Hopfield Nets with Time-varying Energy Functions for Solving the Travelling Salesman Problem

Sheng-De Wang and Chii-Ming Tsai
Department of Electrical Engineering
National Taiwan University
Taipei 10764, Taiwan
E-mail: sdwang@twmoe10.bitnet

ABSTRACT A new method to solve the Travelling Salesman Problem (TSP) with Hopfield model is introduced in this paper. The method uses a time-varying energy function to enhance performance, instead of the simulated annealing approach of adding a random noise to the input of each neuron. The proposed energy function represents TSP precisely and controls the process of convergence strictly. The method has obtained 100% optimal solutions within 300 iterations to 10 city problem whose map is the same as that Hopfield and Tank [1] used, and has also been proven very efficient for other sets of 10 city problems whose cities are randomly and uniformly distributed within a unit square. Furthermore, some near optimal solutions have also been found to larger scale problems under the same circumstances. Such excellent simulation results reveal that the new energy function indeed greatly improves traditional energy functions.

1 Introduction

The model of artificial neural network has been studied for many years, but it wasn't applied to optimization problems until Hopfield [2] presented conditions under which the network can converge to a stable point corresponding to a local minimum of the energy function. The travelling salesman problem (TSP) has become a typical combinatorial problem in the field of neural network since Hopfield and Tank [1] proposed a neural network method for solving the problem.

One important step to solve TSP using Hopfield nets approach is to look for an energy function which represents TSP better. Many papers about this topic have been presented [3]-[5]. All these energy functions are composed of two components: the penalty for an invalid tour and the length cost of a tour. The former forces the network to converge to a valid solution which is a permutation matrix to our problem, and the latter is the goal of optimization. However, how to balance these two items is an important but difficult task. This is an inherent and inevitable drawback to solve TSP through Hopfield net with an energy function. That is, the solutions are very sensitive to the relative weights between these two items, as can be observed in simulations results.

Furthermore, another difficulty to solve optimization problems is to prevent the network from converging to a local minimum. In fact, many efforts have been made to overcome this problem, such as stochastic neural networks [6]. The use of simulated annealing in Hopfield nets is the most successful one, but it takes too much time to find the global minimum.

This paper therefore proposes a time-varying energy function whose parameter is time-varying to improve performances. Of course, the parameter reaches a stable value finally. The proposed energy function has been proved to be very efficient as demonstrated by the excellent simulation results. The mathematical properties of the energy function are also discussed in this paper.

2 Modified Hopfield Model

Hopfield model is the most well-known neural model to solve optimization problems. The model proposed by Hopfield and Tank [1] is a continuous-time model analog to a neural circuit. The time evolution of the circuit is described by

$$C_i \frac{du_i}{dt} = \sum_{j=1}^n T_{ij} V_j - \frac{u_i}{R_i} + I_i$$

Our neural model is also based on the above model. For the purpose of computer implementation, we approximate the above model with first-order numerical integration method and the time step used to solve the differential equation is assigned to 1 to speed up convergence. Hence the model we proposed is actually a discrete-time model each neuron with graded response as described by the following equations:

$$U_{i,t+1} = \sum_{j=1}^n T_{ij} V_{j,t} + I_i \quad (1-a)$$

$$V_{i,t+1} = f(U_{i,t+1}) \quad (1-b)$$

where T_{ij} denotes the connection weights from neuron i to j , and $U_{i,t}$ and $V_{i,t}$ are the activation (i.e., the internal value) and the output of neuron i at time t , respectively. I_i represents the external bias input for neuron i . In the case of discrete-time model, the order of activating neurons has great influences on the results since the neural dynamics are evolved with time step size 1 which is extremely large as compared to continuous-time model. Therefore, the neuron update order is randomly determined to hold generality.

The transfer function f of each neuron is a sigmoid function as described in (2-a), but the parameter μ is no more a constant than a decreasing value shown in (2-b), where μ_0 is the initial value and τ_0 controls speed of decreasing.

$$f(U) = \frac{1}{2} \left(1 + \tanh\left(\frac{U}{\mu}\right) \right) \quad (2-a)$$

$$\mu = \frac{\mu_0}{1 + t/\tau_0} \quad (2-b)$$

3 Travelling Salesman Problem

The Travelling Salesman Problem (TSP) is a classical NP-complete combinatorial optimization problem. The problem is to find a closed tour which visits each city once. For an n city problem, n^2 neurons are needed to represent it, and the representation of a valid tour must be a permutation matrix.

Let V_{xi} be the output of the neuron which denotes the x -th city is located in the i -th position of the tour, and $d_{x,y}$ is the distance between the x -th city and the y -th city. Then our energy function is shown as equation (3), where S is the column sum and row sum of the outputs of the neurons.

$$\begin{aligned} E = & \frac{A}{2} \sum_{x=1}^n \left(\sum_{i=1}^n V_{xi} - S \right)^2 + \frac{B}{2} \sum_{i=1}^n \left(\sum_{x=1}^n V_{xi} - S \right)^2 \\ & + \frac{C}{2} \sum_{x=1}^n \sum_{y=1}^n \sum_{i=1}^n d_{x,y} V_{xi} (V_{y,i+1} + V_{y,i-1}) \\ & + \frac{A+B}{2} \sum_{x=1}^n \sum_{i=1}^n V_{xi} (1 - V_{xi}) \end{aligned} \quad (3)$$

Derived from the energy function, the weights between neurons and the bias of each neuron are shown in equations (4-a) and (4-b) respectively.

$$T_{xi,yj} = -A \delta_{x,y} (1 - \delta_{i,j}) - B \delta_{i,j} (1 - \delta_{x,y}) - C \delta_{x,y} (\delta_{j,i+1} + \delta_{j,i-1}) \quad (4-a)$$

$$I_{xi}(S) = (A+B) \left(S - \frac{1}{2} \right) \quad (4-b)$$

The first two terms in (3) lead the outputs of the neurons to a permutation matrix if $S=1$. The third term takes into account the distance of a tour. The last term seems to be a redundant one, but it is necessary for our energy function. First, the term increases the energy of points inside the n^2 dimensional hypercube and keep points in the corners unchanged. This is helpful to find a valid solution because it must be in some corner. Second, it eliminates the self-feedback link of each neuron, and enhances the stability of the network.

An important feature of our energy function is that the parameter S is not a fixed value, but a time dependent parameter. Initially, S is assigned to such a high value that all neurons are forced to be "on" independent of their initial conditions. As time goes on, the S is gradually decreased and some neurons will be turned "off" and all connections which correspond to these neurons will be cut off simultaneously. Finally, S reaches a stable value such as 1, then a valid solution will be obtained.

The time-varying feature of S can be in some sense related to the concept of system entropy. The heuristic entropy concept can be referred to as a measure of uncertainty. At the beginning of searching an optimal tour, each neuron in the network are hardly to know when should be "on" or "off". A high value of S would let each have high opportunity to be "on" and thus the solution path would be in a chaotic situation, corresponding to high entropy state. As the system dynamics settling down, the outputs of neurons could enter into a certain state which represents a good solution, which may be expected to be an optimal solution, since the final state is gradually refined from a large set of initial states.

On the other hand, why such a time-varying S can work well can be answered from the mathematical point of view. From the viewpoint of mathematical description, the gradient shown in equation (5) of the energy function is dependent on S due to $I_{xi}(S)$ in (4-b).

$$\frac{\partial E}{\partial V_{xi}} = - \left(\sum_{y=1}^n \sum_{j=1}^n T_{xi,yj} V_j + I_{xi}(S) \right) \quad (5)$$

It is noted that the weight matrix $T_{xi,yj}$ is not dependent on S as can be seen from (4-a). Hence, the direction toward which the energy function evolves is a compromise between the two terms $\sum_{y=1}^n \sum_{j=1}^n T_{xi,yj} V_j$ and $I_{xi}(S)$. At the beginning, a larger value of S will dominate the energy gradient in (5). Furthermore, S is assumed to be decreased very slowly such that the energy is decaying in an almost constant rate or a smooth way that can avoid being trapped in a local minimum, since a trapping in local minimum means there must have a somewhat abrupt change of energy. Therefore the system won't be trapped into a local minimum in the beginning, and a better solution can be obtained later. If S is gradually decreased from a high value so that the column sum and the row sum of the neurons can follow the change of S in time, then the system is strictly controlled during the process of convergence.

A suggested schedule for S is an exponential function as (6),

$$S = 1 + K \exp(-\alpha t) \quad (6)$$

where t is the current iteration number, K is an initial value, and α determines the speed of decrease.

4 Simulation Results

There are totally seven undetermined parameters in our model and energy function : $A, B, C, K, \alpha, \mu_0$, and τ_0 . A and B are symmetric, and therefore can be arbitrarily assigned to 1. The value of C is dependent on the problem size. It is usually assigned to 1 if all cities are located within a unit square. Simulation results are very sensitive to the value of C . Using a smaller C to perform simulations will get higher valid rate, but the average length of all found tours is longer than that gotten by using a larger C .

The assignments of other parameters are shown as follows:

$$K=20 \quad \alpha = 0.015 \quad \mu_0 = 1.1 \quad \tau_0 = 100$$

All these values are not very sensitive and can be selected easily. For example, the same simulation results can be obtained with larger K and smaller α , or larger μ_0 and τ_0 . Of course, there exist a lower bound or upper bound for each parameter, and it must be satisfied to ensure the performances.

Now let's take the same map that Hopfield and Tank used before as an example. If S is fixed at 1, only 4 trials are optimal and 82 trials are valid tours. If we set $S=1.577$ as Akiyama *et. al.* [3] suggested, 46 trials are optimal and another 46 trials are other valid tours in 100 trials. The results are improved a little. However, the optimal tour is found in every one of 100 trials if a time-varying S as (6) is used. And less than 300 iterations are needed to find the solution. Obviously, the performances are promoted greatly. Detailed comparisons are shown in Fig-1 to compare the effects of various types of S . Fig-2 shows the evaluated energies of three types of S during the process of convergence.

Fig-1 (a) is the solved map; (b) is the comparison of effects of three types of S .

Our simulation study show that not only the example mentioned above but also many other maps can be solved well. All maps in Fig-3 are examples whose simulation results are 100% optimal, too. Though the same results are not obtained for every map, more than 80% trials are predicted to be optimal if the cities are uniformly distributed in a unit square.

Fig-2 Evaluated energies of three types of S.

Fig-3 All maps in the figure get 100% optimal solutions in simulation.

Fig-4 Some discovered tours to 20 city TSP problem. (a) is the best; (b) is the worst; (c) is the second best; (d) is a typical tour near average.

Larger scale problems are also solved well. After properly scaling up the parameters used: $K=40$, $a=0.01$, $m_0=2.2$, and $t_0=200$, some reasonable tours are also found to 20 city problems. In 100 trials, 98% trials converge to a valid tour. The length of the best tour in Fig-4(a) is 4.0265 and the length of the worst tour is 4.456 in Fig-4(b). Fig-4(c) is the second best tour whose length is 4.105. The average length of all valid tours is 4.3619 and Fig-4(d) is a typical tour whose length is 4.312 near average. All tours are found within 1000 iterations.

5 Conclusion

Optimization is an important application of neural networks. Usually, people construct a network according to a fixed-parameter energy function and try to add a random noise to the input of each neuron. There are various stochastic approaches available to achieve high performances. However, from the viewpoint of mathematical analysis, the global minimum will be obtained if quite a long time is spent. This paper proposes a new energy function which introduces a time-varying column sum and row sum S without adding any random noise. Though the parameter of the energy function is dependent on time, the weights between neurons are always fixed. The parameter S affects the bias of each neuron only.

The new energy function has several advantages over traditional energy functions. Applying it to TSP, we have obtained very excellent simulation results. The energy function doesn't suit for all optimization problems, but it can work well for some combinatorial problems such as Job-scheduling problem, n-Queen problem, and TSP because all their optima are all represented as permutation matrices.

References

- [1] J.J. Hopfield and D.W. Tank, "Neural Computation of Decisions in Optimization Problems," *Bio-Cybern*, Vol.55, pp. 141-152, 1985.
- [2] J.J. Hopfield, "Neurons with graded response have collective computational properties like those of two-state neurons," *Biophysics, Proc. Natl. Acad. Sci. USA*, 81, pp.3088-3092, 1984.
- [3] Y. Akiyama, A. Yamashita, M. Kajiura, and H. Aiso, "Combinatorial Optimization with Gaussian Machines," *Proc. of the International Conference on Neural Networks*, Vol. 1, pp. 533-540, 1989.
- [4] S. Kirkpatrick, C.D. Gelatt, Jr. and M.P. Vecchi, "Optimization by Simulated Annealing," *Science*, Vol.220, NO.4598, pp. 617-680, May 13 1983.
- [5] H. Szu, "Fast TSP Algorithm Based on Binary Neuron Output and Analog Neuron Input Using the Zero-diagonal Interconnect Matrix and Necessary and Sufficient Constraints of the Permutation Matrix," *IEEE Second International Conference on Neural Networks*, Vol.2, pp.259- 266, 1988.
- [6] B.C.Levy and M.B. Adams, "Global Optimization with Stochastic Neural Networks," *Proc. 1st Int. Conf. Neural Networks*, San Diego, CA, June 1987.