A New Neuron Dynamics for Solving the Minimum Graph Bisection Problem

Rong-Long Wang, Yoshihiro Yamanishi, and Kozo Okazaki

Faculty of Engineering, University of Fukui, Bunkyo 3-9-1, Fukui-shi, Japan 910-8507

Summary
In this paper, we propose a Hopfield neural network based algorithm for efficiently solving the minimum graph bisection problem. In the proposed method, the internal dynamics of the neuron is modified to permit temporary increases in the energy function in order to avoid local minima. The proposed method is tested on a large number of random graphs. The simulation results show that the proposed algorithm is better than previous works for solving the minimum graph bisection problem.

Key words:
Hopfield neural network, Internal dynamics, Combinatorial optimization problem, NP-complete problem, Minimum graph bisection problem

1. Introduction

The minimum graph bisection problem is to divide the vertices set V of graph $G=(V,E)$ into two equal-size subsets G_0 and G_1 such that the number of edges connecting vertices in G_0 to vertices in G_1 is minimized. The graph bisection problem is an important problem in printed circuit board layout and communication networks [1] [2]. This problem is NP-complete [3]. Since NP-complete problems cannot be solved in polynomial time, at least at present, approximate algorithms are used to approximate the optimal solutions. Several approximate algorithms have been proposed [4][5][6] for the minimum graph bisection problem. The benchmark algorithm for graph bisection is due to Kernighan and Lin [7]. By combining parallel hill climbing [8], Kernighan-Lin algorithm [7] and seed-growth algorithm [9], Marks et al. proposed a new heuristic algorithm called PHC/SG+KL [10]. This algorithm is the best existing heuristic for the graph bisection problem. However the graph bisection problem is an NP-complete problem [1], no tractable algorithm is known for solving it.

For solving such combinatorial optimal problems, the Hopfield neural networks [11] also constitute an important avenue. These networks contain many simple computing elements (or artificial neurons) while cooperatively traverse the energy surface defined by $E(\upsilon)$ to find a local or global minimum. The simplicity of the neurons makes it promising to build them in large numbers to achieve high computing speeds by way of massive parallelism. Almost every type of combinatorial optimization problems has tackled by neural network [12]. In this paper, we introduce an improved Hopfield neural network algorithm for efficiently solving the minimum graph bisection problem. A large number of randomly generated undirected graphs are considered in simulations. The proposed algorithm is compared with the original Hopfield neural network and the best existing heuristic PHC/SG+KL [10]. The experimental results show that the proposed algorithm produces better solutions than the original Hopfield neural network and the PHC/SG+KL [10].

2. Problem Formulation

Let $G=(V,E)$ be an undirected graph, where V is the set of vertices and E is the set of edges. The edge from vertex i to vertex j is represented by $e_{ij} \in E$. Figure 1 shows an example of undirected 6-vertex graph. The minimum graph bisection problem is to find a partition of V into two nonempty, disjoin sets G_0 and G_1, such that $G_0 \cup G_1 = V$, $G_0 \cap G_1 = \Phi$, $|G_0| = |G_1|$, and the number of edges connecting vertices in G_0 to vertices in G_1 (the size of cut set, notated cut(G_0, G_1) is minimized.

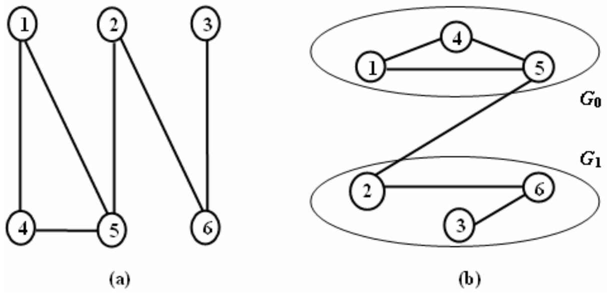

Figure 1. (a) a 6-vertex graph, and (b) a solution of (a).

Manuscript received March 5, 2007
Manuscript revised March 25, 2007

In general, an N-vertex minimum graph bisection problem can be mapped onto a Hopfield network with N neurons. Usually, the output variable y_j for neuron #*i* have the range $0 \le y_i \le 1$ and is a continuous and monotone creasing function of the instantaneous input x_i to neuron #*i*. It is proved by Hopfield [11] that the network converges to a minimum where the output of every neuron is at or near a 0 or 1. Thus, we can use neuron #*i* (*i=1,…,N*) to expresses the *i*-vertex, and the output ($y_i \approx 1$ or $y_i \approx 0$) of neuron #*i* (*i=1,…,N*) to expresses that the *i*-vertex is partitioned into the subset G_0 or G_1, respectively. The constraint condition can be expressed by follow:

$$\text{Subject}\left(\frac{N}{2}-\sum_{i}^{N} y_i\right)^2 = 0 \tag{1}$$

The minimum cut set condition can be expressed by

$$\text{Min}\left(\sum_{i=1}^{N}\sum_{j\neq i}^{N} d_{ij} y_i (1-y_j)\right) \tag{2}$$

where d_{ij} is 1 if edge (i, j) exists in the given graph, 0 otherwise, and is a symmetric matrix. Thus, the energy function of the Hopfield neural network for the minimum graph bisection problem is given by :

$$E=\left(\frac{1}{2}N-\sum_{i}^{N} y_i\right)^2+\sum_{i}^{N}\sum_{j\neq i}^{N} d_{ij} y_i \left(1-y_j\right) \tag{3}$$

For the Hopfield network, the standard energy function is as follow:

$$E=-\frac{1}{2}\sum_{i=1}^{N}\sum_{j\neq i}^{N} w_{ij} y_i y_j-\sum_{i=1}^{N} h_i y_i + C \tag{4}$$

where w_{ij} $(i, j=1,\cdots,N)$ is weight of a synaptic connection from the *j*# neuron to the *i*# one. h_i is external input of neuron #i and is also called threshold. C is constant.

For the minimum graph bisection problem, the resulting weight and threshold can now be obtained by equating the energy specified by Eq.(4) with the energy as in Eq.(3). The weight of the Hopfield network is as follows:

$$w_{ij}=2\left(d_{ij}-1\right) \tag{5}$$

And the threshold is

$$h_i=-\sum_{j=1}^{N}(1-\delta_{ij})-N+1 \tag{6}$$

In Eq. (6), the notation δ_{ij} is 1 if $i = j$, 0 otherwise.

It is proved that the state of the Hopfield neural network converges to a stable state with the energy taking on lower and lower values [11]. It can be viewed as seeking a minimum in a mountainous terrain. Thus, we can find the solution to the minimum graph bisection problem simply by observing the stable state that the Hopfield neural network reaches. However, it is usually difficult to find optimal solution because the inherent local minimum problem of the Hopfield neural network. Furthermore, there is no effective method to lead the network to reach the global minimum from a local minimum.

3. An Improved Neuron Dynamics

Consider the Hopfield neural network with the energy function of Eq. (4). The motion equation is composed of the partial derivation term of the energy function in the gradient descent method.

$$\frac{dx_i}{dt}=-\frac{\partial E}{\partial y_i}=\sum_{j\neq i}^{N} w_{ij} y_j + h \tag{7}$$

The internal potential x_i of neuron #i is updated according to the following equation:

$$x_i(t+1)=x_i(t)+\frac{dx_i(t)}{dt}\cdot\Delta t \tag{8}$$

The output y_i of neuron #i is updated from x_i using the following non-linear function called the neuron model:

$$y_i = 1\Big/\left(1+e^{(-x_i/T)}\right) \tag{9}$$

In Hopfield's work [11] it was shown that the above internal dynamics for a network with symmetric connection $(w_{ij}=w_{ji})$ always lead to a convergence to stable states. Unfortunately, because of the difficulties posed by local minima, the stable states are often far from the optimal solutions [12]. For efficiently solving the minimum graph bisection problem, we now propose an improved Hopfield neural network algorithm which can escape from local minima. In the proposed algorithm, the internal dynamic of neuron is modified as follow:

$$\begin{aligned} x_i(t+1) &= \alpha_i(y_i,t)\cdot x_i(t)+\frac{dx_i(t)}{dt}\cdot\Delta t \\ &= \alpha_i(y_i,t)\cdot x_i(t)+\sum_{j\neq i}^{N} w_{ij} y_j + h_i \end{aligned} \tag{10}$$

where $\Delta t = 1$ and $\alpha_i(y_i,t)$ is defined as follows:

$$\alpha_i\left(y_i,t\right)=1-e^{-\frac{(0.5-y_i(t)^2)\cdot t}{\lambda}} \tag{11}$$

where t is the updating iteration, λ is a positive constant which specify the rate of increase of $\alpha_i(y_i,t)$. Evidently, $\alpha_i(y_i,t)$ increases from 0 to 1 in the updating proceeds. The modified internal dynamic behavior means that the change of the internal potential in neuron #i is now controlled by a new parameter $\alpha_i(y_i,t)$ which represents the stabilization of neuron #i . When the state of the neuron is far from 0 and 1 or the network is in initial stage of updating (far from stable state), the stabilization of the neuron is very low ($\alpha_i(y_i,t)$ is near 0). Thus, the internal potential of neuron ($x_i(t+1)$) is mainly determined by the weight state of other neurons (the second term of Eq. (10)).

As the state of the neuron approaches 0 or 1, the stabilization of the neuron will increase. Finally the stabilization of the neuron ($\alpha_i(y_i,t)$) will come close to 1, and the internal dynamic behavior of the neuron will tend toward the original updating mode, which guarantees that the network always converges to a stable state.

Now we show that the proposed method allows initial increases in energy which become less important as updating proceeds, until finally the network tends toward a steepest descent algorithm. Considering the energy function of Eq. (4), the change of energy caused by the change in the states of neurons is

$$\begin{aligned}\Delta E = &-\frac{1}{2}\sum_{i,j\neq i} w_{ij}\left[y_i(t+1)y_j(t+1) - y_i(t)y_j(t)\right] \\ &-\sum_i h_i\left[y_i(t+1) - y_i(t)\right]\end{aligned} \tag{12}$$

Adding $\left(y_i(t+1)y_j(t)\right)$ and $\left(-y_i(t+1)y_j(t)\right)$ to the first term of Eq. (12), and simplifying, we have:

$$\begin{aligned}\Delta E = &-\frac{1}{2}\sum_{i,j\neq i} w_{ij}y_i(t+1)\Delta y_j(t+1) \\ &-\frac{1}{2}\sum_{i,j\neq i} w_{ij}\Delta y_i(t+1)y_j(t) \\ &-\sum_i h_i\Delta y_i(t+1)\end{aligned} \tag{13}$$

In the asynchronous parallel mode, we can suppose that at updating time t, the state of neuron $\#k$ is changed. Thus we have $\Delta y_k(t+1)\neq 0$ and $\Delta y_i(t+1)=0$ for $i\neq k$. Thus, Eq. (13) can be reduced to

$$\begin{aligned}\Delta E = &-\frac{1}{2}\sum_{i\neq k} w_{ik}y_i(t+1)\Delta y_k(t+1) \\ &-\frac{1}{2}\sum_{i\neq k} w_{ki}\Delta y_k(t+1)y_i(t) \\ &-h_k\Delta y_k(t+1)\end{aligned} \tag{14}$$

$$y_i(t+1) = y_i(t) \qquad for \quad i\neq k \tag{15}$$

Using $w_{ik} = w_{ki}$, Eq. (14) can be rewritten as

$$\Delta E = -\Delta y_k(t+1)\cdot\left[\sum_{i\neq k} w_{ik}y_i(t) + h_k\right] \tag{16}$$

Using Eq. (10), Eq. (16) can be rewritten as follows:

$$\Delta E = -\Delta y_k(t+1)\cdot\left[x_k(t+1) - \alpha_k(y_k,t)x_k(t)\right] \tag{17}$$

Considering the case $x_k(t) > x_k(t+1)$, according to the characteristic of the sigmoid function, we have

$$\Delta y_k(t+1) < 0 \tag{18}$$

Moreover $x_k(t) > x_k(t+1)$, if $\alpha_k(y_k,t)$ is sufficiently small, it is possible that

$$x_k(t+1) > \alpha_k(y_k,t)\cdot x_k(t) \tag{19}$$

Using Eq. (18) and (19), we can know from Eq. (17) that when $\alpha_k(y_k,t)$ is small, $\Delta E > 0$ is possible in the case of $x_k(t) > x_k(t+1)$. The possibility of an increase of energy becomes smaller as $\alpha_k(y_k,t)$ increases until finally the network tends toward a steepest descent algorithm. Thus we can say that the proposed method which uses Eq. (10) and (11) provides a mechanism for escaping from local minima and converging to a good stable state by introducing a stabilization factor $\alpha_k(y_k,t)$ for neurons.

4. Simulation results

The proposed algorithm for the minimum graph bisection problem was experimented on PC Station to a large number of randomly generated graphs defined in terms of two parameters, N and ρ. The parameter N specifies the number of vertices in the graph, the parameter ρ, $0<\rho<1$, specifies the probability that any given pair of vertices constitute an edge. In simulations, N and ρ is defined as follows:

- Vertices (*N*) : 80, 100, 150, 200, 250, 300.
- Probability (ρ): 0.05, 0.15, 0.25.

In Eq. (11), λ is introduced to control the increase speed of stabilization of neuron ($\alpha_i(y_i,t)$). Furthermore, because when $\alpha_i(y_i,t)$ approaches 1 the network will tend to convergence to a stable state, we can see that λ also control the convergence speed of the network. The smaller is the constants, the faster the network convergences to a stable state. For practical purpose we chose these constants that are as small as possible. This offers the most convergence. But too smaller values will cause the network fall into local minima easily. We found λ around 15 worked very well in the minimum graph bisection problem. Besides, the temperature parameter T in the sigmoid function was set to 2.5.

In order to give some evidence that the proposed algorithm behaves very well in practice, we directly compared the performance of the proposed algorithm with the original Hopfield network and the heuristic algorithm PHC/SG+KL [10]. For each of instances, 100 simulation runs were performed. Information on the test graphs as well as all results found by the proposed algorithm and other algorithms were summarized in Table 1. From the table, we can see that the proposed method was not only much better than the original Hopfield network but also superior to the best existing heuristic algorithm in terms of the solution quality.

Table 1: Computational results

Vertices	Probabirity	Edges	Hopfield network	PHC/SG+KL	Proposed algorithm
80	0.05	158	34	26	26
80	0.15	474	164	151	151
80	0.25	790	305	292	292
100	0.05	247	63	51	50
100	0.15	742	266	247	247
100	0.25	1235	494	473	473
150	0.05	558	159	139	139
150	0.15	1676	651	605	605
150	0.25	2790	1165	1113	1113
200	0.05	995	326	274	274
200	0.15	2985	1178	1128	1128
200	0.25	4975	2113	2044	2045
250	0.05	1556	522	464	463
250	0.15	4668	1939	1823	1820
250	0.25	7778	3406	3272	3271
300	0.05	2242	786	711	712
300	0.15	6727	2849	2683	2681
300	0.25	11212	4937	4795	4790

5. Conclusions

We have proposed an efficient algorithm for solve the minimum graph bisection problem, and showed its effectiveness by simulation experiments. The proposed algorithm is based on an improved Hopfield neural network in which the internal dynamics is modified to permit temporary increases in the energy function in order to help the network escape from local minima and increase the exchange of information between neurons. In order to verify the proposed algorithm we tested it with a large number of randomly generated examples. The proposed algorithm was compared with original Hopfield network and the best existing heuristic algorithm. The simulations results showed that the proposed method could provide better solutions than the previous works.

References

[1] L. Tao and Y. C. Zhao, " Effective heuristic algorithm for VLSL circuit partition " IEE Proceedings-G, Vol. 140, No. 2, pp. 127-134, April, 1993.

[2] L. A. Sanchis, "Multiple-way network partitioning" IEEE Trans. Comput, Vol.38, No.1, pp. 62-81, Jan, 1989.

[3] M. R. Garey, D. S. Johnson, and L. Stockmeyer. "Some simplified NP-complete graph problem" Theoretical Computer Science, vol. 1 pp. 237-267, 1976.

[4] T. N. Bui, F. T. Leighton, S. Chaudhuri and M. Sipser, "Graph bisection algorithm with good average case behavior," Combinatorica, Vol.7, pp.171-191, 1987.

[5] E. R. Barnes, A. Vannelli and J. Q. Walker, "A new heuristic for partition the nodes of a graph," SIAM Journal on Discrete Mathematics, Vol.3, pp.299-305, 1988.

[6] C. Tu and H. Cheng, "Spectral methods for graph bisection problem," Computers Ops Res. Vol.25, No.7, pp.519-530, 1998.

[7] B. Kernighan and S. Lin, "An efficient heuristic procedure for partitioning graphs," The Bell System Technical Journal, Vol.49, No.2, pp.291-307, 1970.

[8] C. A. Tovey, "Hill climbing and multiple local optima," SIAM Journal on Algebraic and Discrete Methods, Vol.6, No.3, pp.384-393, 1985.

[9] W. E. Donath, "Logic partitioning, " In B. Preas and M. Lorenzetti, Editors, Physical Design Automation of VLSI Systems, pp.65-86, 1988.

[10] J. Marks, W. Ruml, S. Shieber, and J. Ngo, "A Seed-Growth Heuristic for Graph Bisection," Proceedings of Algorithms and Experiments 98, Italy, R. Battiti and A. Bertossi (eds), pp.76-87, 1998.

[11] J. J. Hopfield, and D. W. Tank, "Neural' computation of decisions in optimization problems" Biol. Cybernet. No. 52, pp. 141-152, 1985.

[12] K. A. Smith, "Neural network for combinatorial optimization: A review of more than a decade of research" INFORMS J. Computing, Vol. 11, No. 1, pp. 15-34, 1999.