_i = M_f = 1$), compared to that without using beam search and using fixed beam search.

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:

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

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

$$Valid_i = \frac{N_{valid,i}}{N_{total,i}} \tag{11}$$

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

$$Error_{i,j} = \frac{d_{i,j} - d_{i,optimal}}{d_{i,optimal}} \tag{12}$$

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:

$$Error_i = \frac{\sum_{j=1}^{N_{valid,i}} Error_{i,j}}{N_{valid,i}} \tag{13}$$

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:

$$Error = \frac{\sum_{i=1}^{N_{CityDist}} (Valid_i Error_i)}{\sum_{i=1}^{N_{CityDist}} Valid_i} \tag{14}$$

The result in Fig. 1 and Fig. 2 show that varied beam search is capable of improving the performance of the Hopfield network. The percentage of valid tours increases with M_m. After $M_m > 5$, it outperforms both "B" and "C". The improvement on the error rate is uniform across all M_m range, and is more significant when $M_m \geq 3$. The largest improvement for $Valid$ is 42.6% compared to "C" (31.8% vs 22.3%) and is 21.8% compared to "B" (31.8% vs 26.1%) at $M_m = 8$. The largest improvement for $Error$ is 44.3% compared to "C" (3.37% vs 6.05%) and is 26.4% compared to "B" (3.37% vs 4.58%) at $M_m = 4$.

Fig. 3 and Fig. 4 show how $Valid$ and $Error$ vary with middle beam intensity α_m (with $\alpha_i = \alpha_f = 0.0$ and all other parameters being the same as for Fig. 1 and Fig. 2 except $M_m = 6$). Fig. 3 shows that $Valid$ is improved when α_m is in the range from 0.05 to 0.12. As α_m becomes larger, $Valid$ drops dramatically ($Valid$ values for $\alpha_m > 0.15$ are too small and not shown in Fig. 3). The improvement on $Error$ is more uniform and stable in a wide range. But if α_m is too large (> 0.275), the performance on $Error$ becomes worse. The largest improvement for $Valid$ is 30.0% compared to "C" (29.0% vs 22.3%) and is 11.1% compared to "B" (29.0% vs 26.1%) at $\alpha_m = 0.075$. The largest improvement for $Error$ is 59.5% compared to "C" (2.45% vs 6.05%) and is 46.5% compared to "B" (2.45% vs 4.58%) at $\alpha_m = 0.275$.

Fig. 5 and Fig. 6 show how $Valid$ and $Error$ vary with beam iterations L_f (with $L_i = 0$, $L_m = L_f/2$ and all other parameters being the same as for Fig. 1 and Fig. 2 except $M_m = 6$). $Valid$ is improved for a wide L_f range. It starts dropping slowly when L_f increases from 250. $Error$ is more sensitive to L_f than $Valid$. It decreases when L_f is changed from 100 to 150 and then starts increasing. It is smaller than "B" and "C" until $L_f = 400$. The largest improvement for $Valid$ is 28.3% compared to "C" (28.6% vs 22.3%) and is 9.58% compared to "B" (28.6% vs 26.1%) at $L_f = 200$. The largest improvement for $Error$ is 44.1% compared to "C" (3.38% vs 6.05%) and is 26.2% compared to "B" (3.38% vs 4.58%) at $L_f = 150$.

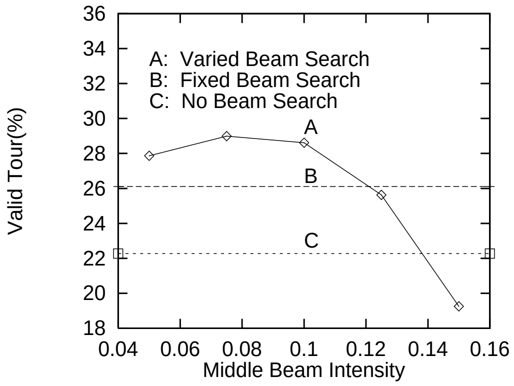

Figure 3: $Valid$ vs middle beam intensity α_m using varied beam search technique ($\alpha_i = \alpha_f = 0.0$), compared to that without using beam search and using fixed beam search.

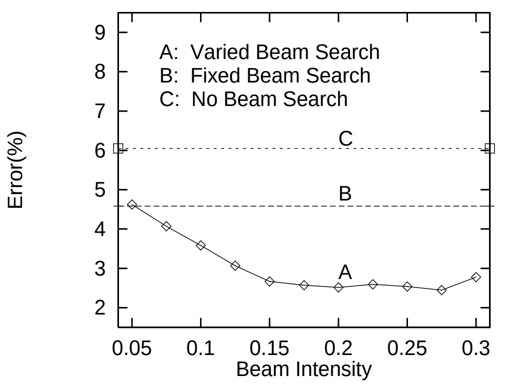

Figure 4: $Error$ vs middle beam intensity α_m using varied beam search technique ($\alpha_i = \alpha_f = 0.0$), compared to that without using beam search and using fixed beam search.

The results in Fig. 1 - Fig. 6 show that some parameter settings favor improvement on $Valid$ while others favor $Error$. The overall performance should consider both measurements with a balanced parameter setting. A quantified estimation on the improvement of varied beam search can be given by choosing an approximately balanced setting: $M_m = 6$, $\alpha_m = 0.1$, $L_f = 200$. For this setting, $Valid$ increases by 28.3% (28.6% vs 22.3%) and $Error$ decreases by 40.8% (3.58% vs 6.05%), compared to that without using beam search. As compared to the fixed beam search, increases $Valid$ increases by 9.58% (28.6% vs 26.1%) and $Error$ decreases by 21.8% (3.58% vs 4.58%).

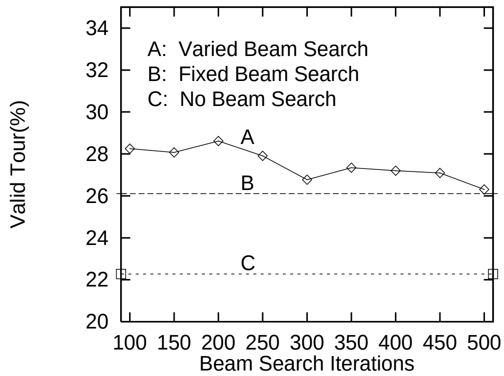

Figure 5: $Valid$ vs beam search iterations L_f using varied beam search technique ($L_i = 0$, $L_m = L_f/2$), compared to that without using beam search and using fixed beam search.

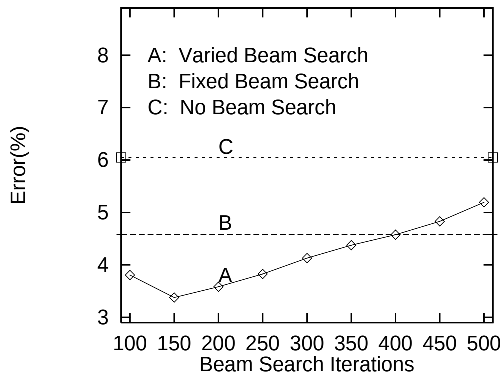

Figure 6: $Error$ vs beam search iterations L_f using varied beam search technique ($L_i = 0$, $L_m = L_f/2$), compared to that without using beam search and using fixed beam search.

The above results show significant performance improvement by using varied beam search within a reasonable range of parameter settings. Its flexible scheme for adjusting parameters enables it to overcome the drawbacks in fixed beam search while keeping the advantages of beam search. Also note that the amount of improvement depends on the choice of parameter setting. For example, if middle beam size M_m becomes too small, $Valid$ starts dropping (as shown in Fig. 1) because only a small amount of the beam search's positive effects is added to the network. $Valid$ also becomes smaller if middle beam intensity α becomes too large (as shown in Fig. 3). One reason is that if α is too large, the activations of multiple neurons becomes too close so that the network could reach a tour that still keeps multiple neurons activated after the iteration timeout (which is counted as an invalid tour). Another negative effect of a large α value is that it can distort much of the Hopfield network's dynamics and thus affect the quality of converged solutions. If the parameters are set within a reasonable range, varied beam search is able to improve the network performance because it takes advantage of the better searching capability offered by beam search with flexibility while minimizing the potential drawbacks.

V. SUMMARY

In this paper, we have proposed a varied beam search approach for the Hopfield network to improve its relaxation process. This approach dynamically adjusts the beam size and beam intensity during beam search. It first sets those parameters at small values, and then gradually increases them as the relaxation proceeds. During the later stage of the beam search, those parameters are decreased gradually to make a smooth transition to the non-beam-search phase. This approach is able to overcome the drawbacks of fixed beam search while keeping the advantages of beam search. The performance of varied beam search and its dependence on middle beam size, middle beam intensity, and beam search iterations were evaluated using a large number of simulations (20,000 for each parameter setting) based on 200 randomly generated city distributions of the 10-city TSP. The results show that the varied beam search is capable of achieving 28.3% increase in the percentage of valid tours and 40.8% decrease in error rate, compared to the original Hopfield network without using beam search. It also outperformed the fixed beam search, with a 9.58% increase in the percentage of valid tours and a 21.8% decrease in error rate.

VI. ACKNOWLEDGMENTS

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

VII. REFERENCES

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

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

[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., II: 333-340, 1988.

[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*, Vol. 1, No. 2, pp. 204-215, 1990.

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

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

[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*, Vol. 62, pp. 129-139, 1989.

[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*, pp. 67-72, 1999.

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

[11] X. Zeng and T. R. Martinez, "Improving the Hopfield Network through Beam Search", *Proceedings of IEEE International Joint Conference on Neural Networks*, pp. 1162-1167, 2001.