

## Adaptive annealing for chaotic optimization

Isao Tokuda,<sup>1</sup> Kazuyuki Aihara,<sup>2</sup> and Tomomasa Nagashima<sup>1</sup>

<sup>1</sup>Department of Computer Science and Systems Engineering, Muroran Institute of Technology, Muroran, Hokkaido 050, Japan

<sup>2</sup>Department of Mathematical Engineering and Information Physics, Faculty of Engineering, The University of Tokyo, Bunkyo-ku, Tokyo 113, Japan

(Received 26 January 1998)

The chaotic simulated annealing algorithm for combinatorial optimization problems is examined in the light of the global bifurcation structure of the chaotic neural networks. We show that the result of the chaotic simulated annealing algorithm is primarily dependent upon the global bifurcation structure of the chaotic neural networks and unlike the stochastic simulated annealing *infinitely slow* chaotic annealing does not necessarily provide an optimum result. As an improved algorithm, the adaptive chaotic simulated annealing algorithm is introduced. Using several instances of 20- and 40-city traveling salesman problems, efficiency of the adaptive algorithm is demonstrated. [S1063-651X(98)15510-1]

PACS number(s): 05.45.+b, 07.05.Mh

Due to the recent successful results in a variety of engineering problems, application of chaotic neural networks [1] to combinatorial optimization problems has received a great deal of attention [2–8]. The idea of optimization by chaotic neural networks can be briefly summarized as follows. In the continuous state space of the chaotic neural network, every possible solution of an optimization problem is embedded. By following a chaotic wandering orbit which visits a variety of the solutions, chaotic dynamics continually searches for the optimum solution. In contrast with the conventional Hopfield-Tank neural network search [9], the nonequilibrium chaotic search overcomes the local minimum problem. Compared to the stochastic search system [10,11] whose search space is essentially the same with the whole state space of the search dynamical system, the chaotic search dynamics is confined in a relatively low-dimensional fractal space, which seems to realize an efficient search for a variety of optimization problems such as the traveling salesman problem (TSP) [2–7].

For the chaotic neural network approach to optimization problems, it is natural to introduce the idea of *simulated annealing* [10]. Technically, it is quite important to gradually cool down the chaotic dynamics to a possibly optimum state by simulated annealing, since the chaotic search is basically everlasting.

Towards the simulated annealing in a chaotic neural network, the chaotic simulated annealing (CSA) algorithm has been recently developed by Chen and Aihara [3]. In the CSA algorithm, the chaotic dynamics is harnessed by a cooling algorithm of a bifurcation parameter. Gradual cooling of the bifurcation parameter controls the chaotic search dynamics to converge to a stable equilibrium state with a possibly optimum or near-optimum solution. The experimental studies in Ref. [3] demonstrate the efficiency of the CSA algorithm which obtains fairly good solutions of TSPs.

The aim of the present study is to reexamine the efficiency of the CSA algorithm in the light of the global bifurcation structure of the chaotic neural networks. On the basis of our bifurcation studies [6,7], we argue that the result of the chaotic annealing is primarily dependent upon the global bifurcation structure of the chaotic neural networks and show

that *infinitely slow* chaotic annealing does not necessarily provide an optimum result. As an improved algorithm, an adaptive chaotic simulated annealing algorithm is then introduced.

As an example of a combinatorial optimization problem, we consider an  $N$ -city symmetric TSP [12]: “Given an  $N \times N$  symmetric matrix  $(d_{ij})$  of distances between a set of  $N$  cities  $(i, j = 1, 2, \dots, N)$ , find a minimum-length tour that visits each city exactly once.”

The chaotic neural network which solves the TSP is described in terms of an  $(N \times N)$ -dimensional mapping [2,6]:

$$p_{ik}(n+1) = rp_{ik}(n) + (1-r)\sigma\left(\sum_{j=1}^N \sum_{l=1}^N T_{ik,jl}p_{jl}(n) + I_{ik}\right), \quad (1)$$

where  $p_{ik}$  stands for an internal state of the  $(i, k)$  neuron  $(i, k = 1, \dots, N)$ ,  $r(0 < r < 1)$  stands for a decay parameter, and  $\sigma(x) = 0.5 + 0.5 \tanh(x/\epsilon)$ . The synaptic connections  $T_{ik,jl}$  are given by

$$T_{ik,jl} = -A[\delta_{ij}(1 - \delta_{kl}) + \delta_{kl}(1 - \delta_{ij})] - Bd_{ij}(\delta_{lk+1} + \delta_{lk-1}), \quad (2)$$

$$T_{ik,ik} = -2\omega A, \quad (3)$$

$$I_{ik} = 2\alpha A, \quad (4)$$

where  $A$  and  $B$  are balancing parameters of the constraint term and the tour-length term of the TSP cost function [9],  $\alpha$  is a control parameter for excitation level of neurons, and  $\omega$  is a negative self-feedback parameter.

By wandering around a variety of temporal network firing states [2]  $\{\rho_{ik}(n) = (1/w)\sum_{j=0}^{w-1} p_{ik}(n-j) | i, k = 1, \dots, N\}$  ( $w$ : averaging duration), which are coded into possible TSP solutions  $J(n) = \{J_{ik}(n) | i, k = 1, \dots, N\}$  by  $J_{ik}(n) = 1[\rho_{ik}(n) - \rho^*]$  ( $1[x] = 1$  if  $x \geq 0$ ,  $1[x] = 0$  if  $x < 0$ ,  $\rho^*$ : the  $N$ th largest among  $\{\rho_{ik}(n)\}$ ), the chaotic neural network searches for the optimum solution among a variety of TSP solutions of  $\{J(n)\}$ .

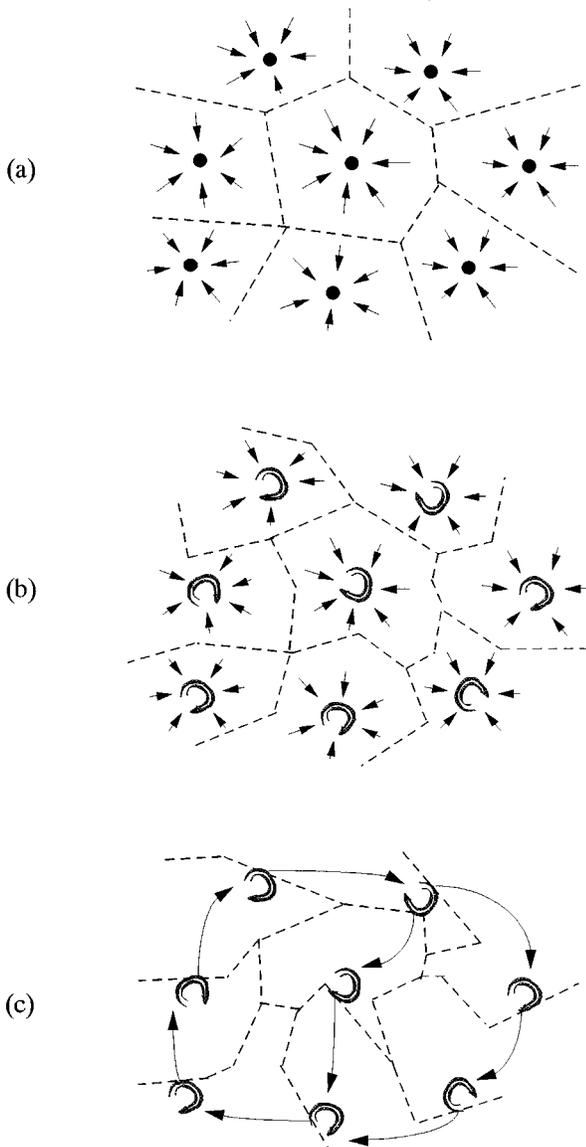


FIG. 1. Global bifurcation scenario for chaotic neural networks: (a) In a parameter region close to  $r=1$ , the chaotic neural network exhibits Hopfield-Tank “convergence” dynamics to many local minima. (b) With a decrease in the bifurcation parameter  $r$ , the local minima bifurcate into chaotic attractors through a period-doubling route. (c) The coexisting chaotic attractors eventually merge into a single global attractor.

On the basis of the numerical studies, we have obtained the following global bifurcation scenario for the chaotic neural networks which solve TSP [6] (see Fig. 1): We take the decay parameter  $r$  as the bifurcation parameter. First, there is a bifurcation parameter region with  $r \approx 1$ , where nonlinear dynamics of the chaotic neural network (1) becomes similar to that of the continuous-time Hopfield-Tank neural network which exhibits “convergence” dynamics to the local minimum solutions of the underlying Lyapunov function [4]. With a decrease in the bifurcation parameter  $r$ , the local minimum solutions bifurcate into chaotic attractors through a period-doubling route. The chaotic attractors are initially localized in the state space and eventually merge into a single global attractor via a series of crises [13]. The merging process gives rise to intermittent switch dynamics among the

previous localized chaotic attractors and the global “chaotic search” dynamics for various TSP solutions takes place.

Our bifurcation scenario provides the following insights. (i) Crisis-induced intermittent switches among the ruins of the previous localized chaotic attractors are the dynamical bases of the “chaotic search.” This type of a dynamical phenomenon is called *chaotic itinerancy* [14–16], and has been observed in a variety of high-dimensional dynamical systems. (ii) The bifurcation scenario provides a guideline for tuning the bifurcation parameter value which gives rise to an efficient “chaotic search.” (iii) The CSA algorithm is primarily dependent upon the global bifurcation structure of the chaotic neural networks and the efficiency of the CSA algorithm can be examined by our bifurcation scenario.

In the CSA algorithm [3], in order to terminate the chaotic search procedure and to obtain the final solution, the chaotic dynamics is eventually controlled to converge to a stable equilibrium state by a gradual cooling of the bifurcation parameter. On the basis of our bifurcation scenario, the CSA algorithm for TSP can be formulated as the following dynamics:

$$p_{ik}(n+1) = r(n)p_{ik}(n) + [1 - r(n)] \times \sigma \left( \sum_{j=1}^N \sum_{l=1}^N T_{ik,jl} p_{jl}(n) + I_{ik} \right), \quad (5)$$

$$r(n+1) = (1 - \beta)[r(n) - r_g] + r_g, \quad (6)$$

where  $\beta$  ( $0 < \beta < 1$ ) stands for an annealing speed parameter and  $r_g$  stands for a bifurcation parameter value which gives rise to Hopfield-Tank “convergence” dynamics.

With an initial condition of random  $p(0) \in [0,1]^{N \times N}$  and  $r(0) = r_s$ , where  $r_s$  stands for a bifurcation parameter value which gives rise to “chaotic search” dynamics, at the first stage of the annealing, the network searches for TSP solutions by chaotic wandering dynamics. As the annealing proceeds with  $r(n) \rightarrow r_g$ , the chaotic search dynamics eventually converges to a single equilibrium solution.

Figure 2 shows the result of the CSA algorithm applied to a five-city TSP. As the annealing speed  $\beta$  is decreased, we see that the convergence rate to the optimum solution is decreased. This phenomenon is due to the following bifurcation mechanism.

As is illustrated in Fig. 3, for a bifurcation parameter region close to  $r=1$  which gives rise to the Hopfield-Tank “convergence” dynamics, there exist two local minimum solutions, the optimum solution  $Q^{(1)}$  and the second-optimum solution  $Q^{(2)}$ . With a decrease in the bifurcation parameter  $r$ , the two solutions  $\{Q^{(1)}, Q^{(2)}\}$  bifurcate into chaotic attractors through a period-doubling route. The two attractors are initially localized in the state space and eventually merge into a single attractor via crises. First, the optimum solution  $Q^{(1)}$  touches the separatrix of  $Q^{(1)}$  and  $Q^{(2)}$  and loses its stability via a boundary crisis. Then, the second-optimum solution  $Q^{(2)}$  merges with the ruin of the optimum solution  $Q^{(1)}$  via an interior crisis. The merger of  $Q^{(1)}$  and  $Q^{(2)}$  can be schematically illustrated in a binary tree structure of Fig. 3. Notice that, in this merging process, whereas the global minimum solution  $Q^{(1)}$  has an unstable parameter region, the second minimum solution  $Q^{(2)}$  is con-

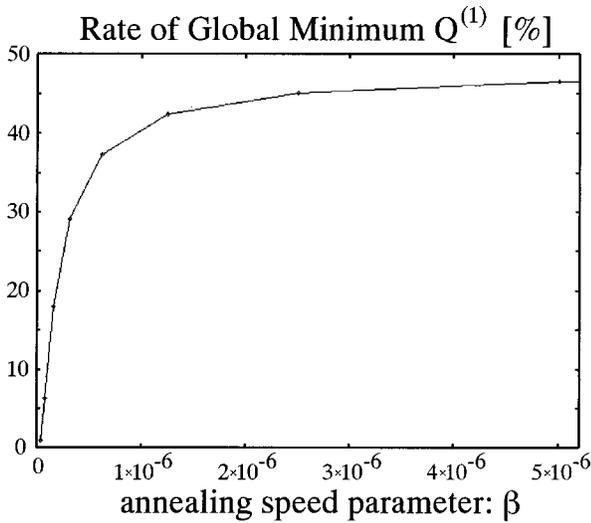


FIG. 2. The global optimization rate of the CSA algorithm for a five-city TSP is drawn with decreasing the annealing speed parameter  $\beta$ . The city locations are given in two-dimensional coordinates as (0.1768, 0.2233), (0.9348, 0.6305), (0.1561, 0.5661), (0.5793, 0.0830), (0.0358, 0.6269) and the system parameters are set to  $(A, B, \omega, \alpha, \epsilon, r_g, r_s) = (1.5, 1.0, 0.70, 0.07, 0.018, 0.95, 0.75)$ .

tinually stable until the final merger. An infinitely slow annealing always provides the second minimum solution  $Q^{(2)}$ , because in the unstable parameter region of  $Q^{(1)}$  every slow annealing is trapped in the second minimum solution  $Q^{(2)}$ .

Let us consider a general case of annealing multiple-attractor systems. As in the previous discussion, the hierar-

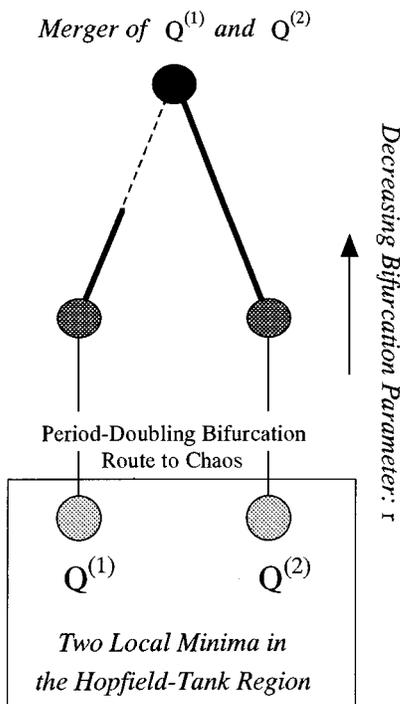


FIG. 3. Schematic illustration of the merging process of two solutions,  $Q^{(1)}$  and  $Q^{(2)}$ . The continual solid line indicates the branch of the stable solution  $Q^{(2)}$ , the broken line indicates the branch of the unstable solution  $Q^{(1)}$  which lost its stability via a boundary crisis, and the node of the two branches indicates the merger of  $Q^{(1)}$  and  $Q^{(2)}$ .

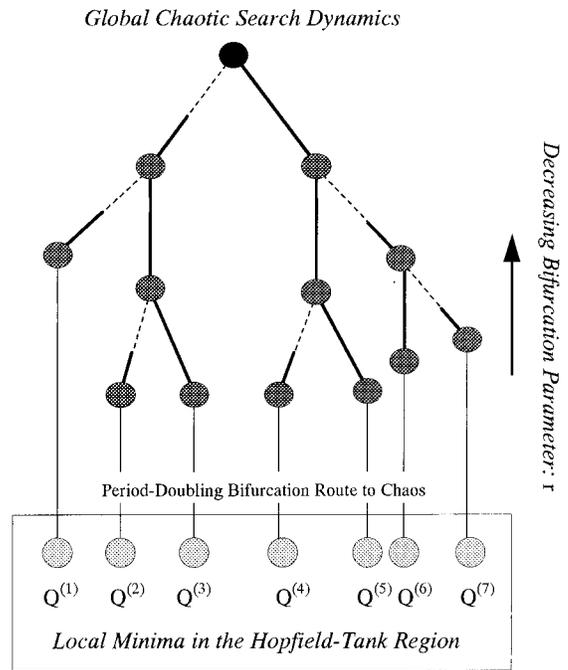


FIG. 4. Schematic illustration of the hierarchical merging structure of multiple TSP solutions  $\{Q^{(1)}, Q^{(2)}, \dots\}$ . The solid line indicates the branch of a stable solution, the broken line indicates the branch of an unstable solution which lost its stability via a boundary crisis, and the node of the two branches indicates the merger of the two solutions.

chical merging process of multiple chaotic attractors  $\{Q^{(1)}, Q^{(2)}, \dots\}$  can be schematically represented in a binary tree structure of Fig. 4. Notice that, in the merging process, there is only a single chaotic attractor which is continually stable until the final merger. An infinitely slow annealing which traces only a stable solution provides such a continual attractor as the final solution. This implies that an *infinitely slow* annealing does not necessarily provide an optimum result, since the optimum solution does not always survive until the final merger. With an analogy from the stochastic simulated annealing [11], it has been conjectured that the

TABLE I. Results of the adaptive CSA algorithm and the slow CSA algorithm against 50 instances of random 20-city TSP and 20 instances of random 40-city TSP. For each instance, 100 sets of random initial conditions are prepared. The parameters of the adaptive CSA algorithm are fixed to  $(A, B, \omega, \alpha, \epsilon, r_g, r_s, \beta, w) = (1.0, 1.0, 0.75, 0.1, 0.018, 0.95, 0.7, 0.04, 100)$ , while the gap parameter is set to  $\epsilon = 0.04$  for 20-city TSP and  $\epsilon = 0.10$  for 40-city TSP. The annealing speed of the slow CSA algorithm is set to  $\beta = 0.001$  for 20-city TSP and  $\beta = 0.0005$  for 40-city TSP where other parameters are set the same as the adaptive algorithm.

	Adaptive CSA	Slow CSA
Average tour length for random 20-city TSP	0.892	1.00
Averaged computation steps for random 20-city TSP	1200.0	5 520.0
Average tour length for random 40-city TSP	0.855	1.00
Averaged computation steps for random 40-city TSP	9624.1	11 041.0

CSA algorithm also provides an optimum result by infinitely slow annealing. The present result provides a counterexample for this conjecture.

In order to improve the conventional CSA algorithm, we introduce the following adaptive chaotic simulated annealing algorithm:

$$p_{ik}(n+1) = r(n)p_{ik}(n) + [1 - r(n)] \times \sigma \left( \sum_{j=1}^N \sum_{l=1}^N T_{ik,jl} p_{jl}(n) + I_{ik} \right), \quad (7)$$

$$r(n+1) = (1 - \beta)[r(n) - r_g] + r_g \quad [\text{if } E(J(n)) < E_{\text{th}}], \quad (8)$$

$$r(n+1) = (1 - \beta)[r(n) - r_s] + r_s \quad [\text{if } E(J(n)) \geq E_{\text{th}}], \quad (9)$$

where the cost function  $E(\cdot)$  is defined with the temporal network firing state  $J(n)$  as

$$E(J(n)) = \frac{A}{2} \sum_{i=1}^N \left( \sum_{k=1}^N J_{ik}(n) - 1 \right)^2 + \frac{A}{2} \sum_{k=1}^N \left( \sum_{i=1}^N J_{ik}(n) - 1 \right)^2 + \frac{B}{2} \sum_{i=1}^N \sum_{j=1}^N \sum_{k=1}^N d_{ij} J_{ik}(n) \{J_{jk+1}(n) + J_{jk-1}(n)\}. \quad (10)$$

The adaptive CSA algorithm utilizes ‘‘chaotic search’’ dynamics to seek for a TSP solution which has a lower cost than the threshold value  $E_{\text{th}}$ . When such a solution is found, the algorithm promptly tunes the bifurcation parameter  $r$  to the Hopfield-Tank ‘‘convergence’’ region and cools down the network dynamics to the equilibrium state.

Table I shows the result of the adaptive CSA algorithm applied to random 20- and 40-city TSPs. The threshold value is set as  $E_{\text{th}} = (1 + \varepsilon)C_{\text{HK}}$  ( $\varepsilon$ : gap parameter) using the Held-Karp lower bound  $C_{\text{HK}}$  [17]. The adaptive CSA algorithm provides much better solutions with fewer numbers of computation steps compared with the conventional slow CSA algorithm.

In conclusion, we have analyzed the CSA algorithm in the light of the global bifurcation structure of the chaotic neural networks and reported the limitation of the conventional slow CSA algorithm. As an improved algorithm, an adaptive chaotic simulated annealing algorithm which finds much improved solutions by fast annealing is introduced.

Our present discussions have been based on the application results to relatively small-scale TSPs. By studying the dependence of the convergence time of the adaptive algorithm to the size of the problem and by analyzing the scaling property of the algorithm, applicability of the present method to large-scale problems could be discussed in a further study. Validity of the present results to other combinatorial optimization problems should also be tested. By comparative studies with various other approximate algorithms such as the 2-opt algorithm, the Tabu search algorithm, the genetic algorithm, and many others [12,18], disadvantages as well as advantages of the present algorithm would be clarified.

We also note that, in the present study, chaotic annealing algorithms have been based on the controlling algorithm of a single bifurcation parameter. It is a challenging but worthwhile investigation to develop an algorithm which directly controls the asymptotic measure of the chaotic neural networks. Controlling the asymptotic measure of the chaotic dynamics to eventually converge to the optimum state may provide us with more natural annealing algorithm. Towards this annealing, we have preliminary results based on the learning algorithm of chaotic neural networks [19]. The learning algorithm approach seems to work quite effectively for TSP and the detailed result will be reported elsewhere.

- 
- [1] K. Aihara, T. Takabe, and M. Toyoda, *Phys. Lett. A* **144**, 333 (1990).  
 [2] H. Nozawa, *Chaos* **2**, 377 (1992).  
 [3] L. Chen and K. Aihara, *Neural Networks* **8**, 915 (1995).  
 [4] L. Chen and K. Aihara, *Physica D* **104**, 286 (1997).  
 [5] M. Hasegawa, T. Ikeguchi, and K. Aihara, *Phys. Rev. Lett.* **79**, 2344 (1997).  
 [6] I. Tokuda, T. Nagashima, and K. Aihara, *Neural Networks* **10**, 1673 (1997).  
 [7] I. Tokuda, K. Aihara, and T. Nagashima, Institute of Electronics, Information, and Communication Engineers Technical Report No. NLP96-60, 1996.  
 [8] Special issue on chaotic dynamics approach to optimization problems can be found in, e.g., Institute of Electrical Engineers of Japan Technical Report No. IP-97-1, 1997.  
 [9] J. J. Hopfield and D. W. Tank, *Biol. Cybern.* **52**, 141 (1985).  
 [10] S. Kirkpatrick *et al.*, *Science* **220**, 671 (1983).  
 [11] S. Geman and D. Geman, *IEEE Trans. Pattern. Anal. Mach. Intell.* **6**, 721 (1984).  
 [12] E. L. Lawler *et al.*, *The Traveling Salesman Problem* (Wiley, New York, 1985).  
 [13] C. Grebogi, E. Ott, and J. A. York, *Phys. Rev. Lett.* **48**, 1507 (1982).  
 [14] I. Tsuda, *World Futures* **32**, 16 (1991).  
 [15] K. Kaneko, *Physica D* **41**, 137 (1990).  
 [16] K. Ikeda, K. Otsuka, and K. Matsumoto, *Prog. Theor. Phys. Suppl.* **99**, 295 (1989).  
 [17] M. Held and R. M. Karp, *Oper. Res.* **18**, 1138 (1970).  
 [18] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning* (Addison-Wesley, Reading, 1989).  
 [19] I. Tokuda *et al.*, Institute of Electronics, Information, and Communication Engineers Technical Report No. NLP97-83, 1997.