A noisy chaotic neural network for solving combinatorial optimization problems : stochastic chaotic simulated annealing

Wang, Lipo.; Fu, Xiuju; Li, Sa; Tian, Fuyu

2004

Wang, L., Li, S., Tian, F., & Fu, X. (2004). A noisy chaotic neural network for solving combinatorial optimization problems: stochastic chaotic simulated annealing. IEEE Transactions on Systems, Man and Cybernetics-Part B: Cybernetics, 34(5), 2119-2125.

https://hdl.handle.net/10356/93979

https://doi.org/10.1109/TSMCB.2004.829778

A Noisy Chaotic Neural Network for Solving Combinatorial Optimization Problems: Stochastic Chaotic Simulated Annealing

Lipo Wang, Sa Li, Fuyu Tian, and Xiuju Fu

***Abstract*—Recently Chen and Aihara have demonstrated both experimentally and mathematically 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 may not find a globally optimal solution no matter how slowly annealing is carried out, because the chaotic dynamics are completely deterministic. In contrast, SSA tends to settle down to a global optimum if the temperature is reduced sufficiently slowly. Here we combine the best features of both SSA and CSA, thereby proposing a new approach for solving optimization problems, i.e., stochastic chaotic simulated annealing, by using a noisy chaotic neural network. We show the effectiveness of this new approach with two difficult combinatorial optimization problems, i.e., a traveling salesman problem and a channel assignment problem for cellular mobile communications.**

***Index Terms*—Channel assignment, chaos, combinatorial optimization, neural network.**

I. Introduction

Chaotic neural networks have a richer spectrum of dynamic behaviors, such as stable fixed points, periodic oscillations, and chaos, in comparison with *static* neural network models. Recently, there have been extensive research interests and efforts in theory and applications of chaotic neural networks (for example, see [1]–[22]).

A chaotic neural network based on a modified Nagumo–Sato neuron model was proposed by Aihara *et al.* [4] in order to explain complex dynamics observed in a biological neural system. Nozawa [5] showed that the Euler approximation of the continuous-time Hopfield neural network [23] (EA-HNN) with a negative neuronal self-coupling exhibits chaotic dynamics and that this model is equivalent to a special case of a Aihara–Takabe–Toyoda chaotic neural network [4] after a variable transformation. Nozawa further showed [5], [7] that the EA-HNN has a much better searching ability in solving the traveling salesman problem (TSP), in comparison with the original Hopfield neural network [23]–[25], the Boltzmann machine, and the Gaussian machine.

Chen and Aihara [8], [9] proposed chaotic simulated annealing (CSA) by starting with a sufficiently large negative self-coupling in the Aihara–Takabe–Toyoda network when the dynamics is chaotic, 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 leads to good solutions for the TSP much more easily compared to the Hopfield-Tank approach [23], [24] and stochastic simulated annealing (SSA) [26]. Chen and Aihara [18] offered the following theoretical explanation for the global searching ability of the chaotic neural network: its attracting set contains all global and local optima 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 searching for good solutions for optimization problems compared to other global search algorithms such as SSA.

It is well-known that SSA tends to find a global optimum if the annealing process is carried out sufficiently slowly [27]. Practically speaking, this implies that SSA is able to find high-quality solutions (global optima or near-global-optima), if the annealing parameter (temperature) is reduced exponentially but with a sufficiently small exponent. However, for many applications, this may mean prohibitively long relaxation time in order to find solutions of acceptable quality, and conversely, reasonably long periods of time may still result in poor solutions. In this sense, SSA searches through the solution space in a much less efficient way compared to CSA, i.e., the stochastic search in SSA covers the entire solution space, rather than a fraction of the solution space covered by the search in CSA.

Although CSA searches in an efficient manner, CSA has completely *deterministic* dynamics and is not guaranteed to settle down at a global optimum no matter how slowly the annealing parameter (the neuronal self-coupling) is reduced [14]. In practical terms, this means that CSA may sometimes be unable to provide a good solution at the end of annealing for some initial conditions of the network, no matter how slowly annealing takes place.

We attempt in this paper to combine the best of both SSA and CSA, i.e., stochastic wandering and efficient chaotic searching, by adding a decaying stochastic noise in the transiently chaotic neural network of Chen and Aihara [8], [9], [18]. We thus obtain a novel method for solving a general class of combinatorial optimization problems: stochastic chaotic simulated annealing (SCSA). Next, to demonstrate the effectiveness of the proposed SCSA, we apply the proposed SCSA to solving two difficult combinatorial optimization problems, i.e., a TSP and a channel assignment problem (CAP) for cellular mobile communications. Our simulation results show that SCSA leads to remarkable improvements over CSA.

This paper is organized as follows. Section II formulates SCSA. Sections III and IV present applications of SCSA to a TSP and a CAP, respectively, including problem statements, mappings of the problems onto chaotic neural networks, and results of computer simulations. Section V concludes the paper.

II. SCSA

By adding decaying stochastic noise into Chen and Aihara's transiently chaotic neural network [8], [9], [18], we propose SCSA as follows:

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

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

$$z(t+1) = (1-\beta_1)z(t), \quad i,j,k,l = 1,\ldots,n \tag{3}$$

$$A\left[n(t+1)\right] = (1-\beta_2)A\left[n(t)\right] \tag{4}$$

where the variables are

- x_{ij} — output of neuron ij;
- y_{ij} — internal state of neuron ij;
- I_{ij} — input bias of neuron ij;
- k — damping factor of the nerve membrane ($0 \leq k \leq 1$);
- α — positive scaling parameter for the inputs;
- $z(t)$ — self-feedback neuronal connection weight or refractory strength ($z(t) \geq 0$);

L. Wang is with School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 639798 and also with Institute of Information Engineering, Xiangtan University, Xiangtan, Hunan, China (e-mail: elpwang@ntu.edu.sg).

S. Li, F. Tian and X. Fu are with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 639798.

β_1, β_2 damping factors for the time-dependent neuronal self coupling and the added random noise, respectively ($0 \le \beta_1 \le 1, 0 \le \beta_2 \le 1$);

I_0 positive parameter;

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

$n(t)$ random noise injected into the neurons, with its actual value being in the range $[-A, A]$ and with a uniform distribution, where $A[n]$ is the noise amplitude;

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

The connection weights can be obtained from

$$\sum_{k,l=1,k,l\neq i,j}^{n} w_{ijkl}x_{kl} + I_{ij} = -\frac{\partial E}{\partial x_{ij}} \tag{5}$$

where E is the energy function of the network or the cost function to be minimized in a give combinatorial optimization problem. In addition, the connection weights satisfy $w_{ijkl} = w_{klij}$ and $w_{ijij} = 0$. In the absence of noise, i.e., $n(t) = 0$, for all t, SCSA as proposed in (1)–(5) reduces to CSA of Chen and Aihara [8], [9], [18].

Furthermore, in the absence of noise and damping of the self-neuronal coupling, i.e., $n(t) = 0$, for all t, and $\beta_1 = 0$, (1)–(5) become the Aihara–Takabe–Toyoda chaotic neural network [4], which is known to have a variety of different dynamical behaviors, including stable fixed points, periodic oscillations, and chaos, depending on the values of the network parameters.

III. Solving the TSP Using SCSA

The TSP is a classical combinatorial optimization problem. The goal of the TSP is to find the shortest route through n cities, visiting each city once and only once, and returning to the starting point. Since Hopfield and Tank [18] first applied their neural networks to solving the TSP, the TSP is often used as a benchmarking problem for testing neural network approaches to solving combinatorial optimization problems.

Hopfield and Tank [24] 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 to be minimized consists of two parts, as follows:

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

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 (6) (inside $\{\}$) represent the constraints, i.e., one and only one x_{ij} is 1 for each j, and one and only one x_{ij} is 1 for each i (each city is visited once and only once). The last term in (6) (without the coefficient W_2) represents the total length of the tour. Coefficients W_1 and W_2 reflect the relative strength of the constraint and the tour length terms. Thus, a global minimum of E represents a *shortest valid* tour.

We note that Hopfield and Tank's prescription of mapping the TSP onto a neural network as described above (6) is *not* the most effective way for solving the TSP using either neural networks or chaotic dynamics. 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., [22]). In this paper, we shall not attempt to adopt other mapping prescriptions in order to solve larger TSP's. Rather, the purpose of the present work is to demonstrate the improved solving ability of SCSA over CSA for a given objective function. In other words, neither the 10-city and the 21-city nor 52-city and 70-city TSP studied below may be considered difficult; however, *finding the global optima for the objective functions given by* (6) with parameters specified below (i.e., parameters drawn for the 10-city TSP, 21-city TSP, 52-city TSP and 70-city TSP) is indeed nontrivial and can therefore be used as benchmarking optimization problems to compare various optimization algorithms, such as CSA and SCSA. Hence, a more precise title for this Section would be "Searching for Global Minima of the Function Given by (6) Using SCSA".

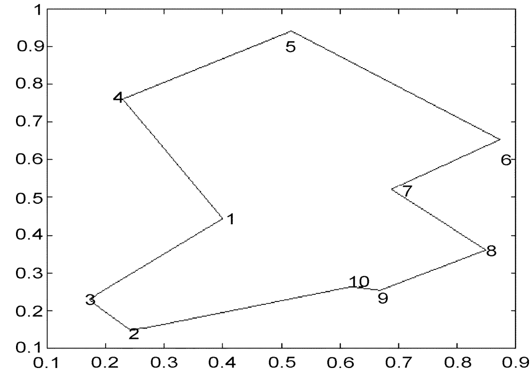

Fig. 1. The optimal tour of the Hopfield-Tank 10 city TSP.

From (2), (5), and (6), we derive the dynamics of the SCSA for the TSP as follows:

$$\begin{aligned} y_{ij}(t+1) = {} & ky_{ij}(t) - z(t)\,(x_{ij}(t) - I_0) \\ & + \alpha\left\{-W_1\left[\sum_{l\neq j}^{n} x_{il}(t) + \sum_{k\neq i}^{n} x_{kj}(t)\right]\right. \\ & \left. + W_1 - W_2\sum_{k\neq i}^{n}(x_{kj+1}(t) + x_{kj-1}(t))d_{ik}\right\} \\ & + n(t). \end{aligned} \tag{7}$$

We first minimize the function given by (6) derived from the Hopfield-Tank 10-city TSP (Fig. 1) [24] using our SCSA. To compare the performance with that of CSA, we use a set of $k, \alpha, \beta, \varepsilon, W_1, W_2$ that are the same as Chen and Aihara's [9]

$$\begin{aligned} & k = 0.90;\ \varepsilon = 0.004;\ I_0 = 0.65;\ z(0) = 0.10 \\ & \alpha = 0.015;\ \beta_1 = \beta_2 = 0.01;\ W_1 = W_2 = 1. \end{aligned} \tag{8}$$

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

As shown in Table I, the performances of CSA and SCSA are about the same for such a relatively small problem.

Next, we minimize the energy function given by (6) with parameters drawn from a 21-city TSP [28]. The optimal tour length is known to be 2707 [28], which is the global minimum of the energy function. The distance matrix d_{ij} is given as follows (only d_{ij} with $i \ge j$ are shown—we assume $d_{ij} = d_{ji}$, i.e., a symmetric TSP):

The system parameters for the noisy chaotic neural network, also shown at the bottom of the next page, are set as follows:

$$\begin{aligned} & k = 0.90;\ \varepsilon = 0.004;\ I_0 = 0.5;\ z(0) = 0.10 \\ & \alpha = 0.015;\ \beta_1 = 5 \times 10^{-5};\ W_1 = W_2 = 1. \end{aligned} \tag{9}$$

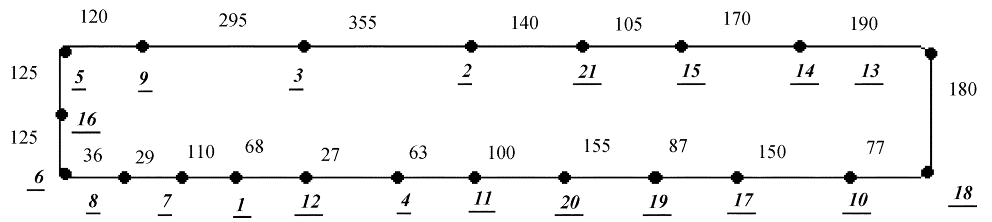

Fig. 2. The optimal tour in the 21-city TSP with tour length 2707. The numbers underlined represent the cities, whereas the numbers not underlined represent the distances between the cities.

TABLE I
COMPARISON OF CSA AND SCSA ON THE HOPFIELD-TANK 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 iterations	119	124

TABLE II
RESULTS OF CSA AND SCSA USING VARIOUS β_2 WITH 100 DIFFERENT INITIAL CONDITIONS

		Number of runs of reaching global minima (%)	Minimum Iteration Times
CSA ($\beta_2=1$)		100 (100%)	29105
SCSA	$\beta_2=10^{-4}$	100 (100%)	27578
	$\beta_2=5\times10^{-5}$	100 (100%)	27989
	$\beta_2=2\times10^{-5}$	100 (100%)	27654
	$\beta_2=10^{-5}$	100 (100%)	27638

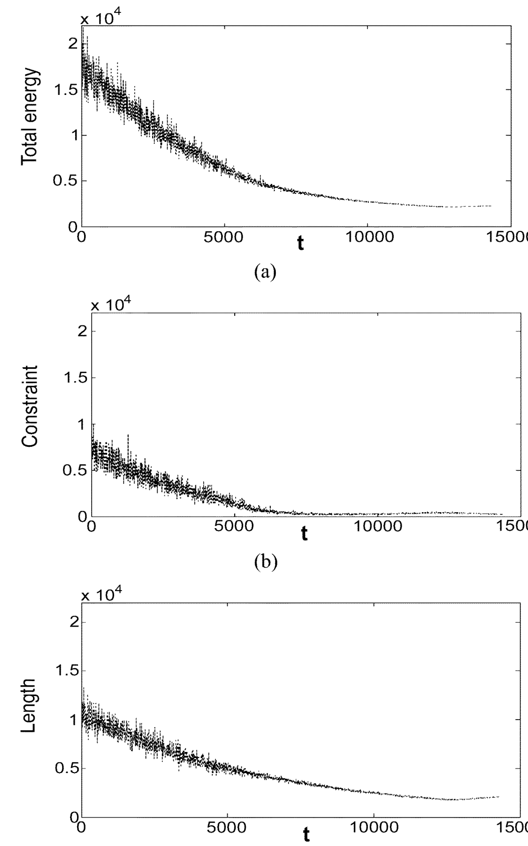

Fig. 3. The energy terms in (6) (TSP) as a function of time in SCSA: (a) the total energy; (b) the constraint energy term; (c) the tour-length energy term.

Compared to the 10-city TSP, we use smaller β_1 and β_2 to allow for longer searching. For SCSA, the initial noise amplitude is set to be the same as in the 10-city case, i.e., $A[n(0)] = 0.002$.

The results are summarized in Table II with 100 different randomly selected initial y_{ij} in the region $[-1, 1]$. Table II shows that both CSA and SCSA can find the optimal route with a tour length 2707 (Fig. 2). In contrast, CSA use longer computational time than SCSA. The minimum time of obtain the global optima by CSA is 86 s (the computer we use is x86 Family 6 Model 8 Stepping 6, AT/AT Compatible).

In our simulations, we used different damping factors β_1 and β_2 for chaos and noise, respectively, i.e., chaos and noise have different cooling schedules during annealing. Table II also shows the simulation results using various β_2, when β_1 was fixed at 5×10^{-5}. With the decrease of the annealing rate of noise, SCSA will find the global optima faster. When β_2 is set as 10^{-5}, it can use the minimum iteration times 27 638 to get the global optima (see Table II).

Coefficients W_1 and W_2, which reflect the relative strength of the constraint and the tour length energy terms (6), are selected such that these two terms are comparable in magnitude, so that neither of them dominates. For this purpose, as well as to show system dynamics during the search, we plot in Fig. 3 the total energy E in (6), the constraint energy

$$E_{\text{Constr}} = \frac{W_1}{2}\left\{\sum_{i=1}^{n}\left(\sum_{j=1}^{n} x_{ij} - 1\right)^2 + \sum_{j=1}^{n}\left(\sum_{i=1}^{n} x_{ij} - 1\right)^2\right\} \quad (10)$$

and the tour-length energy

$$E_{\text{Length}} = \frac{W_2}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n}(x_{kj+1} + x_{kj-1})x_{ij}d_{ik}. \quad (11)$$

Similarly, to help us select the other parameters in (9), we show the three input terms in (7) in Fig. 4 as follows:

the single neuron input term

$$ky_{jk}(t) - z_{jk}(t)(x_{jk}(t) - I_0) + n(t) \quad (12)$$

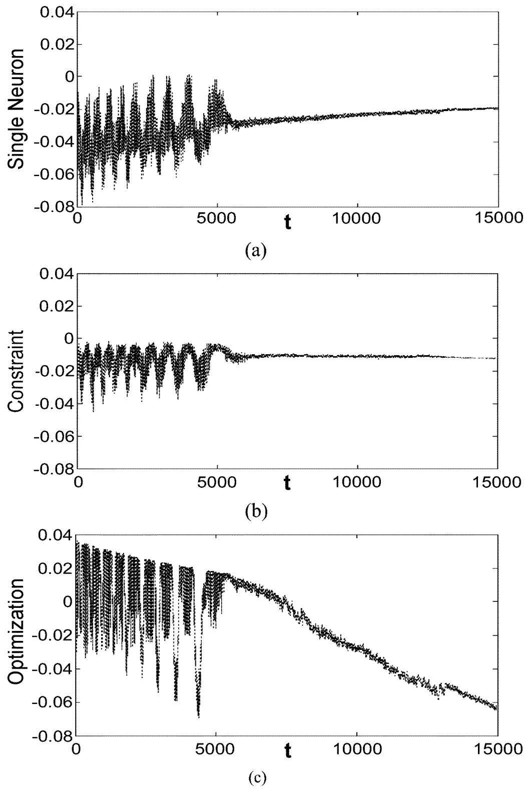

Fig. 4. The three input terms in (7) (TSP) as functions of time in SCSA: (a) the single neuron term; (b) the constraint term; (c) the optimization term.

the constraint input term

$$\alpha\left\{-W_1\left[\sum_{l\neq j}^{n} x_{il}(t)+\sum_{k\neq i}^{n} x_{kj}(t)\right]+W_1\right\} \tag{13}$$

the optimization term

$$\alpha\left\{1-W_2\sum_{k\neq i}^{n}(x_{kj+1}(t)+x_{kj-1}(t))d_{ik}\right\}. \tag{14}$$

Why can SCSA find the global optimum 2707 faster than CSA? One possible reason may be the stochastic nature of SCSA. For comparisons, in Figs. 5 and 6 we plot the corresponding energy and input terms for CSA.

To test our algorithm further, we handle a relatively larger size TSP problem, Berlin52, [28] which includes 52 cities. The best-known tour length listed in TSPLIB is 7542.

We run the simulations with 200 different randomly selected initial y_{ij} in the region $[-1, 1]$. Table III shows the simulation results using various strengths of the tour length energy terms W_2, when the coefficient W_1 is fixed as 1. The system parameters are set as follows:

$$\begin{aligned} &k = 0.90;\ \varepsilon = 0.004;\ I_0 = 0.5;\ z(0) = 0.10 \\ &\alpha = 0.02;\ \beta_1 = \beta_2 = 3\times 10^{-6};\ A\,[n(0)] = 0.02. \end{aligned} \tag{15}$$

From Table III, SCSA shows better performance for the large TSP Berlin52. When W_2 is set as 1.6, SCSA can find the result 7525 better than 7542. Here, we provide one optimal tour of our simulation, which is obtained in a PC computational environment (x86 Family 6 Model 8 Stepping 6, AT/AT Compatible)

$$\begin{aligned} [&24-48-38-37-40-39-36-35-34-44-46- \\ &16-29-50-20-23-30-2-7-42-21-17- \\ &3-18-31-22-1-49-32-45-19-41-8- \\ &9-10-43-33-51-11-52-14-13-47-26- \\ &27-28-12-25-4-6-15-5]. \end{aligned}$$

For comparison, we extract the case of $W_2 = 1.6$, and fix $\beta_1 = 3\times 10^{-6}$ and compare the different performance of CSA and SCSA by using various β_2 in Table IV. In Table IV, the minimum tour length that CSA can find is 7555, which is far greater than the minimum tour length 7525 obtained by SCSA. For the number of valid in the 100 runs, the number obtained by CSA is apparently much lower than the number of valid tours found by SCSA. According to the simulations, the SCSA shows noticeable improvement over CSA.

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

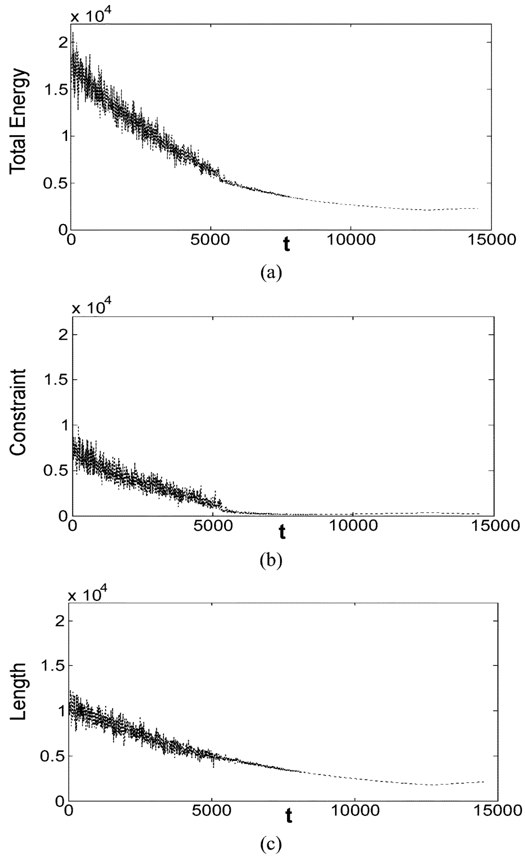

Fig. 5. Same as Fig. 3, for CSA.

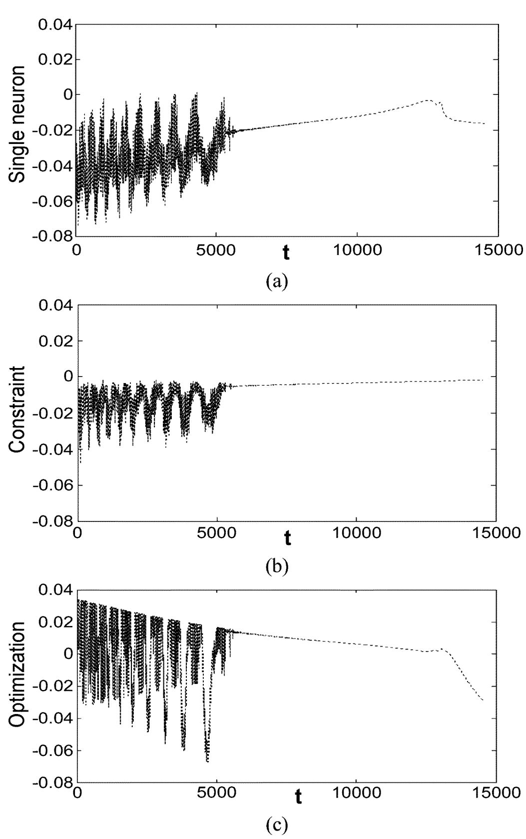

Fig. 6. Same as Fig. 4, for CSA.

To make our work more convincing, a 70-city TSP (ST70) [28] is further used. The best-known tour length listed in TSPLIB is 675. We did the simulations by SCSA with 20 different randomly selected initial y_{ij} in the region $[-1, 1]$. The system parameters are set as same as in (15).

TABLE III
RESULTS FOR BERLIN52 TSP USING SCSA WITH VARIOUS W_2 AND 200 DIFFERENT INITIAL CONDITIONS

W_2	Minimum Tour Length	Average Tour Length (not include the invalid)	The number of valid (%)
1.0	7599	8570.4	175 (87.5%)
1.2	7544	7965.1	70 (35%)
1.4	7559	8002.3	105 (52.5%)
1.6	7525	7885.6	68 (34%)
1.7	7530	7914.5	39 (19.5%)
2.0	7549	7855.5	77 (38.5%)
2.5	7636	7997.1	136 (68%)

TABLE IV
RESULTS OF CSA AND SCSA IN BERLIN52 USING VARIOUS β_2 WITH 200 DIFFERENT INITIAL CONDITIONS IN $W_1 = 1, W_2 = 1.6$

		Minimum Tour Length	Average Tour Length (not include the invalid)	The number of valid (%)
CSA ($\beta_2=1$)		7555	7789.3	5 (2.5%)
SCSA	$\beta_2=10^{-5}$	7991	8131.5	15 (7.5%)
	$\beta_2=7\times10^{-6}$	7549	7909.7	69 (34.5%)
	$\beta_2=3\times10^{-6}$	7525	7885.6	68 (34%)
	$\beta_2=10^{-6}$	7601	7942.4	64 (32%)

While fixing $\beta_1 = 3 \times 10^{-6}$, different performances of CSA and SCSA are compared by using various β_2 in Table V. In Table V, SCSA not only finds the valid tours more frequently than CSA, but also obtains the minimum tour length 666 that is far better than that of CSA, 722. Here, we also provide the optimal tour of our simulation

$$\begin{aligned}&[25-45-39-61-40-9-43-17-21-34-\\&12-33-62-54-48-67-11-56-65-64-\\&51-60-52-53-5-10-50-58-37-47-\\&16-23-1-36-29-13-31-70-35-69-\\&38-59-22-66-63-57-15-24-19-7-\\&2-4-18-6-41-42-32-3-8-26-55-\\&49-28-14-20-30-44-68-27-46].\end{aligned}$$

The simulation results show that CSA performs as well as SCSA in the small-size TSPs, such as 10-city, 21-city. But when used on the larger size TSP's, such as 52-city and 70-city, SCSA indeed achieves a much better performance than CSA.

IV. SOLVING THE CAP USING SCSA

In this section, we test SCSA in another combinatorial optimization problem, i.e., the CAP in cellular mobile communications. Due to rising demand and limited frequency channels available, an effective solution to the CAP is very important to the telecommunications

TABLE V
RESULTS OF CSA AND SCSA IN ST70 USING VARIOUS β_2 WITH 20 DIFFERENT INITIAL CONDITIONS

		Minimum Tour Length	Average Tour Length (not include the invalid)	The number of valid (%)
CSA ($\beta_2=1$)		722	722	1 (0.5%)
SCSA	$\beta_2=10^{-5}$	--	--	0 (0%)
	$\beta_2=7\times10^{-6}$	--	--	0 (0%)
	$\beta_2=3\times10^{-6}$	668	677.5	4 (20%)
	$\beta_2=10^{-6}$	666	686.1	7 (35%)

TABLE VI
TOTAL INTERFERENCE OBTAINED FROM SSA, CSA, AND SCSA FOR VARIOUS CAP2S

	SSA		CSA		SCSA	
	Ave	Min	Ave	Min	Ave	Min
HEX1	0.0	0	0.0	0	**0.0**	**0**
HEX2	0.1	0	0.0	0	**0.0**	**0**
HEX1	50.7	49	48.1	47	**47.1**	**46**
HEX2	20.4	19	18.9	18	**17.5**	**17**
HEX3	82.9	79	77.1	76	**76.8**	**74**
HEX4	21.0	17	17.7	17	**16.4**	**16**

industry and many excellent results have been obtained using different algorithms (e.g. [29]–[35]).

CAP's are often divided into two categories, i.e., CAP1 and CAP2 [34]. CAP1 is to minimize the span of channels subject to demand and interference-free constraints. CAP2 is to minimize interference subject to demand constraints. In this paper, we are concerned with only CAP2 because it is more useful in practical cases compared to CAP1 due to limited frequency channels available and high demands in mobile communications.

Suppose a mobile radio network has N cells and M frequency channels available. The number of calls in cell i is D_i. The constraints specify the minimum distance in the frequency domain by which two calls must be separated in order to guarantee an acceptably low signal/interference ratio in each cell. These minimum distances are stored in an $N\times N$ symmetric *compatibility matrix* C, i.e., C_{ij} is the required separation in frequency channels between a call in cell i and another call in cell j for the two calls to have no interference with each other.

Following Smith and Palaniswami [34], we map the CAP2 onto a neural network with $N\times M$ neurons. Assume x_{jk} the output of neuron jk and

$$x_{jk}=\begin{cases}1, & \text{if cell } j \text{ is assigned to channel } k\\ 0, & \text{otherwise}\end{cases} \tag{16}$$

for $j=1,\cdots,N$ and $k=1,\cdots,M$.

If $x_{jk}=x_{il}=1$, i.e., cell j is assigned to channel k and at the same time, cell i is assigned to channel l, the interference should be at its maximum when $k=l$ and decreases until the two channels are far enough that no interference exists. For simplicity, we assume linear reduction in interference with respect to channel distance [34]. The interference caused by such assignments is therefore given by the following cost tensor $P_{ji(m+1)}$ (where $m=|l-k|$ is the distance in the channel domain between channels k and l):

$$P_{ji(m+1)}=\max(0,P_{jim}-1),\ \forall m=1,\ldots,M-1 \tag{17}$$

$$P_{ji1}=C_{ji},\quad \forall j,i\neq j \tag{18}$$

$$P_{jj1}=0,\quad \forall j. \tag{19}$$

Then the CAP2 can be formulated to minimize the following cost:

minimize

$$f(x)=\sum_{j=1}^{N}\sum_{k=1}^{M}x_{jk}\sum_{i=1}^{N}\sum_{l=1}^{M}P_{ji(|k-l|+1)}x_{il} \tag{20}$$

subject to

$$\sum_{k=1}^{M}x_{jk}=D_j,\quad \forall j=1,\ldots,N \tag{21}$$

$$x_{jk}\in\{0,1\},\quad \forall j=1,\ldots,N,\ \forall k=1,\ldots,M \tag{22}$$

where $f(x)$ is the total interference and $x\equiv\{x_{jk}\}$.

With (20) and (21), the following computational energy may be defined as a sum of the total interferences and constraints:

$$E=\frac{W_1}{2}\sum_{j=1}^{N}\left(\sum_{k=1}^{M}x_{jk}-D_j\right)^2 +\frac{W_2}{2}\sum_{j=1}^{N}\sum_{k=1}^{M}x_{jk}\sum_{i=1}^{N}\sum_{l=1}^{M}P_{ji(|k-l|+1)}x_{il} \tag{23}$$

which is similar to (6), in the previous section. W_1 and W_2 are the weighting coefficients corresponding to the constraints and severity of interferences, respectively.

Connection weight W_{jkil} between neuron jk and neuron il can be obtained similarly using (5) and (23). Thus, the network dynamics of the SCSA for the CAP is as follows:

$$\begin{aligned}y_{jk}(t+1)&=ky_{jk}(t)-z(t)\left(x_{jk}(t)-I_0\right)\\&\quad+\alpha\left\{-W_1\sum_{q\neq k}^{M}x_{jq}+W_1D_j-W_2\right.\\&\quad\left.\times\sum_{p=1,p\neq j}^{N}\sum_{q=1,q\neq k}^{M}P_{j,p,|k-q|+1}x_{pq}\right\}\\&\quad+n(t).\end{aligned} \tag{24}$$

Here, we use the data set of a 21-cell cellular system, i.e., HEX1 in [29], [34], to test our algorithm. In HEX1, the number of cells N is 21 and the number of available channels M is 37. The demand is $D_1^T=(2,6,2,2,2,4,4,13,19,7,4,4,7,4,9,14,7,2,2,4,2)$, where A^T stands for the transposed matrix of matrix A. The minimum distance in frequency domain between two channels in the same cell is 2. A smaller test problem with 4 cells and 11 channels, i.e., the EX problems in Table V.

Table V shows the simulation results of the HEX and EX CAP's using SCSA and we also show the results obtained using SSA and CSA for a comparison. Each of those heuristics is run from ten different random initial conditions and an average is calculated. In Table V, "Min" means the minimum total interference (20) found during these ten times, and "Ave" is the average total interference for the ten runs. These results show that the overall interference obtained from SCSA is lower than the interference obtained from CSA, i.e., SCSA results in better channel assignments compared to CSA. The network parameters are set in Table VI.

V. CONCLUSIONS

In this paper, we proposed a noisy chaotic neural network (NCNN) or SCSA by adding noise to Chen and Aihara's transiently chaotic neural network. Application of this noisy chaotic neural network to a TSP and a CAP showed marked improvements over CSA. In contrast to the conventional SSA, SCSA restricts the random search to a subspace

TABLE VII
PARAMETERS FOR VARIOUS CAP2S

Problem	k	ε	I_0	$\beta_1\beta_2$	α	$Z(0)$	$A[n(0)]$	W_1	W_2
HEX1	0.9	1/250	0.65	0.0005	0.005	0.1	0.5	1.0	0.02
HEX2	0.9	1/250	0.65	0.0005	0.005	0.1	0.5	1.0	0.02
HEX1	0.9	1/250	0.05	0.0005	0.05	0.1	0.05	1.0	0.25
HEX2	0.9	1/250	0.05	0.0005	0.05	0.1	0.05	1.0	0.25
HEX3	0.9	1/250	0.05	0.0005	0.05	0.1	0.05	1.0	0.25
HEX4	0.9	1/250	0.05	0.0005	0.05	0.1	0.05	1.0	0.25

of chaotic attracting sets which is much smaller than the entire state space searched by SSA. In contrast to CSA, SCSA is not completely deterministic and continues to search after the disappearance of chaos. SCSA can be a powerful approach to solving a general class of combinatorial optimization problems and our future work will include applications of SCSA to other practical optimization problems.

ACKNOWLEDGMENT

The authors thank K. Aihara and L. Chen for answering questions about their work and for sending their recent publications.

REFERENCES

[1] C. A. Skarda and W. J. Freeman, "How brains make chaos in order to make sense of the world," *Brain Behav. Sci.*, vol. 10, pp. 161–195, 1987.
[2] L. Wang and J. Ross, "Osillations and chaos in neural networks: an exactly solvable model," in *Proc. National Academy of Sciences*, vol. 87, 1990, pp. 9467–9471.
[3] Y. Yao and W. J. Freeman, "Model of biological pattern recognition with spatially chaotic dynamics," *Neural Networks*, vol. 3, pp. 153–170, 1990.
[4] K. Aihara, T. Takabe, and M. Toyoda, "Chaotic neural networks," *Phys. Lett. A*, vol. 144, no. 6, 7, pp. 333–340, 1990.
[5] H. Nozawa, "A neural network model as a globally coupled map and applications based on chaos," *Chaos*, vol. 2, no. 3, pp. 377–386, 1992.
[6] L. Wang and D. L. Alkon, Eds., *Artificial Neural Networks: Oscillations, Chaos, and Sequence Processing*. Los Alamitos, CA: IEEE Comput. Soc. Press, 1993.
[7] H. Nozawa, "Solution of the optimization problem using the neural network model as a globally coupled map," in *Toward the Harnessing of Chaos*, M. Yamaguti, Ed. Amsterdam, The Netherlands: Elsevier Science, 1994, pp. 99–114.
[8] L. Chen and K. Aihara, "Transient chaotic neural networks and chaotic simulated annealing," in *Toward the Harnessing of Chaos*, M. Yamguti, Ed. Amsterdam, The Netherlands: Elsevier Science, 1994, pp. 347–352.
[9] ——, "Chaotic simulated annealing by a neural network model with transient chaos," *Neural Networks*, vol. 8, no. 6, pp. 915–930, 1995.
[10] Y. Horio and K. Suyama, "Experimental verification of signal transmission using synchronized SC chaotic neural networks," *IEEE Trans. Circuits Syst. I*, vol. 42, pp. 393–395, July 1995.
[11] L. Wang, "Oscillatory and chaotic dynamics in neural networks under varying operating conditions," *IEEE Trans. Neural Networks*, vol. 7, pp. 1382–1388, June 1996.
[12] 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.
[13] H. Lu, Y. He, and Z. He, "A chaos-generator: analyzes of complex dynamics of a cell equation in delayed cellular neural networks," *IEEE Trans. Circuits Syst. I*, vol. 45, pp. 178–181, Feb. 1998.
[14] I. Tokuda, K. Aihara, and T. Nagashima, "Adaptive annealing for chaotic optimization," *Phys. Rev. E*, vol. 58, pp. 5157–5160, 1998.
[15] S. Haykin and J. Principe, "Making sense of a complex world [chaotic events modeling]," *IEEE Signal Processing Mag.*, vol. 15, pp. 66–81, May 1998.
[16] L. Wang and K. Smith, "On chaotic simulated annealing," *IEEE Trans. Neural Networks*, vol. 9, pp. 716–718, July 1998.
[17] K. Jin'no, T. Nakamura, and T. Saito, "Analysis of bifurcation phenomena in a 3-cells hysteresis neural network," *IEEE Trans. Circuits Syst. I.*, vol. 46, pp. 851–857, July 1999.
[18] L. Chen and K. Aihara, "Global searching ability of chaotic neural networks," *IEEE Trans. Circuits Syst. I*, vol. 46, pp. 974–993, Aug. 1999.
[19] T. Kwok and K. A. Smith, "A unified framework for chaotic neural network approaches to combinatorial optimization," *IEEE Trans. Neural Networks*, vol. 10, no. 4, pp. 978–981, 1999.
[20] R. Kozma and W. J. Freeman, "Encoding and recall of noisy data as chaotic spatio-temporal memory patterns in the style of the brains," in *Proc. Int. Joint Conf. Neural Networks*, vol. 5, Como, Italy, July 24–27, 2000, pp. 33–38.
[21] K. Hirasawa, J. Murata, J. Hu, and C. Z. Jin, "Chaos control on universal learning networks," *IEEE Trans. Syst., Man, Cybern. C*, vol. 30, pp. 95–104, Feb. 2000.
[22] K. Tanaka, Y. Horio, and K. Aihara, "A modified algorithm for the quadratic assignment problem using chaotic-neuro-dynamics for VLSI implementation," in *Proc. Int. Joint Conf. Neural Networks (IJCNN '01)*, vol. 1, 2001, pp. 240–245.
[23] J. J. Hopfield, "Neurons with graded response have collective computational properties like those of two-state neurons," in *Proc. Nat. Acad. Sci. USA*, vol. 81, May 1984, pp. 3088–3092.
[24] J. J. Hopfield and D. W. Tank, "Neural computation of decisions in optimization problems," *Biolog. Cybern.*, vol. 52, pp. 141–152, 1985.
[25] G. V. Wilson and G. S. Pawley, "On the stability of the traveling salesman problem algorithm of Hopfield and Tank," *Biolog. Cybern.*, vol. 58, p. 63, 1988.
[26] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, pp. 671–680, 1983.
[27] E. Aarts and J. Korst, *Simulated Annealing and Boltzmann Machines: A Stochastic Approach to Combinatorial Optimization and Neural Computing*. New York: Wiley, 1989.
[28] G. Reinelt, "TSPLIB—A traveling salesman problem library," *ORSA J. Comput.*, vol. 3, no. 4, pp. 376–384, 1991.
[29] A. Gamst and W. Rave, "On frequency assignment in mobile automatic telephone systems," in *Proc. GLOBECOM'82*, 1982, pp. 309–315.
[30] A. Gamst, "Some lower bounds for a class of frequency assignment problems," *IEEE Trans. Veh. Technol.*, vol. VT-35, pp. 8–14, Feb. 1986.
[31] D. Kunz, "Channel assignment for cellular radio using neural networks," *IEEE Trans. Veh. Technol.*, vol. 40, pp. 188–193, Feb. 1991.
[32] M. Duque-Anton, D. Kunz, and B. Ruber, "Channel assignment for cellular radio using simulated annealing," *IEEE Trans. Veh. Technol.*, vol. 42, pp. 14–21, Feb. 1993.
[33] N. Funabiki and Y. Takefuji, "A neural network parallel algorithm for channel assignment problems in cellular radio networks," *IEEE Trans. Veh. Technol.*, vol. 41, pp. 430–437, Nov. 1992.
[34] K. Smith and M. Palaniswami, "Static and dynamic channel assignment using neural networks," *IEEE J. Select. Areas Commun.*, vol. 15, pp. 238–249, Feb. 1997.
[35] J.-S. Kim, S. H. Park, P. W. Dowdy, and N. M. Nasrabadi, "Cellular radio channel assignment using a modified Hopfield network," *IEEE Trans. Veh. Technol.*, vol. 46, pp. 957–967, Nov. 1997.