Solving Combinatorial Optimization Problems Using Augmented Lagrange Chaotic Simulated Annealing

Lipo Wang, Fuyu Tian, Boon Hee Soong, and Chunru Wan*

School of Electrical and Electronic Engineering
Nanyang Technological University
Block S1, 50 Nanyang Avenue,
Singapore 639798
elpwang@ntu.edu.sg
http://www.ntu.edu.sg/home/elpwang
* Corresponding author

Abstract

Chaotic simulated annealing (CSA) proposed by Chen and Aihara has been successfully used to solve a variety of combinatorial optimization problems. CSA uses a penalty term to enforce solution validity as in the original Hopfield-Tank approach. There exists a conflict between solution quality and solution validity in the penalty approach. It is often difficult to adjust the relative magnitude of the penalty term, so as to achieve a high quality solution which is at the same time valid. To overcome this, we incorporate augmented Lagrange multipliers into CSA, obtaining a method that we call augmented Lagrange chaotic simulated annealing (AL-CSA). Simulation results on two constrained optimization benchmarks derived from the Hopfield-Tank formulation of the traveling salesman problem show that AL-CSA can maintain CSA's good solution quality while avoiding the potential difficulties associated with penalty terms. Furthermore, AL-CSA's convergence time is shorter and choice of system parameters is easier compared to CSA.

Keywords: chaotic neural network, Lagrange, simulated annealing, optimization, chaos, Hopfield

1. Introduction

Since Hopfield and Tank's seminal work on solving combinatorial optimization problem with neural networks, many researchers have attempted to improve solution quality and solution validity. Chen and Aihara [1-3] proposed an interesting method called chaotic simulated annealing (CSA) that can harness the advantage of both chaotic neurodynamics and conventional convergent neurodynamics. Reports show that CSA has richer dynamics and higher ability of searching for globally optimal or near-optimal solutions. In particular, Chen and Aihara's experimental studies demonstrated the effectiveness of CSA in obtaining good solutions for traveling salesman problems (TSPs) much more easily compared to the Hopfield-Tank approach. Earlier Nozawa showed [22][23] that a chaotic neural network has much higher searching ability for solving the TSP in comparison with the Hopfield-Tank approach, the Boltzmann machine [7], and the Gaussian machine. Chen and Aihara [24] provided theoretical conditions for the existence of chaos in these chaotic neural networks [14]-[21]. Since then, CSA has been successfully applied to a variety of combinatorial optimization problems [25], such as the channel assignment problem in mobile communications [26], CDMA multi-user detection [27][28], and image restoration [29].

However, in CSA there exists a difficulty in obtaining *solutions of good quality* and *valid solutions* simultaneously, similar to the original Hopfield-Tank approach. From a practical point of view, one needs to perform time-consuming, trial-and-error work to choose the relative sizes between the terms that represent solution quality and the terms that represent solution validity. This is because the original Hopfield-Tank approach and its modified models, including CSA, belong to the penalty method for constrained real optimization. On one hand, in order to converge to a feasible solution with the penalty method, the weighting factor for the penalty term must be sufficiently large. On the other hand, if the penalty term is too large, the role of the original objective function becomes relatively weak. Solutions found in this way may be more likely to be valid; however, the solutions may be less favorable in terms of the original optimization objective. Furthermore, if the penalty term is too large, the problem becomes ill-conditioned. Simulations show that CSA's solution quality and likelihood to find valid solutions are sensitive to the choice of penalty terms and other system parameters [20].

To completely avoid ill-conditioned problems and effectively eliminate the unfavorable influence of the penalty terms on solution quality, we shall in this paper incorporate augmented Lagrange multipliers [8][9][11] into CSA, obtaining a method that we call augmented Lagrange chaotic simulated annealing (AL-CSA). As detailed below, the difference between the present work and the previous papers by Zhang and Constantinides [9], Li [11] is that we include chaotic dynamics resulted from negative self-coupling of the neurons, whereas the previous papers [9][11] considered only Hopfield-type neural networks without negative self-coupling in the neurons or chaotic dynamics, and chaotic dynamics increase search capabilities compared to Hopfield-type gradient descent dynamics. With the Lagrange multipliers, the constraints are satisfied exactly without the need of penalty terms (Sections 2 and 3). To demonstrate AL-CSA, we shall conduct computer simulations on two constrained optimization benchmarks derived from the Hopfield-Tank formulation of the TSP (Section 4).

2. A Brief Review of the Augmented Lagrange Method

In this section, we present a brief review on the augmented Lagrange method for solving constrained optimization problems. A combinatorial optimization problem can be formulated as the following constrained minimization:

$$\text{minimize} \quad E(x) \tag{1}$$

$$\text{subject to} \quad C_p(x) = 0, \qquad p = 1,2,\ldots,P \quad . \tag{2}$$

Here $x = (x_1, x_2, \ldots, x_n)^T \in R^n, E : R^n \to R$, $E(x)$ is the objective function to be optimized (minimized). $C = (C_1, C_2, \ldots, C_P)^T$ is a set of real functions that represent some equality constraints and take the value of zero when the constraints are satisfied. $C : R^n \to R^P$. E and C are assumed to be twice continuously differentiable. P is the number of constraints. An augmented Lagrange function can be formed:

$$L_a(x,\lambda) = E(x) + \sum_{p=1}^{P} \lambda_p C_p(x) + \frac{1}{2}\sum_{p=1}^{P} a_p [C_p(x)]^2 \tag{3}$$

Here $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_P)^T$ is a set of Lagrange multipliers. $a_p > 0$, $p = 1,2,\ldots,P$ are finite weighting factors. The introduction of the quadratic term $\sum_{p=1}^{P} a_p [C_p(x)]^2$ with $C_p(x) = 0$ does not alter the location of the saddle points or the optima of the system. In fact, this term can effectively stabilize the system.

Then the first-order necessary condition of optimality can be expressed as a stationary point (x^*, λ^*) of $L_a(x,\lambda)$ over x and λ. That is,

$$\nabla_x L_a(x^*, \lambda^*) = \nabla f(x^*) + \nabla C(x^*)(\lambda^* + aC(x)) = 0 \qquad (4)$$

$$\nabla_\lambda L_a(x^*, \lambda^*) = C(x^*) = 0 \qquad (5)$$

The $n+P$ unknowns, $x_1^*, x_2^*, ..., x^*{}_n$ and $\lambda^*{}_1, \lambda_2{}^*, ..., \lambda^*{}_P$, can be solved from equations (4) and (5).

3. Augmented Lagrange Chaotic Simulated Annealing (AL-CSA)

Chen and Aihara proposed a transiently chaotic neural network (TCNN) by modifying a chaotic neural network which Aihara *et al* proposed earlier [6]. In contrast to the conventional stochastic simulated annealing (SSA), the optimization process of the TCNN is deterministically chaotic rather than stochastic, thus the TCNN is also called chaotic simulated annealing (CSA). CSA uses slow damping of the negative self-feedback to produce successive bifurcations so that the neurodynamics eventually converges from chaotic attractors to a stable equilibrium point.

CSA model is defined as follows [3]:

$$x_{ij}(t) = \frac{1}{1 + e^{-y_{ij}(t)/\varepsilon}} \qquad (6)$$

$$y_{ij}(t+1) = ky_{ij}(t) + \alpha(\sum_{k,l=1,k,l \neq i,j}^{n} w_{ijkl}x_{kl}(t) + I_{ij}) - z(t)(x_{ij}(t) - I_0) \qquad (7)$$

$$z(t+1) = (1-\beta)z(t) \qquad (i, j, k, l = 1, ..., n) \qquad (8)$$

where

x_{ij} $=$output of neuron i, j ;

$y_{i,j}$ $=$ internal state of neuron i, j ;

$I_{i,j}$ $=$ input bias of neuron i, j ;

k $=$damping factor of nerve membrane ($0 \le k \le 1$); α $=$positive scaling parameter for inputs;

$z(t)$ $=$self-feedback connection weight or refractory strength $(z(t) \ge 0)$;

β $=$damping factor of the time dependent $z(t)$ ($0 \le \beta \le 1$); I_0 $=$positive parameter;

ε $=$steepness parameter of the output function (ε >0);

$w_{ijkl} = w_{klij}$; $w_{ijij} = 0$; $\sum_{k,l=1,k,l\neq i,j}^{n} w_{ijkl} x_{kl} + I_{ij} = -\partial E' / \partial x_{ij}$ = connection weight from neuron k,l to neuron $i,\ j$;

E' = energy, which in CSA consists of the objective to be minimized (E in eq.(3)) and penalty terms representing constraints [3].

We propose the following augmented Lagrange chaotic simulated annealing (AL-CSA):

$$x_{ij}(t) = \frac{1}{1+e^{-y_{ij}(t)/\varepsilon}} \tag{9}$$

$$\begin{aligned} y_{ij}(t+1) &= k y_{ij}(t) - z(t)(x_{ij}(t) - I_0) \\ &\quad - \alpha \left\{ \frac{\partial E(x)}{\partial x_{ij}} + \sum_{p=1}^{P} \lambda_p \frac{\partial C_p(x)}{\partial x_{ij}} + \sum_{p=1}^{P} a_p C_p(x) \frac{\partial C_p(x)}{\partial x_{ij}} \right\} \end{aligned} \tag{10}$$

$$z(t+1) = (1-\beta) z(t) \tag{11}$$

$$\lambda_p(t+1) = \lambda_p(t) + a_p C_p(x_{ij}(t+1)) \tag{12}$$

In the experiments, a_p is usually increased to speed up convergence. So it can be changed into γa_p , where γ is a non-decreasing factor. In eq.(10), the energy does not include penalty terms associated with constraints and includes only the objective function to be minimized.

4. Computer Simulations

To demonstrate AL-CSA proposed in the previous section, we now apply the algorithm to the traveling salesman problem (TSP) that was used to demonstrate the original Hopfield-Tank approach [4] and CSA without augmented Lagrange [3][10][12][13].

Hopfield and Tank [4] mapped the solution of an *n*-city TSP to a network with $n \times n$ neurons. $x_{ij} = 1$ represents the fact that city i is visited in visiting order j, whereas $x_{ij} = 0$ represents that city i is not visited in visiting order j. The energy function representing the length of a tour is

$$E(x) = \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{\substack{k=1 \\ k \neq i}}^{n} d_{ik} x_{ij} (x_{kj+1} + x_{kj-1}) \quad , \qquad (13)$$

where d_{ik} is the distance between city i and city k [4][5]. We use the following equality constraints that were used by Hopfield and Tank, as well as many other researchers:

$$C_1(x) = \sum_{i=1}^{n} x_{ij} - 1 = 0, \forall j, \quad C_2(x) = \sum_{j=1}^{n} x_{ij} - 1 = 0, \forall i,$$

$$C_3(x) = \sum_{\substack{l=1 \\ l \neq j}}^{n} x_{ij} x_{il} = 0, \forall i, j, \quad C_4(x) = \sum_{k \neq i} x_{ij} x_{kj} = 0, \forall i, j, \qquad (14)$$

Hence the total number of constraints is $n + n + n^2 + n^2 = 2n^2 + 2n$. Eq. (10) for the TSP consists of the following:

$$\frac{\partial E(x)}{\partial x_{ij}} = \sum_{\substack{k=1 \\ k \neq i}}^{n} d_{ik} (x_{kj+1} + x_{kj-1}) \qquad (15)$$

$$\sum_{p=1}^{P} \lambda_p \frac{\partial C_p(x)}{\partial x_{ij}} = \lambda_1 + \lambda_2 + \lambda_3 \sum_{\substack{l=1 \\ l \neq j}}^{n} x_{il} + \lambda_4 \sum_{\substack{k=1 \\ k \neq i}}^{n} x_{kj} \qquad (16)$$

$$\sum_{p=1}^{P} a_p C_p(x) \frac{\partial C_p(x)}{\partial x_{ij}} = a_1 (\sum_{i=1}^{n} x_{ij} - 1) + a_2 (\sum_{j=1}^{n} x_{ij} - 1) + a_3 x_{ij} (\sum_{\substack{l=1 \\ l \neq j}}^{n} x_{il})^2 + \qquad (17)$$
$$a_4 x_{ij} (\sum_{\substack{k=1 \\ k \neq i}}^{n} x_{kj})^2$$

First we experiment on Hopfield-Tank’s 10-city TSP. Figure 1 is the known optimal route with length=2.691 [4,5].

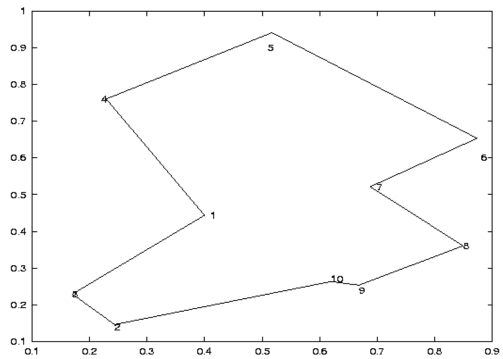

Figure 1. The optimal tour of Hopfield-Tank's 10 city TSP.

To compare the performance with CSA, we use a set of $k, \alpha, \beta, \varepsilon$ similar to Chen and Aihara's [3]:

$$k, \alpha, \beta, \varepsilon = 0.99, 0.01, 0.015, 1/250, I_0 = 0.65; z(0) = 0.8$$

The other parameters of AL-CSA we used are as follows:

$$\lambda_p(0) = 0,\ a_1 = a_2 = 0.05,\ \ a_3 = a_4 = 0.00001 \cdot$$

To increase the convergence speed, γ is increased from 0.1 to 10 (or 100) by 1% in each iteration, i.e. $\gamma \to 1.01\gamma$. In our experiment, one iteration means that all neuron states are cyclically updated once. The neuron outputs are discrete as in Chen and Aihara [3].

Our experiment reveals that the above parameters can be chosen quite flexibly in AL-CSA, i.e., the solutions of AL-CSA are not very sensitive to system parameters, especially their initial values. For example, in AL-CSA the self-feedback term $z(0)$ can be chosen from 0.08 to 0.8. On the contrary, $z(0)$ in CSA needs to be chosen carefully, since the constraint weights are fixed during all the evolution period [20].

Table 1. Results of 5000 runs with CSA and AL-CSA using independently selected random initial neuronal states.

	CSA	AL-CSA
Global minima	4946 (98.9%)	4952 (99.04%)
Local minima	31 (0.6%)	48 (0.96%)

Infeasible solutions	23 (0.5%)	0 (0%)

As shown in Table 1, among the 5000 cases with different initial conditions of neuron inputs generated randomly in the region [-1,1], 4952 cases in AL-CSA converge to global minima (tour length=2.691), the other 48 cases to local minima, such as tour length = 2.74, 2.78, etc. There are no infeasible solutions in AL-CSA. In contrast, CSA has a slightly lower rate of reaching local minima; however, it does lead to some infeasible solutions.

We also experimented on the 48-USA-city TSP that has been solved by Chen and Aihara [3]. CSA parameters we used are the same as Chen and Aihara's [3]:

$$k, \alpha, \beta, \varepsilon = 0.90, 0.015, 5\times10^{-5}, 1/250, I_0 = 0.50; z(0) = 0.10$$

The other AL-CSA parameters we used are as follows:

$$\lambda_p(0) = 1,\ a_1 = a_2 = 0.0003,\ \ a_3 = a_4 = 0.00001$$

We ran the simulations 100 times with different randomly selected initial network state using AL-CSA. Table 2 shows a comparison between our results and those of CSA (The data of CSA on the 48-city TSP come from [3]). AL-CSA can always converge to a feasible solution, i.e., with 0% invalid tour. In contrast, Chen and Aihara's CSA produces 5% invalid tours after selecting the penalty weighting. It is therefore difficult to compare average tour length with their results. If the invalid tours are not included in averaging, their average tour length is 10805.7, which is comparable to ours. Their average time of iterations is 25632, which is much longer compared to ours, i.e., AL-CSA converges much faster than CSA does when solving larger TSPs.

Table 2. Comparison of CSA and AL-CSA on a 48-city TSP.

	CSA	AL-CSA
Valid Tours	95%	100%
Invalid Tours	5%	0%
Average iterations	25632	12016

Average tour lengths	10805.7*	10839.0

*The 5% invalid tours were not included in the calculation.

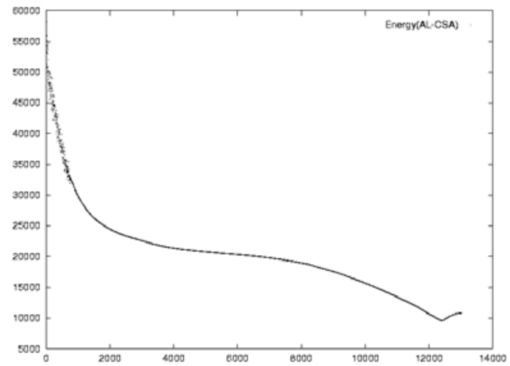

Figure 2. Lagrangian energy evolution process in AL-CSA.

Fig.2 shows the evolution of energy $L_a(x, \lambda)$ in equation (3). Its behavior is apparently different from the usual decreasing Lyapunov energy, which is due to the presence of the negative self-coupling in the neurons (a non-zero z in equation (7)) which takes away the monotonicity in the Hopfield dynamics. It is this non-monotonous behavior that helps the neural network to search for better optimal solutions. Fig.3 shows the Lyapunov energy (eq.(13)) evolution process in CSA.

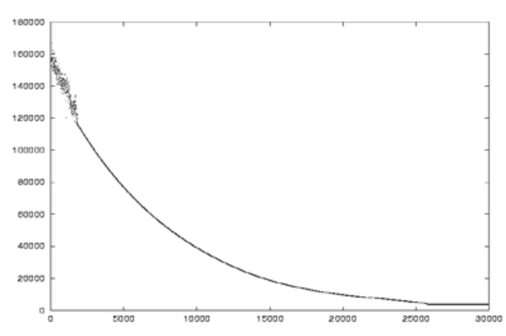

Figure 3. Lyapunov energy evolution process in CSA.

We note that Hopfield and Tank's prescription of mapping the TSP onto a neural network as described above, i.e., eqs.(13) and (14), is *not* the most effective way for solving the TSP using either neural networks [30] or chaotic dynamics [31]. Because of the need for n^2 neurons for an n-city TSP, the size of the TSP that can be handled by this prescription is limited. Other prescriptions specially tailored for the TSP can significantly increase the size of the TSP that can be handled (e.g., [30],[31]). In our present paper, we shall not attempt to adopt other mapping prescriptions in order to solve larger TSPs. Rather, the purpose of the present work is to demonstrate the improvements of AL-CSA over CSA for a given objective function and a given set of constraints. In other words, neither the 10-city TSP nor the 48-city TSP studied in this paper may be considered difficult; however, *minimizing the objective functions given by eq.(13) together with the constraints given in eq.(14) and the parameters drawn from the 10-city TSP and the 48-city TSP* is indeed non-trivial and can therefore be used as benchmarking optimization problems to compare various optimization algorithms, such as CSA and AL-CSA.

5. Conclusion

In this paper we have incorporated augmented Lagrange multipliers into Chen and Aihara's CSA model, obtaining a method that we call augmented Lagrange chaotic simulated annealing (AL-CSA). Compared to CSA, AL-CSA eliminates the need for trial-and-error adjustments of the weighting of the penalty terms. We demonstrated the approach using two constrained optimization benchmarks. Our experiments showed that the neural network converges faster and the results are less sensitive to system parameters in comparison with CSA. The solution validity can always be guaranteed, which is quite desirable for real-world applications.

References

[1] L. Chen and K. Aihara, "Chaotic simulated annealing for combinatorial optimization," N. Aoki et al (eds.), *Dynamical Systems and Chaos*, vol. 1, (Singapore, World Scientific, 1994) pp. 319-322.

[2] L. Chen and K. Aihara, "Transient chaotic neural networks and chaotic simulated annealing," M. Yamguti (ed.), *Towards the Harnessing of Chaos*. (Amsterdam, Elsevier Science Publishers, 1994) pp. 347-352.

[3] L. Chen and K. Aihara, “Chaotic simulated annealing by a neural network model with transient chaos,” *Neural Networks*, vol. 8, no. 6, (1995) pp. 915-930.

[4] J. J. Hopfield and D. W. Tank, “Neural computation of decisions in optimization problems,” *Biological Cybernetics*, vol. 52, (1985) pp. 141-152.

[5] G.V. Wilson and G. S. Pawley, “On the stability of the traveling salesman problem algorithm of Hopfield and Tank,” *Biological Cybernetics*, vol. 58, (1988) pp. 63.

[6] K. Aihara, T. Takabe and M. Toyoda, “Chaotic neural networks,” *Physics Letters A*, vol. 144, no. 6, 7, (1990) pp. 333-340.

[7] E. Aarts and J. Korst, *Simulated Annealing and Boltzmann Machines*, Chichester: John Wiley, 1989.

[8] D. P. Bertsekas, “*Constrained optimization and Lagrange multiplier methods*,” Academic Press, New York, 1982.

[9] S. Zhang and A. G. Constantinides, “Lagrange programming neural networks,” *IEEE Transactions on Circuits and Systems-II: Analog and Digital Signal processing*, vol. 39, no. 7, pp. 441-452, 1992.

[10] G. Reinelt, “TSPLIB-A traveling salesman library,” ORSA Journal on Computing, vol. 3, pp. 376-384, 1991.

[11] S. Z. Li, “Improving convergence and solution quality of Hopfield-type neural networks with augmented Lagrange multipliers,” *IEEE Transactions on Neural Networks*, vol. 7, no. 6, (1996) pp. 1-14.

[12] R. D. Brandt, Y. Yang, A. J. Laub and S. K. Mitra, “Alternative networks for solving the travelling salesman problem and the list-matching problem,” in *IEEE International Conference on Neural Networks*, vol. 2, (1988) pp. 333-340.

[13] S. Abe, “Theories on the Hopfield neural networks,” in *Proc. IEEE INNS International Joint Conference on Neural Networks*, vol. 1, (1989) pp. 557-564.

[14] L. Wang and D.L. Alkon, (eds.), *Artificial neural networks: Oscillations, chaos, and sequence processing*. (IEEE Computer Society Press, Los Alamitos, California, 1993).

[15] L. Wang, “Oscillatory and chaotic dynamics in neural networks under varying operating conditions,” *IEEE Transactions on Neural Networks*, vol.7, no. 6, (1996) pp. 1382-1388.

[16] T. Kwok, K. Smith and L. Wang, “Solving combinatorial optimization problems by chaotic neural networks,” C. dagli et al. (eds.) *Intelligent Engineering Systems through Artificial Neural Networks*, vol. 8, (1998) pp. 317-322.

[17] T. Kwok, K. Smith and L. Wang, “Incorporating chaos into the Hopfield neural network for combinatorial optimization,” in *Proc. 1998 World Multiconference on Systemics, Cybernetics and Informatics*, N. Callaos, O. Omolayole, and L. Wang, (eds.) vol.1, (1998) pp. 646-65.

[18] L. Wang and K. Smith, “Chaos in the discretized analog Hopfield neural network and potential applications to optimization,” *Proc. International Joint Conference on Neural Networks*, vol. 2, (1998) pp. 1679-1684.

[19] L. Wang and K. Smith, “On chaotic simulated annealing,” *IEEE Transactions on Neural Networks*, vol. 9, no. 4, (1998) pp. 716-718.

[20] F. Tian and L. Wang, “Solving optimization problems using transiently chaotic neural networks: choices of network parameters,” *Proc. Second International Conference on Information, Communications & Signal Processing* (Singapore, Dec 7-10, 1999).

[21] F. Tian and L. Wang, “Chaotic simulated annealing with augmented Lagrange for solving combinatorial optimization problems,” *Proc. th*e *Third Asia-Pacific Conference on Simulated Evolution And* Learning, (Japan, 25-27 October 2000).

[22] H. Nozawa, "A neural network model as a globally coupled map and applications based on chaos," *Chaos*, vol.2, no.3, p. 377-86, 1992.

[23] H. Nozawa, "Solution of the optimization problem using the neural network model as a globally coupled map," in *Towars the Harnessing of Chaos*, M. Yamaguti (ed), pp. 99-114, 1994.

[24] L. Chen and K. Aihara, "Global searching ability of chaotic neural networks." *IEEE Trans. Circuits and Systems - I: Fundamental Theory and Applications*, vol. 46, no. 8, August, pp. 974 - 993, 1999.

[25] K. Aihara, “Chaos engineering and its application to parallel distributed processing with chaotic neural networks,” *Proceedings of the IEEE*, Vol.90 , no.5 , pp.919 – 930, May 2002.

[26] Lipo Wang, Sa Li, Chunru Wan, and Boon Hee Soong, “Minimizing interference in cellular mobile communications by optimal channel assignment using chaotic simulated annealing”, in *Soft Computing in Communications*, Lipo Wang (Editor), Springer-Verlag, pp.131-145, 2004.

[27] Baoyun Wang, Jingnan Nie, and Zhenya He, “A transiently chaotic neural-network implementation of the CDMA multiuser detector,” *IEEE Transactions on Neural Networks*, Vo.10, no.5 , pp.1257 – 1259, Sept. 1999.

[28] W. Zhong and S.-X;Cheng, “Multiuser detection using time-varying scaling-parameter transiently chaotic neural networks,” Electronics Letters, Vol.35 , no.12 , pp.987 – 989, 10 Jun 1999.

[29] Leipo Yan and Lipo Wang, "Image restoration with chaotic simulated annealing", *Proceedings of the 2003 IEEE International Joint Conference on Neural Networks* (Portland, USA, July 20-24, 2003), pp. 3060-3064, 2003.

[30] K. Tanaka, Y. Horio, and K. Aihara, “A modified algorithm for the quadratic assignment problem using chaotic-neuro-dynamics for VLSI implementation,” *Proc. International Joint Conference on Neural Networks (IJCNN '01)*, vol.1, pp. 240 -245, 2001.

[31] M. Hasegawa, T. Ikeguchi, and K. Aihara, “Combination of chaotic neurodynamics with the 2-opt algorithm to solve traveling salesman problems,” *Phys. Rev. Lett*, vol. 79, pp. 2344-2347, 1997.