Adaptive Coefficients for Hopfield Neural Networks Solving Combinatorial Optimization Problems
--최적화를 위한 홉필드 신경망의 적응적 신경계수 결정--

Chiyeon Park*
박 지 연
Kuinam J. Kim*
김 귀 남

Abstract

본 논문에서는 에너지 함수의 직접적인 평가에 기초해 홉필드 신경망을 진화시킴에 따라 적응적으로 에너지 계수를 결정하는 기법을 제시하고자 한다. 이 기법에 근거하여 구해지는 계수들의 효과를 검증하기 위해 응용 모델인 TSP(Traveling Salesman Problem)에 적용하여, 실험을 통한 계수 값의 변화 추이를 분석하고 그 결과를 기존의 기법들과 비교한다. 또한 제안된 방법에 필수적인 각 단계에서의 에너지 값의 평가를 위한 부가적인 연산을 줄이기 위해 단계적으로 증감분만을 계산하는 효율적인 연산법을 제시한다.

1. Introduction

Since Hopfield neural networks were first employed to solve combinatorial optimization problems, especially for the Traveling Salesman Problem(TSP) in the original article[7], many applications[8,11,13,16] have been found for this network because of its potential for parallel computation and computational advantage when it is implemented with analogue VLSI techniques. But when the Hopfield network with the original form of the energy function is used, many reports[8,15] have shown that the solutions obtained for the TSP were either invalid or of low quality. Some alterations of the energy formulation[1,3,4,8] have been suggested to overcome this difficulty, but none of them have shown great improvement.

Now most concern seems to be not with finding more promising energy formulations, but more robust ways of selecting proper coefficients for the terms in the energy function, which represent the constraints and objective function to be minimized, because of the extreme sensitivity of the network to the values of the parameters. To find a fundamental explanation for the behavior of the network, some theoretical investigations[2,14] were conducted by analyzing the eigenvalues of the weight matrix and Lagrange multipliers. While these investigations yielded satisfactory explanations for the behavior of the network in terms of the energy coefficients, they did not offer a practical method for computing effective values for the coefficients.

To alleviate the difficulty of finding suitable parameters for each term in the energy function, which seems to be another hard problem added to the original problem to be

*Department of Computer Science & Engineering, Kwandong University

This Research was supported by Kwandong University Research Grant 1997

solved, we propose a methodical procedure for determining the coefficients adaptively based on the energy values as the network evolves. By utilizing the energy function directly, we are able to control the amount which each term contributes to the total energy. The energy terms can be used to guide the network toward valid, high quality solutions, by properly balancing the coefficients of competing terms. This approach is applied to the TSP, which is known[9] to be among the harder problems to be solved by pure Hopfield type neural networks. In order not to increase the computational burden due to the evaluation of the energy of each term as the network progresses, an efficient method of incrementally updating the energy value of each term with minimal calculation is described. In a similar manner, the values of individual terms in other forms of Hopfield energy functions also can be easily calculated and updated at each iteration.

2. Energy Formulation and Dynamics of Hopfield Network

TSP is a well known example of a class of computationally hard problems[6] which requires impractical amounts of time (exponential computational complexity) to find an optimal solution as the problem size increases. In this problem, every city in a given set of N cities is to be visited once and only once. A tour may begin with any city, and ends by returning to the initial city. The goal for the solution of this problem is to find a tour which has the shortest possible length.

With a Hopfield network, this problem is represented by a N by N matrix of neurons in which a valid solution is achieved when the network reaches a state of a permutation matrix. One form of the energy function which has been suggested[12] is :

$$\begin{aligned} E= & \frac{C_1}{2}\sum_x\sum_i\sum_{j\neq i} V_{x,i}V_{x,j}+\frac{C_2}{2}\sum_i\sum_x\sum_{y\neq x} V_{x,i}V_{y,i} \\ & +\frac{C_3}{2}\sum_x\Big(\sum_i V_{x,i}-1\Big)^2+\frac{C_4}{2}\sum_i\Big(\sum_x V_{x,i}-1\Big)^2 \\ & +\frac{C_5}{2}\sum_x\sum_{y\neq x}\sum_i d_{x,y}V_{x,i}(V_{y,i+1}+V_{y,i-1}) \end{aligned} \tag{1}$$

In the network model, Indices x and y refer to cities, and i and j refer to positions of the cities in the tour. The detailed explanation can be found in other literatures[7,12].

When the energy is minimized, the network is expected to have reached a state corresponding to a valid solution of TSP. The implication of the minimization of the energy terms is as follows. The first two terms are inhibitory in nature while the next two are excitatory. All four constraints can be satisfied simultaneously when each row and column have a single neuron activated. These four terms together work to enforce the validity constraints of the problem. When the constraints of the TSP are satisfied by these terms, the fifth term gives the value of the corresponding tour length. This term represents the TSP objective function. Its value is to be made as small as possible while maintaining the validity of the tour.

To guarantee convergence of the network[5,12], the motion dynamics are obtained from :

$$\begin{aligned}\frac{du_{x,i}}{dt} &= -\frac{\partial E}{\partial V_{x,i}} \\ &= -C_1 \sum_{j \neq i} V_{x,j} - C_2 \sum_{y \neq x} V_{y,i} - C_3\Big(\sum_j V_{x,j} - 1\Big) \\ &\quad - C_4\Big(\sum_y V_{y,i} - 1\Big) - C_5 \sum_{y \neq x} d_{x,y}(V_{y,i+1} + V_{y,i-1})\end{aligned} \tag{2}$$

where $V_{x,i} = \frac{1}{2}\left[1 + \tanh\left(\frac{u_{x,i}}{u_0}\right)\right]$ (3)

For simulation with a digital computer, updating of each neuron usually follows Euler's first order difference equation :

$$u_{x,i}(t + \triangle t) = u_{x,i}(t) + \left(\frac{du_{x,i}}{dt}\right)\triangle t \tag{4}$$

A simulation is performed by initiating the neurons' activations with random values, then allowing the activations to evolve according to the governing equations for the network dynamics. The activations are iteratively updated until the total energy converges, and the status of the neurons in the network is inspected to determine whether or not a valid solution of TSP has been obtained.

It is easily seen that the difficulty of finding valid, high quality solutions in part due to the problem of selecting appropriate values for the coefficients of the energy components. The behavior of the network is highly sensitive to the values of these coefficients, previously[7,15] the determination of appropriate values was primarily done by experimental trial and error.

3. Hopfield Network with Adaptive Coefficients

3.1 Determination of Coefficients

As the network evolves in the direction of minimization of the total energy, each term in the energy function competes with the other terms to influence the path to be followed. The explicit way in which each term contributes to the total energy at each stage is seen by an analysis of the changes of the values of each energy term. In our approach, to find good coefficients for the energy function, the energy of each component is traced through each epoch of iteration. At the same time, it is utilized to control the effect of each term on the evolution of the network toward the point we hop to reach. In this way, according to the distance to the goal (minimal value) of each term, the corresponding coefficients are either relatively emphasized by increasing the value, or de-emphasized by decreasing the value, until a balanced relationship among the coefficients is reached.

The coefficients are updated after every epoch of iteration until they reach a state of near equilibrium in which all coefficients are relatively stable, thereafter competing with each other with balanced strength. Then the final coefficient values obtained are used to set weight connections between all neurons, and the network is run again to obtain the final solution. The overall algorithm is described as follows:

```
(a) initialize all energy coefficients to 1 ;
(b) calculate each energy term ;
(c) repeat until all coefficients are stabilized ;
        (c-1) update coefficients ;
        (c-2) run network one epoch updating neurons ;
        (c-3) update each energy term incrementally ;
```

After every epoch, the new coefficients are computed(c-1) depending on the current energy of each term. The new coefficients are computed as follows :

$$\begin{aligned} C_i^* &= \frac{\partial E_{total}(V)}{\partial C_i} \\ &= \frac{\partial[\sum_j C_j \cdot E_j(V)]}{\partial C_i} \\ &= E_i(V) \end{aligned} \tag{5}$$

where V is the vector of outputs of all neuron.

This is a steepest ascent procedure. It has the effect of maximizing the energy function with respect to its coefficients. While this may seem counter-intuitive at first, it has the desired effect of increasing the coefficients of those terms that are contributing the most to the value of the energy function. It is those terms that most need to be reduced during network iteration. Normalization is necessary in order for the coefficient values to stabilize, which simply is done by :

$$C_i = \frac{C_i^*}{\sum_j C_j^*} \tag{6}$$

3.2 Computation of Energy Update

Since the computational work required to evaluate directly the current energy level of each term of the energy function is somewhat expensive, an incremental method has been devised. It is performed during each update of the neurons. In this way, the new energy value can be easily computed without requiring a complete re-evaluation at every change of activations.

The change of energy when a neuron's output $V_{x,i}(t)$ is changed to $V_{x,i}(t+1)$ is

$$\begin{aligned} E(t+1) - E(t) = \ & \triangle V(C_1 \sum_{j \neq i} V_{x,j}) + \triangle V(C_2 \sum_{y \neq x} V_{y,i}) \\ & + \triangle V[C_3(\sum_j V_{x,j} - 1) + \frac{C_3}{2} \triangle V] \\ & + \triangle V[C_4(\sum_y V_{y,i} - 1) + \frac{C_4}{2} \triangle V] \\ & + \triangle V[C_5 \sum_{y \neq x} d_{x,y}(V_{y,i+1} + V_{y,i-1})] \end{aligned} \tag{7}$$

where $\triangle V = V_{x,i}(t+1) - V_{x,i}(t)$.

Since most of the terms in Eq.(7) are already computed when updating the neuron activation $u_{x,i}(t+1)$, such as in Eqs.(2) and (4), the only parts requiring additional computation are the ones outside the summations. In this way, once the total energy is computed at the initialization stage of the main procedure, the new energy value can be efficiently obtained after each update of neurons.

4. Simulation and Result

In our simulations, a 10 city TSP, which appeared in the original Hopfield paper, and a 20 city TSP (randomly generated), are simulated with different random initializations of neurons. The sigmoid gain function and neuron initialization as proposed by Hopfield[7] are used. Since the steepness of the gain function seems to be proportional to the size of time step(Δt) in our experiments, altering either one of them is enough to maintain network stability. In our studies, Δt is set to 0.01. The threshold for deciding neuron's on-off status is 0.5.

Figure 1 shows the change of each coefficient, with all coefficients being adapted, with normalization so that $C_1+C_2+C_3+C_4+C_5=1$. It is apparent that C_1 and C_2 tend toward a common value, as do C_3 and C_4 also. As seen from the graph, the coefficient of the objective term is relatively very large, because that term has a large portion of the total energy. This is due to the fact that the energy of that term (tour length of solution) cannot be made lower than a certain but unknown value corresponding to an optimal valid solution, unlike the other energy terms. The terms representing row and column constraints (1st through 4th terms) all go to zero for a valid solution. This can cause the network to reach invalid solutions by trying to reduce the objective function value below the minimal distance necessary for a solution. In order to overcome this situation, we must allow the network to maintain a reasonable distance energy. Thus the coefficient of the objective term may be specified before the network begins adaptive determination of the other coefficients. A reasonable value of C_5 is specified first, with some experimentation.

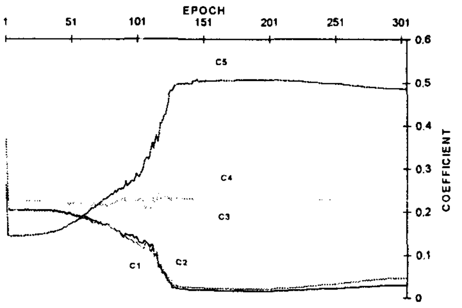

Figure 1. Coefficients (all adapted)

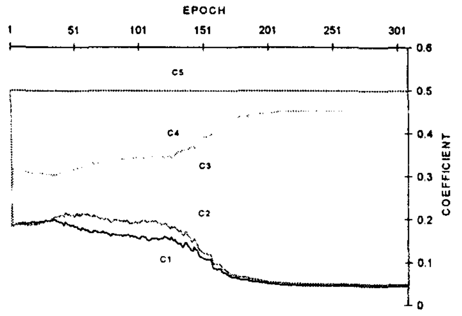

Figure 2. Coefficients (C_5 fixed)

Figure 2 depicts the change of all constraint coefficients with C_5 fixed, and $C_1+C_2+C_3+C_4=1$. Again we see that C_1 and C_2 tend toward a common value, as do C_3 and C4 also. In this way, the network found valid solutions more than 80% of the time. Some examples of the solutions produced are shown in Table 1. Since the final coefficients with this adaptive method are the ones which balance the network, the coefficients obtained are

Table 1. Solutions found during adaptation

case	10 cities	20 cities
1	3.08	5.76
2	3.38	6.81
3	2.85	4.59
4	3.16	5.48
5	3.45	*
6	3.23	*
7	3.34	6.44
8	3.02	5.26
9	*	5.22
10	2.98	5.15
* no valid solution found		

Table 2. Solutions using final values of coefficients

case	10 cities	20 cities
1	*	*
2	2.92	4.73
3	*	3.89
4	2.75	4.52
5	2.83	5.56
6	2.69	4.77
7	2.83	4.92
8	2.69	4.87
9	3.13	5.02
10	2.89	4.94
* no valid solution found		

used to rerun the network with the values of the coefficients being held fixed. Some examples of the solutions produced with these coefficients are shown in Table 2. For comparison, the same initialization with fixed values of coefficients as given by Hopfield[7] for 10 cities yielded 15 valid solutions out of 100 trials with average length of 3.40.

In more than 80% of the 100 trials, the network produced valid solutions and the quality of the solutions is much better than the solutions found during adaptation of the coefficients. Some of the solutions are the optimal solution, and the others found are also good solutions. Examples of solutions found are depicted in Figure 3.

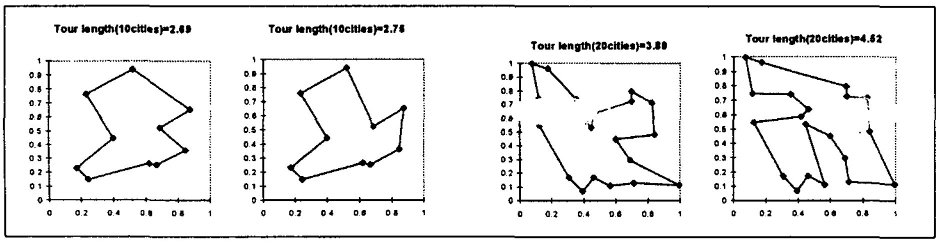

Figure. 3 Examples of Tours including the optimal solutions for 10 and 20 city TSP

By finding coefficient values adaptively, the quality of solutions using the same energy formulation are much better than those reported in the literature[10]. Tables 3 and 4 show the quality of the solutions using the following measure :

$$Quality = \frac{C_{random} - C_{sol}}{C_{random} - C_{opt}} \tag{7}$$

where C_{Random} is the average length of 10,000 random tours, C_{sol} is the average length of 100 tours found by different initializations of the Hopfield network, and C_{opt} is the optimal tour length.

Table 3. Quality of solutions found during adaptation

	10 cities	20 cities
C_{opt}	2.69	3.89
C_{random}	4.76	10.13
C_{sol}	3.16	5.58
Quality	0.77	0.73

Table 4. Quality of solutions found after adaptation

	10 cities	20 cities
C_{opt}	2.69	3.89
C_{random}	4.76	10.13
C_{sol}	2.84	4.80
Quality	0.93	0.86

5. Summary and Discussion

We have proposed a systematic way to determine properly balanced coefficients in a Hopfield network for solving TSP. Because the path of the network is guided by the energy terms, its coefficients are found adaptively by the network itself. In this way, we obtain good coefficients which help the network to find not only valid but also high quality solutions. In spite of the improvements over previous reports, there are some other issues to point out. To be more practical, the performance in the sense of solution quality needs to be improved to be more competitive with currently available problem-tailored methods for TSP. Also, as the problem size increases, the dimensionality requirement can be another problem to be solved.

References

[1] S.Abe, "Theories on the Hopfield Neural Networks", Proc. of IEEE JCNN, Vol. 1, pp. 557-564, 1989.

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

[3] A.R.Bizzarri, "Convergence Properties of a Modified Hopfield-Tank Model, Biological Cybernetics", Vol. 64, pp. 293-300, 1991.

[4] R.D.Brandt, Y.Wang, A.J.Laub, and S.K.Mitra, "Alternative Networks for Solving the Traveling Salesman Problem and List-Matching Problem", Proc. of IEEE ICNN, Vol. 2, pp. 333-340, 1988.

[5] L.Fausett, Fundamental of Neural Networks, Prentice Hall, 1994.

[6] M.R.Garey, and D.S.Johnson, Computers and Intractability : A Guide to the Theory of NP-Completeness, Freeman, 1979.

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

[8] A.Jagota, "Approximating Maximum Clique with a Hopfield Network", IEEE Trans. on Neural Networks, Vol. 6, No. 3, pp. 724-735, 1995.

[9] B.Kamgar-Parsi, and B.Kamgar-Parsi, "On Problem Solving with Hopfield Neural Networks", Biological Cybernetics, Vol. 62, pp. 415-423, 1990.

[10] W.Lin, J.G.Delgado-Frias, G.G.Pechanek, and S.Vassiliadis, "Impact of Energy Function on a Neural Network Model for Optimization Problems", Proc. of IEEE ICNN, Vol. 7, pp. 4518-4523, 1994.

[11] H.Shirai, A.Ishigame, S.Kawamoto, and T.Taniguchi, "A Solution of Combinatorial Optimization Problem by Uniting Genetic Algorithms with Hopfield's Model", IEEE ICNN, Vol 7, pp. 4704-4709, 1994.

[12] H.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", Proc. of IEEE ICNN, Vol 2, pp. 259-265, 1988.

[13] Y.Takefuji, Neural Network Parallel Computing, Kluwer Academic Publishers, 1992.

[14] J.Van den Berg, and J.C.Bioch, (1994). "Capabilities of the Symbiosis Between the Hopfield Model and Lagrange Multipliers in Resolving Constrained Optimization Problems", Proc. of Computing Science in the Netherlands, pp. 54-65, 1994.

[15] 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.

[16] W.J.Wolfe, "Hopfield Networks and Scheduling Problems", Proc. of SPIE, Vol. 2492, pp. 315-322, 1995.