Augmented Lagrange Chaotic Simulated Annealing For Combinatorial Optimization Problems

Fuyu Tian and Lipo Wang

School of Electrical and Electronic Engineering
Nanyang Technological University
Block S2, Nanyang Avenue, Singapore 639798
Email: elpwang@ntu.edu.sg
URL: http://www.ntu.edu.sg/home/elpwang

Abstract – **Chaotic simulated annealing (CSA) has recently been proposed and successfully used in solving combinatorial optimization problems by Chen and Aihara. In comparison with the Hopfield-Tank approach, CSA significantly improves the network's ability to find solutions of good quality and even global minima. However, CSA still uses a penalty term to enforce solution validity like the Hopfield-Tank approach. There exists a conflict between solution quality and solution validity in the penalty approach. In addition, the relative magnitude of the penalty term often needs to be determined by trial-and-error. In this paper we incorporate augmented Lagrange multipliers into CSA, obtaining a method that we call augmented Lagrange chaotic simulated annealing (AL-CSA), which eliminates the need of the penalty term and guarantees solution validity, and at the same time maintains CSA's solution quality. We demonstrate this method with the 10-city Traveling Salesman Problem.**

1. Introduction

Following Hopfield and Tank's seminal work on solving combinatorial optimization problem with neural network, a lot of efforts have been made to improve the solution quality and solution validity [1-4]. Recently, Chen and Aihara proposed a chaotic simulated annealing (CSA) method that can harness the advantage of both chaotic neurodynamics and conventional convergent neurodynamics [5-7]. They reported that the CSA has richer dynamics and higher ability of searching for globally optimal or near-optimal solutions. Their experimental studies demonstrate the effectiveness of the CSA and obtain good solutions for TSPs with a set of parameters chosen [5-7] much more easily compared to the Hopfield-Tank approach [1][2].

However, in CSA there still exists a difficulty in obtaining solutions of *good quality* and *valid* solutions simultaneously, and one still needs to perform time-consuming, trial-and-error work to choose the relative sizes between the terms that represent solution quality and solution validity. From the optimization viewpoint, the original Hopfield-Tank approach and its modified versions, as well as CSA, essentially belong to the penalty method for constrained real optimization. In order to converge to a feasible solution with the penalty method, the weighting factor for the penalty term must be sufficiently large. However, as the penalty term becomes stronger, the role of the original objective function becomes relatively weaker. Solutions found in this way are affected more by the penalty terms and hence are less favorable in terms of the original objective. Furthermore, when the penalty term is too larger, the problem becomes ill-conditioned. This is a typical problem with the penalty method and explains why it is difficult to obtain good quality solutions and valid solutions simultaneously with Hopfield-type networks [11].

Li [11] introduced an augmented Lagrange method to Hopfield-Tank approach (ALH) and showed its superiority over the existing Hopfield-type neural networks in solution validity and solution quality. However, the ALH is similar to Hopfield-Tank-type methods in that it carries out a gradient decent in the energy landscape and may therefore often be trapped in local minima. In addition, the convergence time was long [11].

In this paper we incorporate augmented Lagrange multipliers into the CSA to overcome the difficulties in the CSA and the ALH [8-11], obtaining a method that we call AL-CSA. With the Lagrange multipliers, the constraints are satisfied exactly without the need of penalty terms. This not only avoids ill-conditioned problems but also effectively eliminates the unfavorable influence of the penalty terms on solution quality. With the CSA, the convergence time is shortened because of the efficient searching ability of chaotic dynamics.

0-7695-0619-4/00 $10.00 © 2000 IEEE

2. The Augmented Lagrange (AL) Method

A combinatorial optimization problem can be converted into the following constrained minimization:

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

$$\text{subject to} \quad C_k(x) = 0, \qquad k = 1,\dots K \tag{2}$$

where $x = (x_1, x_2, \dots, x_n)^T \in R^n, E: R^n \to R$, $E(x)$ is the energy function, $C: R^n \to R^K \cdot C = (C_1, C_2, \dots, C_K)^T$ are real functions that represent some equality constraints and take the value of zero when the constraints are satisfied. E and C are assumed to be twice continuously differentiable. An augmented Lagrange function can be formed:

$$L_a(x,\lambda) = E(x) + \sum_k \lambda_k C_k(x) + \frac{1}{2}\sum_k a_k [C_k(x)]^2 \tag{3}$$

where $\lambda = (\lambda_1, \lambda_2, \dots, \lambda_K)^T$ are Lagrange multipliers. $a_k > 0$ are finite weighting factors. The introduction of the quadratic term $\sum_k a_k [C_k(x)]^2$ with $C_k(x) = 0$ doesn't alter the location of the saddle point. In fact, this term can effectively stabilize the system.

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

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

$$\nabla_\lambda L_a(x^*, \lambda^*) = C(x^*) = 0 \tag{5}$$

The $n+k$ unknowns, $x_1^*, x_2^*, \dots, x^*_n$ and $\lambda^*_1, \lambda_2^*, \dots, \lambda^*_K$ can be solved by the equation (4-5).

3. Chaotic Simulated Annealing (CSA)

Chen and Aihara recently proposed a transiently chaotic neural network (TCNN) by modifying a chaotic neural network which Aihara et al proposed earlier [4][5]. In contrast to the conventional stochastic simulated annealing (SSA), the optimization process of the TCNN is deterministically chaotic rather than stochastic, the TCNN is also called chaotic simulated annealing (CSA). The CSA uses slow damping of the negative self-feedback to produce successive bifurcations so that the neurodynamics eventually converges from strange attractors to a stable equilibrium point.

The CSA model is defined as follows:

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

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

$$z(t+1) = (1-\beta) z(t) \qquad (i, j, k, l = 1, \dots, n) \tag{8}$$

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 ($0 \le \beta \le 1$); I_0 = positive parameter;

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

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

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

From (3-5), it is easy to write the following dynamic equations for a saddle point of L_a:

$$\frac{dx_{ij}}{dt} = -\frac{\partial L_a(x,\lambda)}{\partial x_{ij}} = -\frac{\partial E(x)}{\partial x_{ij}} - \sum_k \lambda_k \frac{\partial C_k(x)}{\partial x_{ij}} - \sum_k a_k C_k(x) \frac{\partial C_k(x)}{\partial x_{ij}} \quad (9)$$

$$\frac{d\lambda_k}{dt} = +\frac{\partial L_a(x,\lambda)}{\partial \lambda_k} = +C_k(x) \quad (10)$$

The augmented Lagrange Chaotic Simulated Annealing (AL-CSA) can be written as follows:

$$x_{ij}(t) = \frac{1}{1+e^{-y_{ij}(t)/\varepsilon}} \quad (11)$$

$$y_{ij}(t+1) = ky_{ij}(t) - z(t)(x_{ij}(t) - I_0) - \alpha\left\{\frac{\partial E(x)}{\partial x_{ij}} + \sum_k \lambda_k \frac{\partial C_k(x)}{\partial x_{ij}} + \sum_k a_k C_k(x) \frac{\partial C_k(x)}{\partial x_{ij}}\right\} \quad (12)$$

$$z(t+1) = (1-\beta)z(t) \quad (13)$$

$$\lambda_k(t+1) = \lambda_k(t) + a_k C_k(x_{ij}(t+1)) \quad (14)$$

In the above, $k,\alpha,\beta,\varepsilon$ are CSA's damping factor of nerve membrane $(0 \le k \le 1)$, positive scaling parameter for inputs, damping factor of the self-feedback connection weight $z(t)$ $(0 \le \beta \le 1)$ and steepness parameter of the output function, respectively. a_k is usually increased to speed up convergence. So it can be changed into γa_k , where γ is a non-decreasing factor for the penalty term. The λ_k are Lagarange multipliers.

5. Solving TSP Using AL-CSA

In the Traveling Salesman Problem (TSP), the energy function [1][2] is $E(x) = \frac{1}{2}\sum_i \sum_j \sum_{k \ne i} d_{ik} x_{ij} (x_{kj+1} + x_{kj-1})$.

We use the following equality constraints that were used by Hopfield-Tank and other researchers:

$$C_1(x) = \sum_i x_{ij} - 1 = 0, \forall j \quad (15)$$

$$C_2(x) = \sum_j x_{ij} - 1 = 0, \forall i \quad (16)$$

$$C_3(x) = \sum_{l \ne j} x_{ij} x_{il} = 0, \forall i, j \quad (17)$$

$$C_4(x) = \sum_{k \ne i} x_{ij} x_{kj} = 0, \forall i, j \quad (18)$$

$$C_5(x) = x_{ij}(1 - x_{ij}) = 0, \forall i, j \quad (19)$$

Hence the total number of constraints is $n + n + n^2 + n^2 + n^2 = 3n^2 + 2n$. Here n is the city number of TSP, as shown in equation (1).

The three terms of (12) for the TSP are as follows:

$$\frac{\partial E(x)}{\partial x_{ij}} = \sum_{k \ne i} d_{ik}(x_{kj+1} + x_{kj-1}) \quad (20)$$

$$\sum_k \lambda_k \frac{\partial C_k(x)}{\partial x_{ij}} = \lambda_1 + \lambda_2 + \lambda_3 \sum_{l \ne j} x_{il} + \lambda_4 \sum_{k \ne i} x_{kj} + \lambda_5(1 - 2x_{ij}) \quad (21)$$

$$\sum_k a_k C_k(x) \frac{\partial C_k(x)}{\partial x_{ij}} = a_1(\sum_i x_{ij} - 1) + a_2(\sum_j x_{ij} - 1) + a_3 x_{ij}(\sum_{l \ne j} x_{il})^2 + a_4 x_{ij}(\sum_{k \ne i} x_{kj})^2 + a_5 x_{ij}(1 - x_{ij})(1 - 2x_{ij}) \quad (22)$$

We experiment on Hopfield-Tank's 10-city TSP that has been widely tackled by other neural network algorithms [1,2]. To compare the performance with the CSA, we use a set of $k, \alpha, \beta, \varepsilon$ similar to Chen and Aihara's [7]:

$$k, \alpha, \beta, \varepsilon = 0.99, 0.01, 0.015, 0.004; \ I_0 = 0.65; z(0) = 0.8 .$$

The other parameters can be chosen as follows:

$\lambda_k(0) = 0$; γ is increased from 0.1 to 10 (or 100) according to $\gamma \to 1.01\gamma$.

$a_1 = a_2 = 0.05$; $\quad a_3 = a_4 = 0.00001$, $\quad a_5 = 0$;

It should be noted that the above parameters can be chosen quite flexibly. The simulation results show that the AL-CSA is not sensitive to these parameters, especially the initial values.

In the experiment, one iteration means that all neuron states are cyclically updated once. The neuron outputs are discrete as used by Chen and Aihara [7].

As shown in Table 1, AL-CSA converges on average in 450 iterations when $\beta = 0.015$. Among the 5000 cases with different initial conditions of y_{ij} generated randomly in the region [-1,1]. 4952 cases 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.

Table 1 Results of 5000 different cases for different γ with AL-CSA

	$\gamma \to 10$	$\gamma \to 100$
Global minima	4952 (99.04%)	4963 (99.26%)
Local minima	48 (0.96%)	37 (0.74%)
Infeasible solutions	0	0
Average iterations for convergence	458	451

$\gamma \to 10$ means γ is increased from an initial value to 10.

Table 2 Results of 5000 different cases for different β with CSA

	$\beta = 0.015$	$\beta = 0.010$
Global minima	4946 (98.9%)	4969 (99.4%)
Local minima	31 (0.6%)	13 (0.3%)
Infeasible solutions	23 (0.5%)	18 (0.3%)
Average iterations for convergence	81	119

Compared with Chen and Aihara's CSA [7] (see Table 2), AL-CSA needs more iterations to converge. But AL-CSA can ensure a feasible solution found while the rate of global minima is still comparable with CSA. By the augmentd Lagrange method introduced, the time-consuming trial-and-error process for finding penalty parameters can be avoided.

In addition, the self-feedback $z(0)$ in CSA needs to be chosen carefully, since the constraint weights are fixed [21]. On the contrary, the Lagrange multipliers λ_k are time-dependent and the performance of AL-CSA depends on the choice of $z(0)$ less sensitively compared to CSA.

Compared with Li's ALH [11], in which the average iteration times are over 10000, the AL-CSA converges much faster, because chaotic dynamics can perform efficient searching.

6. Summary

In this paper we incorporate augmented Lagrange multipliers into Chen and Aihara's CSA, obtaining a method that we call augmented Lagrange chaotic simulated annealing (AL-CSA). AL-CSA eliminates the need of the penalty terms and guarantees solution validity. Our simulations on the Hopfield-Tank's 10-city TSP show that AL-CSA can also maintain the CSA's solution quality at the same time.

The AL-CSA network is more complicated than the CSA because of the additional Lagrange neurons. However, it is these auxiliary neurons that help to achieve the improvements over the CSA; Compared with Li's ALH, it is the slow damping of the negative self-feedback in the AL-CSA that helps to improve the searching efficiency for optimal or near-optimal solutions.

In our previous work, we found that the solution quality of Chen and Aihara's CSA is sensitive to the choices of the penalty terms when the city number of TSP becomes larger [21]. Our future work will apply the AL-CSA to larger and practical optimization problems.

References

[1] J. J. Hopfield and D. W. Tank, "Neural computation of decisions in optimization problems," *Biological Cybernetics*, vol. 52, pp. 141-152, 1985.

[2] 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, 1988.

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

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

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

[6] 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 B.V. pp.347-352, 1994.

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

[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] B. T. Polyak, "*Introduction to optimization*," Optimization Software, Inc., Publications Division. 1987.

[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, pp. 1-14, 1996.

[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, pp. 333-340, 1988.

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

[14] L. Wang and J. Ross, "Osillations and chaos in neural networks: an exactly solvable model." *Proc. National Academy of Sciences* (USA) vol. 87, pp. 9467-9471, 1990.

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

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

[17] 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, pp. 317-322, 1998.

[18] 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, pp. 646-651, 1998.

[19] 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, pp. 1679-1684, 1998.

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

[21] 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.