Paper

A New Parallel Algorithm Analogous to Elastic Net Method for Bipartite Subgraph Problem

Zheng Tang* Non-member
Rong Long Wang** Member
Jia Hai Wang* Non-member
Qi Ping Cao*** Non-member

The goal of the bipartite subgraph problem, which is an NP-complete problem, is to remove the minimum number of edges in a given graph such that the remaining graph is a bipartite graph. Enlightened by the elastic net method that was introduced by Durbin and Willshaw for finding shortest route for the Traveling Salesman Problem (TSP), we proposed a new parallel algorithm for the bipartite subgraph problem. The approach jointly tends to satisfy the constraint condition and minimizes the number of removed edges. The collective computational properties of the proposed approach are also proved theoretically. A large number of instances have been simulated to verify the proposed algorithm. The simulation results show that our algorithm finds a solution superior to that of the best existing parallel algorithms.

Keywords: bipartite subgraph problem, elastic net, Hopfield neural network, NP-complete problem

1. Introduction

The bipartite subgraph problem is a classical problem in combinatorial optimization. The task is to find a bipartite subgraph with maximum number of edges of a given graph. This problem is known to be NP-complete, and it is generally believed that the computational power needed to solve it grows exponentially with the number of edges (1)~(3). Thus, the efficient determination of maximum bipartite subgraph is a question of both practical and theoretical interest. Because efficient algorithms for this NP-complete combinatorial optimization 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 a triangle-free graph with maximum degree three has been proposed 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 optimization, Hopfield neural networks (7)~(11) constitute an important avenue. Using the Hopfield neural network, Lee et al. (12) presented a parallel algorithm for the bipartite subgraph problem. Unfortunately, with the Hopfield network, the state of the system is forced to converge to a local minimum and the rate 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, and furthermore there is no more efficient parallel algorithm than Lee, Funabiki and Takefuji (12) available for solving the bipartite subgraph problem.

In this paper, we introduce a parallel algorithm analogous to elastic net method for the bipartite subgraph problem. The procedure of the algorithm is similar to that of the Hopfield network. But different to the Hopfield network, we construct the energy function for the bipartite subgraph problem using a nonlinear Gaussian function, which is enlightened by the elastic net method. The energy function has two terms, the constraint term and the cost term, which are reflected in the energy function with a nonlinear Gaussian function and a linear function, respectively. The energy function is minimized using gradient descent algorithm and the tradeoff between these two terms is controlled by a scale parameter K. A large number of randomly generated examples are simulated to verify the proposed algorithm. Simulation results are compared with the ones found by the algorithm of Lee et al. (12). The reasons to compare the proposed algorithm with Lee et al.'s algorithm is that 1) Lee et al.'s algorithm is one of the most popular parallel algorithms, and 2) Lee et al. claimed that their algorithm could find optimum solutions. The simulation results show that the proposed algorithm works well on finding a maximum or a better bipartite subgraph than the algorithm of Lee et al.

2. The Bipartite Subgraph Problem

Let $G=(V, E)$ be an undirected graph, where V is the set of vertices and E is the set of edges, if the vertex set

* The Faculty of Engineering, Toyama University, Toyama-shi, Japan, 930-8555
** The Faculty of Engineering, Fukui University, Fukui-shi, Japan, 910-8507
*** Tateyama Systems Institute, Toyama-shi, Japan 930-0001

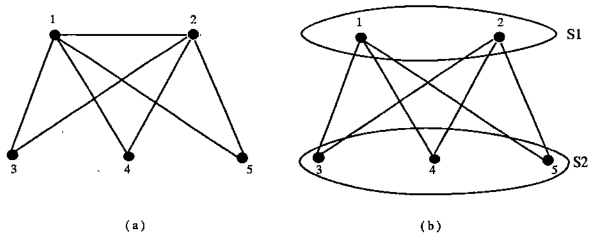


Fig. 1. (a) A simple undirected graph composed of five vertices and four edges (b) One of its bipartite graphs.

V of graph G can be partitioned into two subsets S_1 and S_2, such that $S_1 \cup S_2 = V$ and $S_1 \cap S_2 = \Phi$ and each edge of G is incident to a vertex in S_1 and to a vertex in S_2, then the graph G is called the bipartite graph. Given a graph $G = (V, E)$, the goal of the bipartite subgraph problem is to find a bipartite subgraph with maximum number of edges of the given graph. In other words, the goal of the bipartite subgraph problem is to remove the minimum number of the edges from a given graph such that the remaining graph is a bipartite graph. Consider a simple undirected graph composed of five vertices and seven edges as shows in Fig.1 (a). The graph is bipartite as long as one edge must be removed. Fig.1 (b) shows a bipartite graph of the graph. A bipartite graph is usually shown with the two subsets as top and bottom rows of vertex, as in Fig.1 (b), or with the two subsets as left and right columns of vertex. And the objective function can be formulated for this optimization problem whose minimum value corresponds to the optimal solution. In a reasonable formulation, there are two components to the objective function: one which is used to minimize the number of removed edges and one which is used to guarantee every vertex is distributed into one and only one vertex subset. This optimization problem can be mathematically stated as finding minimum of the following objective function.

$$\sum_{i}^{N}\sum_{j\neq 1}^{N} d_{ij}(V_{is1}V_{js1} + V_{is2}V_{js2}) + \sum_{i=1}^{N} |V_{is1} + V_{is2} - 1|$$

where

$$V_{ip} = \begin{cases} 1 & \textit{if vertex } i \in S_p \\ 0 & \textit{otherwise} \end{cases}$$

and $d_{ij}=1$ if there is an edge between vertex i and vertex j, 0 otherwise.

3. Description of the Proposed Algorithm

Let y_{ij} represent whether or not i-vertex (i=1, $\cdots$, N) should be grouped into vertex subset P (P=1, 2, which represent subset S_1 and subset S_2). For example, the state (y_{i1}=1, y_{i2}=0) indicates that the i-vertex is grouped into subset S_1. The following states ($y_{i1}=y_{i2}=0$) and ($y_{i1}=y_{i2}=1$) express no partition and double partition violation, respectively. These partition violation conditions can be expressed by follow:

$$\sum_{i}^{N}(\sum_{j}^{2} y_{ij} - 1)^2 = 0 \cdots\cdots\cdots\cdots\cdots\cdots \quad (1)$$

The minimum number of removed edges condition can be expressed by follow:

$$\min(\sum_{i}^{N}\sum_{k\neq i}^{N}\sum_{j}^{2} d_{ik}y_{ij}y_{kj}) \cdots\cdots\cdots\cdots \quad (2)$$

where d_{ik} is 1 if edge(i , k) exits in the given graph, 0 otherwise, and corresponds to a symmetric matrix. For example, the matrix for Fig.1 (a) can be written as:

$$[d_{ik}] = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \end{bmatrix}$$

Enlightened by the elastic net method [15][16], we use a different energy function from that usually used in the Hopfield networks [9]. Our energy function is expressed as the following equation:

$$e = -KA\sum_{i}^{N}\phi(C_i, K) + B\sum_{i}^{N}\sum_{k\neq i}^{N}\sum_{j}^{2} d_{ik}y_{ij}y_{kj} \cdots\cdots\cdots\cdots \quad (3)$$

where $\phi(C, K) = e^{-C^2/(2K^2)}$, and

$$C_i = \sum_{j=1}^{2} y_{ij} - 1$$

which is the i-term of the constraint condition (Eq.(1)) of the bipartite subgraph problem. A and B are positive constants, and the K is a positive scale parameter. From Eq.(3), we can see that the first term of Eq.(3) is smallest if the constraint condition (Eq.(1)) is satisfied, and the second term of Eq.(3) is smallest if the bipartite subgraph is largest.

In order to decrease the energy function (Eq.(3)), so as to find the optimal solution of the bipartite subgraph problem, we can use the gradient descent rule of the Hopfield network to update the y_{ij}. Takefuji and Lee [17] showed that McCulloch-Pitts neuron model [18] could guarantee an n-variable function converge to a local minimum. The McCulloch-Pitts neuron model used in Takefuji and Lee's theory is represented as :

$y_{ij} = 1$ if $x_{ij} > 0$, and $y_{ij} = 0$ if $x_{ij} \leq 0$

where x_{ij} is the input of neuron $\sharp ij$ and is updated according the following equation:

$$x_{ij}(t+1) = x_{ij}(t) - \frac{dx_{ij}}{dt}\Delta t \cdots\cdots\cdots\cdots \quad (4)$$

where $\frac{dx_{ij}}{dt}$ is defined by following motion equation:

$$\frac{dx_{ij}}{dt} = -\frac{\partial e}{\partial y_{ij}} \cdots\cdots\cdots\cdots\cdots\cdots \quad (5)$$

In the similar procedure with Takefuji and Lee, we use a sigmoid function as the input-output characteristics of the neurons:

$$y_{ij} = \frac{1}{1+e^{(-x_{ij}/T)}} \cdots\cdots (6)$$

The input x_{ij} to the neuron $\sharp ij$ is updated according the following rule.

$$x_{ij}(t+1) = x_{ij}(t) - \frac{dx_{ij}}{dt}\Delta t \cdots\cdots (7)$$

and the motion equation is defined by follow:

$$\frac{dx_{ij}}{dt} = -K\frac{\partial e}{\partial y_{ij}} \cdots\cdots (8)$$

where K is a positive scale parameter which is as same as that in Eq.(3). Using Eq.(3) we have:

$$\frac{\partial e}{\partial y_{ij}} = AC_i\phi(C_i, K)/K + B\sum_{k\neq i}^{N} d_{ik}y_{kj} \cdots\cdots (9)$$

First, we will prove that our updating rule (Eq.(6) - Eq.(8)) decreases the energy function (Eq.(3)) with the time evolution-collective computational properties.

Consider the derivatives of the energy function e (Eq.(3)) with respect to time t,

$$\frac{de}{dt} = \sum_{ij}\frac{dy_{ij}}{dt}\frac{de}{dy_{ij}} \cdots\cdots (10)$$

Using Eq.(8) we have:

$$\begin{aligned}\frac{de}{dt} &= -\frac{1}{K}\sum_{ij}\frac{dy_{ij}}{dt}\left(\frac{dx_{ij}}{dt}\right)\\ &= -\frac{1}{K}\sum_{ij}\frac{dx_{ij}}{dt}\frac{dy_{ij}}{dx_{ij}}\frac{dx_{ij}}{dt}\\ &= -\frac{1}{K}\sum_{ij}\left(\frac{dy_{ij}}{dx_{ij}}\right)\left(\frac{dx_{ij}}{dt}\right)^2 \cdots\cdots (11)\end{aligned}$$

Since y_{ij} is a monotone increasing function of x_{ij} (the sigmoid function), dy_{ij}/dx_{ij} is positive, and K is a positive parameter, each term in Eq.(11) is nonnegative. Therefore

$$\frac{de}{dt} \leq 0 \text{ for all } i, j \cdots\cdots (12)$$

Together with the bound of e, Eq.(12) shows that the time evolution of the system is a motion in state space that seeks out minima in e and comes to stop at such points.

We have proved that for fixed K ($K > 0$), the update (Eq.(6) - (8)) would result in a convergence to a local minimum of the energy function e. Now we discuss the behavior of the energy function as the constant K changes. Informally, the first term of Eq.(3) tends to impel the solutions to satisfy the constraints, and the second term of Eq.(3) tries to make the number of removed edges small. Furthermore, it has been proved [16] that because the Gaussian function $\phi(C, K)=e^{-C^2/(2K^2)}$ is a positive bounded function, at large values of K the energy function is smoothed and there is only one minimum. At small values of K, the energy function contains many local minima, all of which correspond to possible solutions to the problem, and deepest minimum is the optimal solution. Thus, in the same way as the elastic net method, our algorithm proceeds by starting at large K, and gradually reducing K, keeping to a local minimum of e. We would like this minimum that is tracked to remain the global minimum as K becomes small. In our algorithm, when K tends to zero, for e to remain bounded and for every i, $(\sum_{j=1}^{2} y_{ij} - 1)$ tends to zero, i.e., the constrains must be satisfied. Then the second term (i.e. the cost term) in the expression for e is minimized, and finally, a feasible solution (a local or global minimum) is reached by the second term.

4. Algorithm Procedure

The following procedure describes the proposed algorithm for the bipartite subgraph problem of an N-vertex graph.

1. Set t=0, the initial value of x_{ij} for i=1, $\cdots$, N, for j=1, 2 are randomized.
2. Update the value of y_{ij} using Eq.(6)
3. Compute the $\frac{\partial e}{\partial y_{ij}}$ using following equation.

$$\frac{\partial e}{\partial y_{ij}} = AC_i\phi(C_i, K)/K + B\sum_{k\neq i}^{N} d_{ik}y_{kj}$$

for i=1, $\cdots$, N, for j=1, 2.

4. Change the x_{ij} for i=1, $\cdots$, N, for j=1, 2

$$x_{ij} = x_{ij} - K\frac{\partial e}{\partial y_{ij}}.$$

5. The output y_{ij} is updated from x_{ij} using following sigmoid function:

$$y_{ij} = \frac{1}{1+e^{(-x_{ij}/T)}}$$

for i =1, $\cdots$, N, for j =1, 2.

6. Increment the t by 1.
7. If t modulo 5=0, K=0.99 $*$ K
8. If the network reaches a stable state, terminate the procedure.
9. Go to step 2.

By this procedure, we can find the solution of the bipartite subgraph problem simply by observing the final configuration of y_{ij}, which is similar to the Hopfield neural network.

5. Simulation Results

The proposed algorithm was implemented in C++ to a large number of graphs. The parameter values used in the simulations were A=3, B=0.08, T=1.42, the initial value of K was 0.65 and was reduced by 99% every 5 iterations to a final value in the range 0.08-0.09.

The first graph we tested was a graph with 30 vertices and 50 edges. For this graph, Lee et al. found a maximum bipartite subgraph with 42 edges embedded

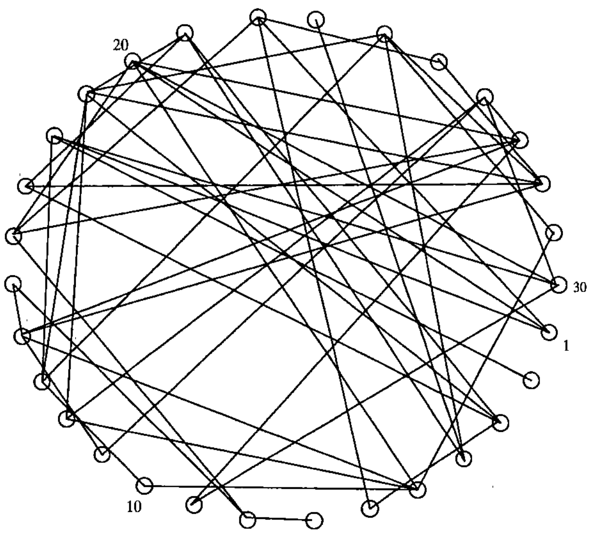


Fig. 2. The original graph with 30 vertex and 50 edges.

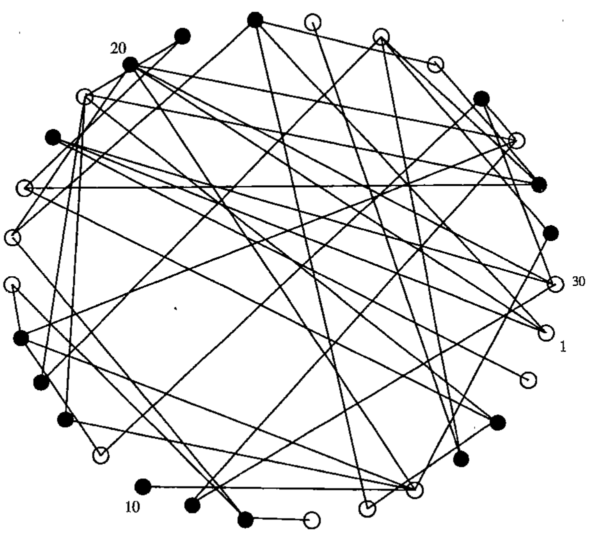


Fig. 3. Lee's solution with 42 edges embedded.

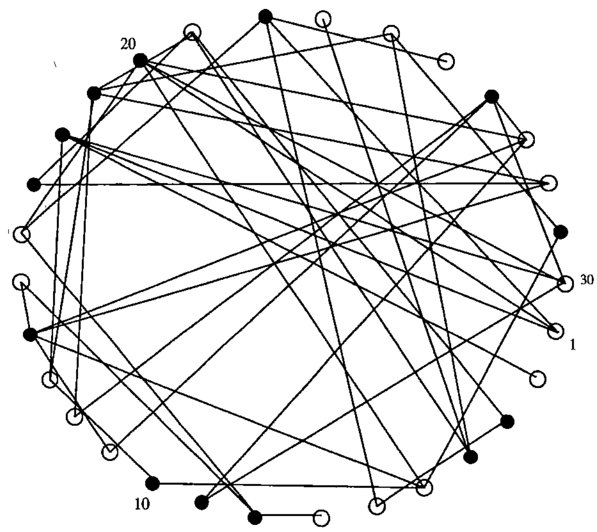


Fig. 4. The optimal bipartite subgraph with 43 edges embedded produced by proposed parallel algorithm.

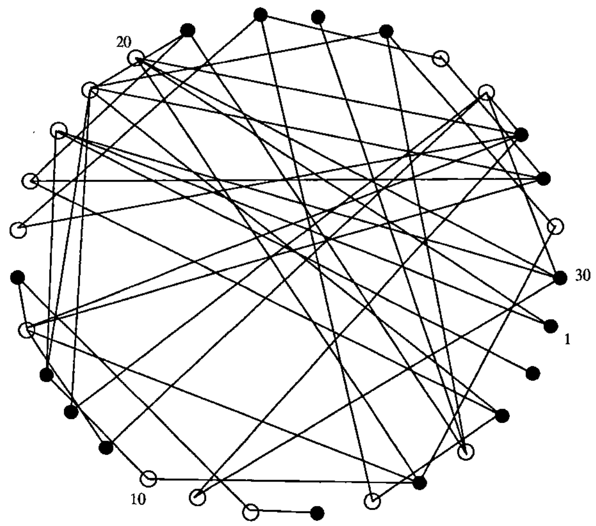


Fig. 5. One of the another optimal solutions for the graph in Fig.2.

and claimed that this maximum bipartite subgraph was the optimum solution by exhaustive search on $O(109)$ searching space [12]. Using the proposed parallel algorithm, we found a new maximum bipartite subgraph with 43 edges embedded. To see if our solution is a global optimal solution, we used exhaustive search to this graph on the full searching space, and found that our solution with 43 edges is a global optimal solution. Fig.2 shows the original graph of the 30 vertices and 50 edges graph problem. Fig.3 and Fig.4 show the solutions found by Lee et al.'s algorithm and our algorithm, respectively, where black circles (rectangle) and white circles represent two disjoint subsets for vertices. In our 100 simulations with different initial values of inputs to neurons, we found 9 different optimal solutions. Fig.5 shows one of the another optimal solutions. We also illustrated a typical progressive intermediate solution during the variation of K. In the calculation shown here, K was reduced to 0.08 in 990 iterations. Figures (Fig.6(a)-(f)) show the solutions found by our algorithm at K =0.60, 0.49, 0.32, 0.15, 0.08. Fig.6(a) has "no partition" and "double partition" violation, because vertices 3, 5, 7, 8, 9, 17, 20, 21, 24 were partitioned into both subsets, and vertices 6, 12, 16, 19, 25, 26 were not partitioned into either subsets. Fig.6(b) has "no partition" violation, because vertices 17, 24, 29 were not partitioned into either subset. Fig.6(c) and Fig.6(d) only had 40 edges and 42 edges, respectively, although the solutions fulfilled the constraint requirement for e. Fig.6(e) shows an optimal solution with 43 edges embedded when the scale constant K changes to a very small value (K=0.08).

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

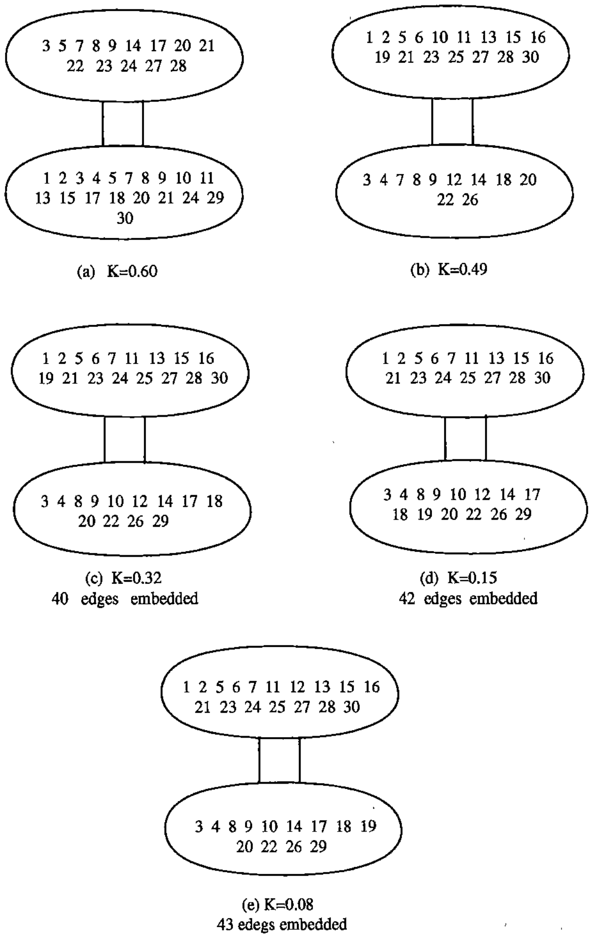


Fig. 6. Example of the progress of the proposed algorithm for the graph of Fig.2.

Table 1. Solutions produced by the algorithm of Lee et al. and our algorithm.

No.vertex	Probability	No.edges	Lee et al.	Propose algorithm
20	0.05	10	10	10
20	0.15	30	25	25
20	0.25	50	40	40
50	0.05	61	52	53
50	0.15	183	133	136
50	0.25	305	203	205
80	0.05	158	127	134
80	0.15	474	325	330
80	0.25	790	504	513
100	0.05	247	196	207
100	0.15	742	492	500
100	0.25	1235	761	777
150	0.05	558	402	419
150	0.15	1676	1062	1068
150	0.25	2790	1645	1683
200	0.05	995	685	715
200	0.15	2985	1838	1857
200	0.25	4975	2886	2918
250	0.05	1556	1060	1086
250	0.15	4668	2809	2832
250	0.25	7780	4435	4498
300	0.05	2242	1486	1510
300	0.15	6727	3987	4039
300	0.25	11212	6393	6417

algorithm. The simulation result were also compared with that found by Lee et al.'s algorithm. For each of instances, 100 simulation runs were performed. Information on the test graphs as well as all results are 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, respectively. Table 1 shows that the proposed parallel method could find a better solution than Lee et al.'s algorithm in all problems.

6. Conclusions

Enlightened by the elastic net method, we have demonstrated a successful parallel algorithm of solving the bipartite subgraph problem, and showed its effectiveness by simulation experiments. The algorithm used two different expressions for the constraint condition and the cost function in energy function. The former used a Guassian function, which is strongly dependent on the scale parameter K, and the latter used the cost function directly. Thus the network works to force the solution to satisfy the constraints first in the limit where K tends to zero, and finally get a local or global minimum. Specially, using this algorithm we were able to find a new maximum bipartite subgraph with 43 edges embedded in the graph with 30 vertices and 50 edges which is better than the best results [12] which were reported to be the "optimum solution" for the same problem. The simulation results also showed that the proposed parallel algorithm could provide better solutions than the best existing parallel algorithms for most tested graphs.

Acknowledgment

The work was supported in part by the grant-in-aid for Science Research of the Ministry of Education, Science and Culture of Japan under Grant (c)(2)12680392.

(Manuscript received Feb. 28, 2002,
revised Feb. 19, 2003)

References

(1) M.R. Garey, D.S. Johnson, and L.J. Stockmeyer: "Some simplified NP-complete graph problem", *Theor. Comput. Sci.*, Vol.1, pp.237–267 (1976)

(2) R.M. Karp: "Reducibility among combinatorial problems", Complexity of Computer Computations, Plenum, New York, pp.85–104 (1972)

(3) S. Even and Y. Shiloach: "NP-completeness of several arrangement problems", Technical Report 4, Department of computer Science, Technion, Haifa Israel (1975)

(4) J.A. Bondy and S.C. Locke: "Largest bipartite subgraph in triangle-free araphs with maximum degree three", J. Graph Theory, Vol.10, pp.477–504 (1986)

(5) M. Grotschel and W.R. Pulleyblank: "Weakly bipartite graphs and the max-cut problem", *Oper. Res. Lett.*, Vol.1, pp.23–27 (1981)

(6) F. Barahona: "On some weakly bipartite graph", *Oper. Res. Lett.*, Vol.2, No.5, pp.239–242 (1983)

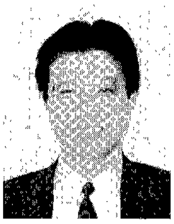
(7) J.J. Hopfield: "Neural networks and physical systems with emergentcollective computational abilities", *Proc. Nat. Acad. Sci.* U.S., Vol.81, pp.2554–2558 (1982)

(8) J.J. Hopfield: "Neurons with graded response have collective computational properties like those of two-state neurons", *Proc. Nat. Acad. Sci.* U.S., Vol.81, pp.3088–3092 (1984)

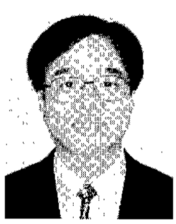
(9) J.J. Hopfield and D.W. Tank: "'neural' computation of decisions in optimization problems", *Bio. Cybern*, No.52, pp.142–152 (1985)

(10) J.J. Hopfield and D.W. Tank: "Simple 'neural' optimization networks: An a/d converter, signal decision circuit, and

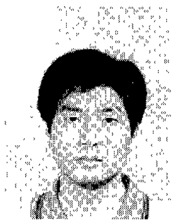
a linear programming circuit", *IEEE Trans, Circuits Syst.*, Vol.33, No.5, pp.333–543 (1986)
(11) J.J. Hopfield and D.W. Tank: "Computing with neural circuits: A model". *Science*, Vol.233, pp.625–633 (1986-8)
(12) K.C. Lee, N. Funabiki, and Y. Takefuji: "A parallel improvement algorithm for the bipartite subgraph problem", *IEEE Trans. Neural Networks*, Vol.3, No.1, pp.139–145 (1992)
(13) D.H. Ackley, G.E. Hinton, and T.J. Sejnowski: "A learning algorithm for Boltzman Machines", *Cognitive Science*, No.9, pp.147–169 (1985)
(14) J. Hertz, A. Krogh, and R.G. Palmer: Introduction to the Theory of Neural Computation, Addison Wesley Publishing Company (1991)
(15) R. Durbin and D. Willshaw: "An analogue approach of the traveling salesman problem using an elastic net method", *Nature*, 326:pp.689–691 (1987)
(16) R. Durbin, R. Szeliski, and A. Yuille: "An analysis of the elastic net approach to the traveling salesman problem," Neural Computation, Vol.1, pp.348–358 (1989)
(17) Y. Takefuji and K.C. Lee: "Artificial neural networks for four-coloring map problems and K-colorability problems", *IEEE Trans. Circuits Syst.*, Vol.38, No.3, pp.326–333 (1991-3)
(18) W.S. McCulloch and W.H. Pitts: "A logical calculus of ideas immanent in nervous activity", *Bull. Math. Biophys*, Vol.5, pp.115–133 (1943)



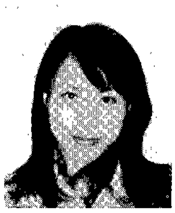
Zheng Tang (Non-member) received a B.S. degree from Zhejiang University, Zhejiang, China in 1982 and an M.S. degree and a D.E. degree from Tsinghua University, Beijing, China in 1984 and 1988, respectively. From 1988 to 1989 he was an Instructor in the Institute of Microelectronics at Tsinghua University. From 1990 to 1999, he was an Associate Professor in the Department of Electrical and Electronic Engineering, Miyazaki University, Miyazaki, Japan. In 2000, he joined Toyama University, Toyama, Japan, where he is currently a Professor in the Department of Intellectual Information Systems. His current research interests include intellectual information technology, neural networks, and optimizations.



Rong Long Wang (Member) received a B.S. degree from Hangzhou Teacher's College, Zhejiang, China and an M.S. degree from Liaoning University, Liaoning, China in 1987 and 1990, respectively. He received his D.E. degree from Toyama University, Toyama, Japan in 2003. From 1990 to 1998, he was an Instructor in Benxi University, Liaoning, China. In 2003, he joined Fukui University, Fukui Japan, where he is currently an Instructor in Department of Electrical and Electronics Engineering. His main research interests include neural networks and optimizations.



Jia Hai Wang (Non-member) received a B.S. degree from Gannan teacher's college, Jiangxi, China and an M.S. degree from Shandong University, Shandong, China in 1999 and 2001, respectively. From 2003 he is working toward the Ph.D degree at Toyama University, Toyama, Japan. His main research interests are neural networks and optimization problems.



Qi Ping Cao (Non-member) received a B.S. degree from Zhejiang University, Zhejing, China and an M.S. degree from Beijing University of Posts and Telecommunications, Beijing, China in 1983 and 1986, respectively. From 1987 to 1989, she was an Assistant Professor in the Institute of Microelectronics at Shanghai Jiaotong University, Shanghai, China. From 1989 to 1990, she was a Research Student in Nagoya University, Nagoya, Japan. From 1990 to 2000, she was joined Tateyama Systems Institue, Japan. Her current research interests include multiple-valued logic, neural nerworks, and optimizations.