Solving Combinatorial Optimization Problems Using Stochastic Chaotic Simulated Annealing

Lipo Wang, Sa Li, and Fuyu Tian*

School of Electrical and Electronic Engineering
Nanyang Technological University
Block S2, Nanyang Avenue, Singapore 639798

* To whom all correspondence should be addressed

Email: elpwang@ntu.edu.sg
URL: http://www.ntu.edu.sg/home/elpwang

Abstract – Chen and Aihara have showed recently that their chaotic simulated annealing (CSA) has better search ability for solving combinatorial optimization problems compared to both the Hopfield-Tank approach and stochastic simulated annealing (SSA). However, CSA is not guaranteed to find a globally optimal solution no matter how slowly annealing is carried out. In contrast, SSA is guaranteed to settle down to a global minimum with probability 1 if the temperature is reduced sufficiently slowly. In this paper, we attempt to combine the best of both heuristics by proposing a new approach to simulated annealing using a noisy chaotic neural network, i.e., stochastic chaotic simulated annealing (SCSA). We demonstrate this approach with the traveling salesman problem.

1. Introduction

Recently there have been extensive research interests and efforts in theory and applications of chaotic neural networks (e.g., [1]-[16]). In particular, Freeman and co-workers [1] have demonstrated strong evidence, through both biological experiments and theoretical investigations, that chaos play an important role in information processing in real and artificial neural systems.

Aihara, Takabe, and Toyoda [3] proposed a chaotic neural network based on a modified Nagumo–Sato neuron model, in order to explain complex dynamics observed in a biological neural system. Nozawa [4] showed that the Euler approximation of the continuous-time Hopfield neural network [17] (EP-HNN) with a negative neuronal self-coupling has chaotic dynamics and that this model is equivalent to a special case of Aihara-Takabe-Toyoda chaotic neural network [3] after a variable transformation. Nozawa further showed [4][6] that the EP-HNN has much higher searching ability for solving the traveling salesman problem (TSP), in comparison with the Hopfield neural network [17][18][19], the Boltzmann machine, and the Gaussian machine.

Chen and Aihara [7][8] proposed chaotic simulated annealing (CSA) by starting with a sufficiently large negative self-coupling in the Aihara-Takabe-Toyoda network and gradually decreasing the self-coupling so that the network eventually stabilizes, thereby obtaining a transiently chaotic neural network. Their computer simulations showed that CSA obtains good solutions for TSP much more easily compared to the Hopfield-Tank approach [17]-[19] and stochastic simulated annealing (SSA) [20]. Chen and Aihara [15] provided the following theoretical explanation for the global searching ability of the chaotic neural network: its attracting set contains all global and local minima of the optimization problem under certain conditions, and since the chaotic attracting set has a fractal structure and covers only a very small fraction of the entire state space, CSA is more efficient in finding good solutions for optimization problems compared to other global search algorithms such as SSA.

SSA is known to relax to a global minimum if the annealing takes place sufficiently slowly, i.e., no faster than logarithmatically. In practical terms, this means that SSA is capable of producing good (optimal or near-optimal) solutions for many applications, if the annealing parameter (temperature) is reduced exponentially but with a reasonably small exponent. However, unlike SSA, CSA has completely *deterministic* dynamics and is not guaranteed to settle down at a global minimum no matter how slowly the annealing parameter (the self-coupling) is reduced [22]. Practically speaking, this implies that CSA sometimes may not be able to provide a good solution at the conclusion of annealing, even when annealing is carried out very slowly.

In this paper, we combine the best features of both SSA and CSA, i.e., stochastic wandering and efficient chaotic searching, by adding a decreasing noise in the transiently chaotic neural network of Chen and Aihara [7][8][15]. We therefore obtain a novel method for solving optimization problems: stochastic chaotic simulated annealing (SCSA). We then use the proposed SCSA to solve the TSP. Our results show marked improvement over CSA.

2. Stochastic Chaotic Simulated Annealing

We propose stochastic chaotic simulated annealing (SCSA) as follows:

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

$$y_{ij}(t+1) = ky_{ij}(t) + \alpha\Big(\sum_{k,l=1,k,l\neq i,j}^{n} w_{ijkl}x_{kl}(t) + I_{ij}\Big) - z(t)(x_{ij}(t) - I_0) + n(t) \tag{2}$$

$$z(t+1) = (1-\beta)z(t) \quad (i,j,k,l = 1,\dots,n) \tag{3}$$

$$A[\,n(t+1)\,] = (1-\beta)\,A[\,n(t)\,] \tag{4}$$

where

x_{ij}: output of neuron ij;

y_{ij}: internal state of neuron ij;

I_{ij}: input bias of neuron ij;

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 (ε >0);

n(t): random noise injected into the neurons, i.e., in [-*A*, *A*] with a uniform distribution, where *A[n]* is the noise amplitude;

w_{ijkl}: connection weight from neuron kl to neuron ij.

The weights satisfy the following:

$$w_{ijkl} = w_{klij};\, w_{ijij} = 0; \qquad \sum_{k,l=1,k,l\neq i,j}^{n} w_{ijkl}x_{kl} + I_{ij} = -\partial E/\partial x_{ij} \tag{5}$$

In the absence of noise, i.e., *n(t) = 0*, for all *t*, SCSA reduces to CSA [7][8][15].

3. Solving the Traveling Salesman Problem Using Stochastic Chaotic Simulated Annealing

A classical combinatorial optimization problem is the travelling salesman problem (TSP). It is to seek the shortest route through *n* cities, visiting each city once and only once, and returning to the starting point. Since Hopfield and Tank [18] applied their neural networks to the TSP, the TSP is often used as a test problem in neurocomputing.

Table 1. Results of CSA and SCSA on Hopfield-Tank's 10-city TSP for 5000 runs with different random initial conditions of the network.

Algorithm	CSA	SCSA
Number of runs reaching global minima (%)	4969 (99.4%)	4972 (99.4%)
Number of runs reaching others solutions (%)	31 (0.6%)	28 (0.6%)
Average number of iterations	119	124

Following Hopfield and Tank [18], we map 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 to be minimized consists of two parts:

$$E = \frac{W_1}{2}\left\{\sum_{i=1}^{n}\left(\sum_{j=1}^{n} x_{ij} - 1\right)^2 + \sum_{j=1}^{n}\left(\sum_{i=1}^{n} x_{ij} - 1\right)^2\right\} + \frac{W_2}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n}\left(x_{kj+1} + x_{kj-1}\right)x_{ij} d_{ik} \quad (6)$$

where $x_{i0} = x_{in}$ and $x_{in+1} = x_{i1}$. d_{ij} is the distance between city i and city j. The first two terms in eq.6 (inside {}) represent the constraints, i.e., one and only one x_{ij} is 1 for each *j*, one and only one x_{ij} is 1 for each *i* (each city is visited once and only once). The last term in eq.6 shows the total length of the tour. Coefficients W_1 and W_2 reflect the relative strength of the constraint and the tour length terms. Hence a global minimum of E represents a *shortest valid* tour.

Here we solve Hopfield-Tank's 10-city TSP with our model. To compare the performance with the CSA, we use a set of $k, \alpha, \beta, \varepsilon, W_1, W_2$ that are the same as Chen and Aihara's [15][16]:

$$k = 0.90; \varepsilon = 0.004; I_0 = 0.65;$$
$$z(0) = 0.08; \alpha = 0.015;$$
$$\beta = 0.01; W_1 = W_2 = 1 \quad (7)$$

In SCSA, *A*[*n*(0)] = 0.002. The results are summarized in Table 1with 5000 different initial conditions of y_{ij} generated randomly in the region [-1,1].

As shown in Table 1, the performance of CSA and SCSA is about the same for the 10-city TSP.

We also use a 21-city TSP [21] to compare the performance of our SCSA with CSA. The optimal (shortest) tour length is known to be 2707 [21].

The distance matrix d_{ij} is given in following page (Only d_{ij} with $i \geq j$ are shown. We assume $d_{ij} = d_{ji}$, i.e., symmetric TSP).

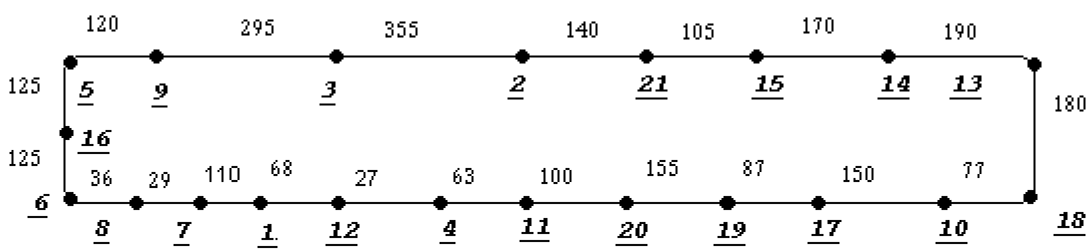
The network parameters are set as follows:

$$k = 0.90; \varepsilon = 0.004; I_0 = 0.5;$$
$$z(0) = 0.10; \alpha = 0.015;$$
$$\beta = 10^{-4}; W_1 = W_2 = 1 \quad (8)$$

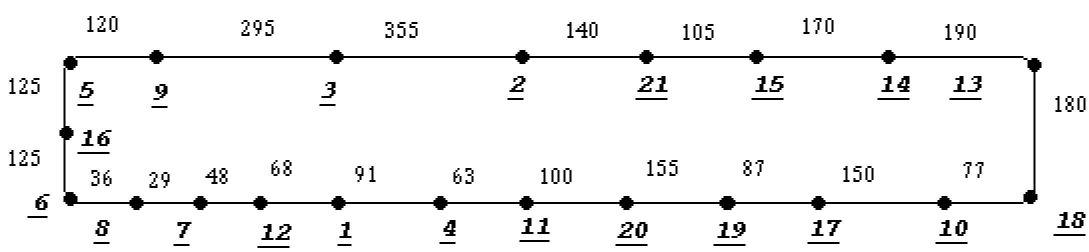
Compared to the 10-city TSP, we use a smaller β to allow for longer searching. For SCSA, the

0																				
510	0																			
635	355	0																		
91	415	605	0																	
385	585	390	350	0																
155	475	495	120	240	0															
110	480	570	78	320	96	0														
130	500	540	97	285	36	29	0													
490	605	295	460	120	350	425	390	0												
370	320	700	280	590	365	350	370	625	0											
155	380	640	63	430	200	160	175	535	240	0										
68	440	575	27	320	91	48	67	430	300	90	0									
610	360	705	520	835	605	590	610	865	250	480	545	0								
655	235	585	555	750	615	625	645	775	285	515	585	190	0							
480	81	435	380	575	440	455	465	600	245	345	415	295	170	0						
265	480	420	235	125	125	200	165	230	475	310	205	715	650	475	0					
255	440	755	235	650	370	320	350	680	150	175	265	400	435	385	485	0				
450	270	625	345	660	430	420	440	690	77	310	380	180	215	190	545	225	0			
170	445	750	160	495	265	220	240	600	235	125	170	485	525	405	375	87	315	0		
240	290	590	140	480	255	205	220	515	150	100	170	390	425	255	395	205	220	155	0	
380	140	495	280	480	340	350	370	505	185	240	310	345	280	105	380	280	165	305	150	0

(a)



(b)



(c)

Figure 1. (a) The distances between the cities. (b) The optimal tour in the 21-city TSP with tour length 2707. (c) The near-optimal tour in the 21-city TSP with tour length 2709. The numbers underlined represent the cities, whereas the numbers not underlined represent the distances between the cities. The only difference between the optimal tour and the near-optimal tour is that cities 1 and 12 are swapped.

Table 2. Results of CSA and SCSA on the 21-city TSP for 400 runs with randomly generated initial network conditions.

Model	CSA	SCSA
Number of runs reaching the global optimum (%)	0 (0%)	186 (46.5%)
Number of runs reaching the near-optimum (%)	400 (100%)	214 (53.5%)
Average number of iterations	14500	14500

initial noise amplitude is the same as in the previous case, i.e., $A[n(0)] = 0.002$.

The results are summarized in Table 2 with 400 different initial conditions of y_{ij} generated randomly in the region [-1,1]. Table 2 shows that SCSA can find the optimal route (tour length=2707, shown in Fig.1(b)) in 46.5% of the runs, and the rest 53.5% runs converge to a near-optimal solution (tour length=2709, shown in Fig.1(c)). On the other hand, CSA can not find the optimal route, but always converges to the near-optimal solution.

We note that Hopfield and Tank’s prescription of mapping the TSP onto a neural network as described at the beginning of this section (eq.6) is not the most effective way for solving the TSP using neural networks. Because of the need for n^2 neurons, the size of the TSP that can be handled by this prescription is limited. Other prescriptions specially tailored for the TSP can increase the size of the TSP significantly (e.g., [22]). In this 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 improved searching ability of SCSA over the same objective functions. In other words, neither the 10-city TSP nor the 21-city TSP studied above may be considered difficult; however, finding the global optima for the objective functions given by eq.6 with parameters specified above is indeed non-trivial and can therefore be used as benchmarking optimization problems to compare various optimization algorithms.

5. Conclusions

In this paper, we proposed stochastic chaotic simulated annealing (SCSA) by adding noise to Chen-Aihara's transiently chaotic neural network. Application of this noisy chaotic neural network to the TSP showed marked improvement over chaotic simulated annealing (CSA). In contrast to the conventional stochastic simulated annealing (SSA), SCSA restricts the random search to a sub-space of chaotic attracting set which is much smaller than the entire state space searched by SSA. In contrast of CSA, SCSA continues to search after the disappearance of chaos. Future work will include applications of SCSA to other practical optimization problems.

References

[1] C. A. Skarda and W. J. Freeman, "How brains make chaos in order to make sense of the world," *Brain and behavioral Science*, vol. 10, pp. 161-195, 1987.

[2] 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.

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

[4] 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.

[5] L. Wang and D.L. Alkon, (eds.), *Artificial neural networks: Oscillations, chaos, and sequence processing*. (IEEE Computer

Society Press), Los Alamitos, California, 1993.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] R. Kozma, N.K. Kasabov, J.A. Swope, M.J.A. Williams, "Combining neuro-fuzzy and chaos methods for intelligent time series analysis-case study of heart rate variability", *1997 IEEE International Conference on Systems, Man, and Cybernetics, Computational Cybernetics and Simulation*, vol.4, Page(s): 3025 -3029, 1997.

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

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] J. J. Hopfield, "Neurons with graded response have collective computational properties like those of two-state neurons," *Proc. Natl. Acad. Sci. USA*, vol. 81, pp. 3088-3092, May, 1984.

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

[19] 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.

[20] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimisation by simulated annealing. *Science*, vol. 220, pp. 671-680, 1983.

[21] G. Reinelt, "TSPLIB - A traveling salesman problem library," *ORSA J. Computing*, vol. 3, no. 4, pp.376-384, 1991. http://www.iwr.uniheidelberg.de/iwr/comopt/software/TSPLIB95/tsp/gr21.tsp.gz

[22] I. Tokuda, K. Aihara, and T. Nagashima, "Adaptive annealing for chaotic optimization", *Phys. Rev. E*, vol.58, pp. 5157-5160, 1998.

[23] 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.