A Saturation Binary Neural Network for Bipartite Subgraph Problem

Cui Zhang[1], Li-Qing Zhao[2], and Rong-Long Wang[2]

[1] Department of Autocontrol, Liaoning Institute of Science and Technology, Benxi, China
[2] Graduate School of Engineering, University of Fukui, Bunkyo 3-9-1, Fukui-shi, Japan
bxlkyzhangcui@163.com, nkzlq@hotmail.com, wang@u-fukui.ac.jp

Abstract. In this paper, we propose a saturation binary neuron model and use it to construct a Hopfield-type neural network called saturation binary neural network to solve the bipartite sub-graph problem. A large number of instances have been simulated to verify the proposed algorithm, with the simulation result showing that our algorithm finds the solution quality is superior to the compared algorithms.

Keywords: Saturation binary neuron model, Saturation binary neural network, Combinatorial optimization problems, Bipartite sub-graph problem.

1 Introduction

The objective of the bipartite subgraph problem[1] is to find a bipartite subgraph with maximum number of edges of a given graph. It was proved by Garey, Johnson, and Stockmeyer [1], Karp [2], and Even and Shiloach [3] that the problem of finding a bipartite subgraph of a given graph with the maximum number of edges is NP-complete. Thus, the efficient determination of "maximum" bipartite subgraphs is a question of both practical and theoretical interest. Because efficient algorithms for this NP-complete combinatorial optimal problem are unlikely to exist, the bipartite subgraph problem has been widely studied by many researchers on some special classes of graphs. An algorithm for solving the largest bipartite subgraphs in triangle-free graph with maximum degree three has been propossed for practical purpose [4]. Grotschel and Pulleyblank [5] defined a class of weakly bipartite graphs. Barahona [6] characterized another class of weakly bipartite graphs. For solving such combinatorial optimal problems, Hopfield neural networks [7]-[11] constitute an important avenue. These networks contain many simple computing elements (or artificial neurons), which cooperatively traverse the energy surface 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. Using the Hopfield neural network, Lee, Funabiki and Takefuji [12] presented a parallel algorithm for bipartite subgraph problem. However with the Hopfield network, the state of the system is forced to converge to local minimum and the rate

D.-S. Huang et al. (Eds.): ICIC 2011, LNBI 6840, pp. 64–70, 2012.

to get the maximum bipartite subgraph is very low. Global search methods such as simulated annealing can be applied to such problem [13], but they are generally very slow [14]. No tractable algorithm is known for solving the bipartite subgraph problem.

In this paper we propose a saturation binary neuron model and use it to construct a Hopfield-type neural network for efficiently solving the bipartite sub-graph problems. In the proposed saturation binary neuron model, once the neuron is in excitatory state, then its input potential is in positive saturation where the input potential can only be reduced but cannot be increased, and once the neuron is in inhibitory state, then its input potential is in negative saturation where the input potential can only be increased but cannot be reduced. Using the saturation binary neuron model, a saturation binary neural network is constructed to solve the bipartite subgraph problems. The simulation results show that the saturation binary neural network can find better solutions than the other method, for example the Lee et al.'s algorithm.

2 Saturation Binary Neural Network

In the proposed algorithm, a novel neuron updating rule is proposed. To avoid the neural network entering a local minimum stagnation, the neuron network is updated according to the state of the neuron. According to different states, the updating rule is different. In this section, the proposed saturation binary neural network is introduced.

For solving the bipartite subgraph problem, the state of the neuron is binary, which is called binary neuron model. Note that the standard energy function of Hopfield network can be written as follows:

$$E = -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\omega_{ij}V_iV_j - \sum_{i=1}^{N}\theta_i V_i \tag{1}$$

Where ω_{ij} is weight of a synaptic connection from j_{th} neuron to the i_{th} neuron, θ_i is the external input of neuron #i and is also called threshold. V_i, V_j is the output of the neuron which also called the state of the neuron.

The Hopfield neural network can find the solution by seeking the local minimum of the energy function E using the motion equations of neurons.

$$\frac{dU_i(t)}{dt} = \sum_{j \neq i}^{N}\omega_{ij}V_j + \theta_i \tag{2}$$

In this paper, the output Vi is updated from ui using sigmoid function:

$$V_i = 1/(1 + e^{(-U_i/T)}) \tag{3}$$

In the Hopfield neural network, the updating method of input potential U_i is especially important. In conventional neuron model, the input potential U_i is updated from the Eq. 4 or Eq. 5.

$$U_i(t+1) = \frac{dU_i(t)}{dt} \tag{4}$$

$$U_i(t+1) = U_i(t) + \frac{dU_i(t)}{dt} \tag{5}$$

Where $dU_i(t)/dt$ derives from the energy function E based on the gradient descent method:

$$\frac{dU_i(t)}{dt} = -\frac{\partial E(V_1, V_2, ..., V_n)}{\partial V_i} \tag{6}$$

The Hopfield-type binary neural network is usually constructed using the above neuron model. In order to improve the global convergence quality and shorten the convergence time, we propose a new neuron model called saturation binary neuron model (SBNM) which consists of the following important ideas.

(1). Once the neuron is in excitatory state (the output V_i=1), then the input potential is assumed to be in positive saturation. In the positive saturation, the input potential U_i can only be reduced but cannot be increased.

For the case of V_i=1:
if $dU_i(t)/dt < 0$

$$U_i(t+1) = U_i(t) + \frac{dU_i(t)}{dt} \tag{7}$$

else

$$U_i(t+1) = U_i(t) \tag{8}$$

(2). Once the neuron is in inhibitory state (the output V_i=0), then the input potential is assumed to be in negative saturation. Then the input potential can only be increased but cannot be reduced.

For the case of V_i=0
if $dU_i(t)/dt > 0$

$$U_i(t+1) = U_i(t) + \frac{dU_i(t)}{dt} \tag{9}$$

else

$$U_i(t+1) = U_i(t) \tag{20}$$

The above process can be presented in Fig. 1. The neuron update process can be set to three parts according to Eq. 7~Eq.10.

Note that the input/output function in the McCulloch-Pitts neuron [15] or hysteresis McCulloch-Pitts neuron [16] can be used to update the output V_i. Using the above neuron model, we construct a Hopfield-type neural network called saturation binary neuron network (SBNN). The following procedure describes the synchronous parallel algorithm using the SBNN to solve a COP. Note that N is the number of

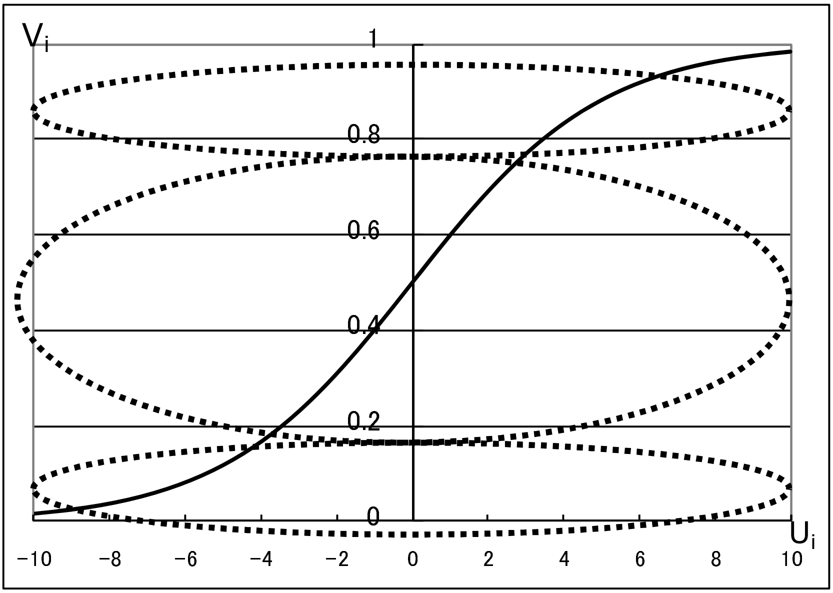

Fig. 1. The neuron state of the saturation binary neuron model

neuron, *targ_cost* is the target total cost set by a user as an expected total cost and *t_limit* is maximum number of iteration step allowed by user.

1. Set t=0 and set *targ_cost*, *t_limit*, and other constants.
2. The initial value of U_i for $i=1,\ldots,N$ is randomized.
3. Evaluate the current output V_i(t) for $i=1,\ldots,N$.
4. Check the network, if *targ_cost* is reached, then terminate this procedure.
5. Increment t by 1. if t> *t_limit*, then terminate this procedure.
6. For $i=1,\ldots,N$
   a. Compute Eq. 6 to obtain $dU_i(t)/dt$
   b. Update $U_i(t+1)$, using the proposed saturation binary neuron model(Eq. 7~Eq.10).
7. Go to the step 3.

3 Solving Biprtite Sub-graph Problems Using SBNN

For an N-vertex M-edge graph G=(V, E), if the vertex set V can be partitioned into two subsets $V1$ and $V2$, such that each edge of G is incident to a vertex in $V1$ and to a vertex in $V2$, then the graph G is called bipartite graph, and be denoted by G=($V1$, $V2$, E). Given a graph G=(V, E) with a vertex set V and an edge set E, the goal of bipartite sub-graph problem is to find a bipartite sub-graph with the maximum number of edges. In other words, the goal of bipartite sub-graph problem is to remove the minimum number of edges from a given graph such that the remaining graph is a bipartite graph. Consider a simple undirected graph composed of four vertices and four edges as shown in Fig.2 (a). The graph is bipartite as long as one edge is removed. Fig.2 (b) shows a bipartite graph. A bipartite graph is usually shown with the two subsets as top and bottom rows of vertices, as in Fig.2 (b), or with the two subsets as left and right columns of vertices.

An *N*-vertex bipartite sub-graph problem can be mapped onto a Hopfield neural network with *N* neurons [17]. Neuron *#i* (*i=1,...,N*) expresses the *i*-vertex, and the output ($y_i = 1$ or $y_i = 0$) of neuron *#i* (*i=1,...,N*) expresses that the *i*-vertex is partitioned into the subset *V1* or *V2*, respectively. Thus the *N*-vertex bipartite sub-graph problem can be mathematically transformed into the following optimization problem.

$$\text{Min}\,[\sum_{i=1}^{N}\sum_{j\neq i}^{N} d_{ij} y_i y_j + \sum_{i=1}^{N}\sum_{j=1}^{N} d_{ij}(1-y_i)(1-y_j))\,] \tag{11}$$

where d_{ij} is 1 if edge (*i, j*) exists in the given graph, 0 otherwise. The first term of Eq. 8 is the number of edges connecting two vertices in the subset *V1*, and the second term is the number of edges in the subset *V2*. The number of the edges both in the subset *V1* and *V2* is smaller the number of the edges between the *V1* and *V2* is larger. Thus, we can remove the minimum number of edges defined by Eq. 8 to obtain a bipartite sub-graph from a given graph.

When we follow the mapping procedure by Hopfield and Tank, the energy function for the bipartite sub-graph problem is given by

$$E = A\sum_{i=1}^{N}\sum_{j\neq i}^{N} d_{ij} y_i y_j + B\sum_{i=1}^{N}\sum_{j=1}^{N} d_{ij}(1-y_i)(1-y_j)) \tag{12}$$

where A, B are parameters which are used to balance the two terms of Eq. 12.

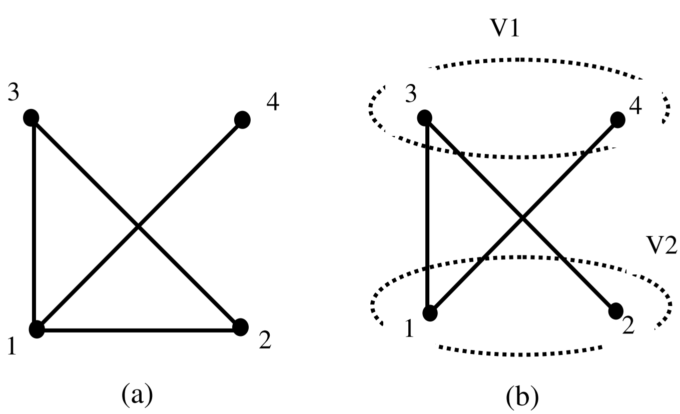

Fig. 2. (a) Graph is not bipartite. (b) Graph is bipartite.

Table 1. The simulation results

No.vertex	No.edges	Lee et al.[12]	Proposed algorithm
50	61	52	53
50	183	133	136
50	305	203	205
80	158	127	134
80	474	325	329
80	790	504	513
100	247	196	207
100	742	492	502
100	1235	761	779
150	558	402	423
150	1676	1062	1076
150	2790	1645	1693
200	995	685	719
200	2985	1838	1862
200	4975	2886	2954
250	1556	1060	1104
250	4668	2809	2854
250	7778	4435	4516
300	2242	1486	1532
300	6727	3987	4051
300	11212	6393	6451

4 Simulation Results

To widely verify the proposed algorithm, we have tested the algorithm with a large number of randomly generated graphs [18] defined in terms of two parameters, n and p. The parameter n specifies the number of vertices in the graph; the parameter p, $0<p<1$, specifies the probability that any given pair of vertices constitutes an edge. In the experiment, up to 300-vertex graphs with different probability are used.

To evaluate our results, we compared the results of Lee et al. with our results. For each of instances, 100 simulation runs were performed. Information on the test graphs as well as all results is shown in Table.1. The results that we recorded for each graph are the solutions in number of embedded edges, produced by the algorithm of Lee et al., and by the proposed algorithm. Table.1 shows that the proposed method can find better solutions than Lee et al.'s algorithm in all problems, although the computing time becomes longer.

5 Conclusion

We have proposed a saturation binary neuron model and used it to construct a Hopfield-type neural network which is called saturation binary neural network (SBNN). The SBNN is used to solve the bipartite sub-graph problems. The simulation results show that SBNN is capable of finding better solutions than other methods. Also, it can be seen that the SBNN is problem independent and can be used to solve other combinatorial optimization problems.

References

1. Garey, M.R., Johnson, D.S., Stockmeyer, L.J.: Some simplified NP-complete graph problem. Theor. Comput. Sci., 237–267 (1976)
2. Karp, R.M.: Reducibility among combinatorial problems. In: Complexity of Computer Computations, pp. 85–104. Plenum, New York (1972)
3. Even, S., Shiloach, Y.: NP-completeness of several arrangement problems. Technical Report 43, Department of computer Science, Technion, Haifa Israel (1975)
4. Bondy, J.A., Locke, S.C.: Largest bipartite subgraph in triangle-free graphs with maximum degree three. J. Graph Theory, 477–504 (1986)
5. Grotscheland, M., Pulleyblank, W.R.: Weakly bipartite graphs and the max-cut problem. Oper. Res. Lett., 23–27 (1981)
6. Barahona, F.: On some weakly bipartite graph. Oper. Res. Lett. 2(5), 239–242 (1983)
7. Hopfield, J.J.: Neural networks and physical systems with emergent collective computational abilities. Proc. Nat. Acad. Sci. U.S. 79, 2554–2558 (1982)
8. Hopfield, J.J.: Neurons with graded response have collective computation properties like those of two-state neurons. Proc. Nat. Acad. Sci. U.S. 81, 3088–3092 (1982)
9. Hopfield, J.J., Tank, D.W.: 'Neural' computation of decisions in optimization problems. Bio. Cybern. 52, 141–152 (1985)
10. Hopfield, J.J., Tank, D.W.: Simple 'Neural' optimization networks: An a/d converter, signal decision circuit, and a linear programming circuit. IEEE Trans. Circuits Syst. 33(5), 533–541 (1986)
11. Hopfield, J.J., Tank, D.W.: Computing with neural circuits: A model. Science 233, 625–633 (1986)
12. Lee, K.C., Funabiki, N., Takefuji, Y.: A parallel improvement algorithm for the bipartite subgraph problem. IEEE Trans. Neural Networks 3(1), 139–145 (1992)
13. Garey, M.R., Johnson, D.S.: Computers and Intractability: a Guide to the Theory of NP-Completeness. W.H. Freeman, New York (1979)
14. Ackley, D.H., Hinton, G.E., Sejnowski, T.J.: A learning algorithm for Boltzman Machines. Cognitive Science 9, 147–169 (1985)
15. McCulloch, W.S., Pitts, W.H.: A logical calculus of ideas immanent in nervous activity. Bull. Math. Biophys 5, 115–133 (1943)
16. Takefuji, Y., Lee, K.C.: An artificial hysteresis binary neuron: A model suppressing the oscillatory behaviors of neural dynamics. Biol. Cybern. 64, 353–356 (1991)
17. Wang, R.L., Tang, Z., Cao, Q.P.: A Hopfield Network Learning Method for Bipartite Subgraph Problem. IEEE Trans. Neural Networks 15(6), 1458–1465 (2004)
18. Johnson, D.S., Aragon, C.R., McGeoch, L.A., Schevon, C.: Optimization by simulated annealing: An experimental evaluation; Part 1, graph partitioning. Operations Research 37(6), 865–892 (1989)