Noisy Chaotic Neural Networks for Combinatorial Optimization

Lipo Wang and Haixiang Shi

School of Electrical and Electronic Engineering, Nanyang Technological University,
Block S1, Nanyang Avenue, Singapore 639798

Summary. In this Chapter, we review the virtues and limitations of the Hopfield neural network for tackling NP-hard combinatorial optimization problems (COPs). Then we discuss two new neural network models based on the noisy chaotic neural network, and applied the two methods to solving two different NP-hard COPs in communication networks. The simulation results show that our methods are superior to previous methods in solution quality. We also point out several future challenges and possible directions in this domain.

1 Introduction

Since Hopfield and Tank's innovative work on solving the traveling sales man problem (TSP) using neural networks, there are numerous research efforts on applying the Hopfield neural network (HNN) and HNN-based neural network techniques to solving combinatorial optimization problems (COPs) [3, 10, 26, 42, 36, 27, 24]. However, Wilson and Pawley [43] raise doubts on the validity of the HNN to solving COPs after they were unable to reproduce the results in Hopfield and Tank's work. They claimed that the original HNN formulation for the TSP is unreliable even for small-sized problems. Many explanations for the poor solution quality of the TSP had been made in terms of energy function formulation [8, 29, 4] and parameter selection [15, 25, 22, 7].

Poor solution quality, dependences on energy function formulations, and the difficulties in parameter selection of the original HNN are due to its gradient descent dynamics leading local minima. This chapter introduces the chaotic neurodynamics which can help to avoid local minima and converge to better solutions in solving NP-hard combinatorial optimizations (COPs).

In 1983, Kirkpatrick *et al* [23] developed simulated annealing, which emulates the annealing processing in metals by first heating the metal to its melting point and then slowly cooling the material. Because of the stochastic nature of the optimization process, simulated annealing can also be called stochastic simulated annealing (SSA) [23]. SSA is known to relax to

Lipo Wang and Haixiang Shi: *Noisy Chaotic Neural Networks for Combinatorial Optimization*, Studies in Computational Intelligence (SCI) **63**, 467–487 (2007)
www.springerlink.com

a global minimum with probability 1 if the annealing takes place sufficiently slowly, i.e., at least inversely proportional to the logarithm of time [13]. In a practical term, 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 with a reasonably small exponent [40].

SSA has been widely used in various optimization problems with great success, but it still suffers from several deficiencies:

1) For large problems, the method requires prohibitively long relaxation time in order to find solutions with acceptable quality, i.e., SSA consumes too much iteration time due to its Monte Carlo scheme. To guarantee convergence to an exact solution, SSA will require more iterations than complete enumeration does for some problems [5].
2) SSA often requires subtle adjustments of parameters in the annealing schedule, such as the length of the temperature steps during annealing, the temperature range, the number of re-starts and re-direction during the search [32, 1, 19, 20, 23].

In order to improve the searching ability of the SSA, complex neurodynamics such as chaotic simulated annealing (CSA) was proposed [46]. Compared with the gradient descent dynamics of the HNN models and neural networks with SSA dynamics, neural networks with CSA have a richer spectrum of dynamic behaviors, such as stable fixed points, periodic oscillations, and chaos.

Nozawa demonstrated the search ability of the chaotic neural networks (CNN) [30, 45]. Chen and Aihara [46, 5] proposed the chaotic simulated annealing (CSA) by starting with a sufficiently large negative self-coupling in the Aihara-Takabe-Toyoda [2] 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 (TCNN). Their computer simulations showed that the CSA leads to good solutions for the TSP much more easily compared to the Hopfield-Tank approach [16, 17] and SSA. Chen and Aihara [6] 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.

Other kinds of CSA have also been proposed. Wang and Smith proposed another chaotic annealing by annealing the time-step in the Euler approximation of the continuous Hopfield network [41]. Hayakawa et al. [14] obtained the CSA by adding the chaotic noise into the Hopfield network. Zheng et al. [47] improved the Wang-Smith's chaotic simulated annealing which reaps the benefits of Wang-Smith model and Chen-Aihara model.

There are mainly three significant differences between SSA and CSA [5]:

1) SSA is stochastic on the basis of the Monte Carlo scheme while CSA is deterministic with transiently chaotic dynamics.
2) The convergent processing of SSA is controlled by stochastic "thermal" *fluctuations* while that of CSA is controlled by *bifurcation* structures.
3) SSA essentially searches all possible states while temporally changing probability distributions, whereas CSA restricts to a fractal subspace. Because the searching region in CSA is usually smaller compared with the entire state space, CSA can be expected to perform efficient searching if the restriction is adequate to include a global optimum state or some near-global optimum states.

However, 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 [37]. Different from the searching direction of SSA that is probabilistically determined by mutual interactions among neurons, CSA is uniquely determined by mutual interactions among neurons. In practical terms, this means that CSA sometimes may not be able to provide a good solution at the end of annealing for some initial conditions of the network, no matter how slowly annealing takes place, i.e., CSA sometimes may not be able to provide a good solution at the conclusion of annealing even after a long time of searching.

Wang and Tian [40] proposed a new approach to simulated annealing, i.e., stochastic chaotic simulated annealing (SCSA), using a noisy chaotic neural network (NCNN) by adding decaying stochastic noise into the TCNN. Compared with CSA, SCSA performs stochastic searching both before and after chaos disappears and is more likely to find optimal or sub-optimal solutions.

2 Mathematical Formulations of the NCNN

The NCNN model is described as follows [40]:

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

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

$$z(t+1) = (1-\beta_1) z(t) \,, \tag{3}$$

$$n(t+1) = (1-\beta_2) n(t) \,, \tag{4}$$

where the notations are:

x_{jk} : output of neuron jk ;

y_{jk} : input of neuron jk ;

w_{jkil}: connection weight from neuron jk to neuron il, with $w_{jkil} = w_{iljk}$ and $w_{jkjk} = 0$;

$$\sum_{i=1,i\neq j}^{N} \sum_{l=1,l\neq k}^{M} w_{jkil} x_{jk} + I_{ij} = -\partial E/\partial x_{jk} \tag{5}$$

E : energy function;

I_{jk} : input bias of neuron jk ;

k : damping factor of nerve membrane ($0 \leq k \leq 1$);

α : positive scaling parameter for inputs ;

β_1 : damping factor for neuronal self-coupling ($0 \leq \beta_1 \leq 1$);

β_2 : damping factor for stochastic noise ($0 \leq \beta_2 \leq 1$);

$z(t)$: self-feedback connection weight or refractory strength ($z(t) \geq 0$), $z(0)$ is a constant;

I_0 : positive parameter, which is used as threshold for each neuron, can be a fixed number or variable one;

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

$n(t)$: random noise injected into the neurons, in $[-A, A]$ with a uniform distribution;

$A[n]$: amplitude of noise n.

This NCNN model is a general form of chaotic neural networks with transient chaos and decaying noise. In the absence of noise, i.e., $n(t) = 0$, for all t, the NCNN as proposed in eqns. (1) - (4) reduces to the TCNN in [5].

3 Gradual Noisy Chaotic Neural Network (G-NCNN)

Compared with the constant number of neurons used in conventional neural networks, the gradual neural network [11] adopts an increasing number of neurons. Usually, the number of neurons that a neural networks needs for solving a COP is determined by the problem, e.g., N-city TSP problem, $N \times N$ neurons are needed to compose a solution space. For the NCNN, N^2 neurons are needed at the start of neural network updating. But if using the gradual neural network, the neural network needs only a fraction of $N \times N$ neurons at the beginning of neuron computation, which are the most likely selected group of neurons in the final results (each neuron stands for a status in the solution matrix based on different problems, e.g., if neuron ij fires, it means the ith element is assigned to jth element in the assignment problem). The rest of all $N \times N$ neurons are gradually added to the current commutating group.

The reason why the gradual scheme is needed instead of a constant number of neurons is that the gradual scheme can be considered as one kind of objective in the optimization. There are objectives in COPs that need to find the minimal "cost", where the cost can have many different meanings in different problems, e.g., interference, delay, or length of a path. In order to achieve the objective that the solution found by the neural network is with the minimal cost, neurons are divided into several groups and activated or added in several stages with each stage only adopt one group of neurons with smallest cost left over. Through the gradual scheme, we do not need to formulate the objective ("minimal cost") in the energy function, because we think that the objective is realized in the gradual expansion stages which let the neurons with smaller cost be included in neuron computation in an earlier stage. When dealing with this kind of objective, we can adopt the gradual scheme naturally through the following steps [11]:

1) Assume there are $N \times M$ neurons needed for the problem. Compute the cost matrix $C[N \times M]$, i.e., the cost of neuron ij if neuron ij is selected in the final solution. Here N and M can have the same value.
2) Sort the neuron in ascending order of the cost.
3) Divide the $N \times M$ neurons into P groups $(G_1, G_2, \ldots, G_P)$, with each group have p neurons, where p is the maximum integer less than or equal to NM/P. Group G_1 is the group of neurons with smallest cost and the other groups G_2 to G_P contain the neurons with larger cost with an ascending order of cost.
4) Add the neurons in the first group G_1 to the neural network and let the neural network update. If a feasible solution is found, exit, the solution found is the final solution. If the pre-defined steps are used up and still no solutions are found, then apply the gradual expansion scheme by adding the neurons in G_2 and let the neural network update again.

We found that although using gradual scheme can help to reduce the objectives in optimizations, it cannot guarantee to achieve the objective. But it can be made up by adding the objective again into the energy function.

4 Noisy Chaotic Neural Network with Variable Threshold (NCNN-VT)

Besides the gradual noisy chaotic neural network, another extension of the noisy chaotic neural network is proposed by using the variable threshold in the neural network, instead of the fixed threshold value. It can be found that the variable threshold can be used to achieve the objective of the NP-hard optimization problem, and again reduces the number of objectives needed to be formulated in the energy function.

4.1 Adaptive Mapping Scheme

Adaptive Mapping Scheme (AMS) aims to map the objective of optimization problem into the probability of firing of each neuron. If the objective of the problem is to find the solution with minimal cost, then the neurons with smaller cost will have larger probability to be selected in the final solution matrix. By investigating the single neuron dynamics of the NCNN model, it can be seen that the positive parameter I_0 in the NCNN model is responsible for the neuron firing probability. The single neuron dynamics of the NCNN by varying the parameter I_0 is shown in Figs. 1 to 4 (the x-axis is the time steps t, the y-axis is the output of neuron $x_{ij}(t)$). We can see clearly that if the value of parameter I_0 is 0.3, the output of neuron is very close to 0.3 after the network passes the last bifurcation-2 in Fig. 2. After the bifurcation point, the neural network slowly converge to a stable point at about 0.5. The same pattern can be observed in other figures. It means that different value of I_0 ($0 < I_0 < 1$) induces different probability of neuron firing. The neuron with large value of I_0 will be selected to be firing (neuron output equals 1) with bigger probability, whereas, the small value of I_0 will result in less chance the neuron to be firing.

4.2 Model Definition

The difference between the NCNN and the NCNN-VT model is that the positive parameter I_0 in eqn. (2) of the NCNN becomes a variant labeled as I_{jk}, where jk stands for the neuron jk, i.e.,

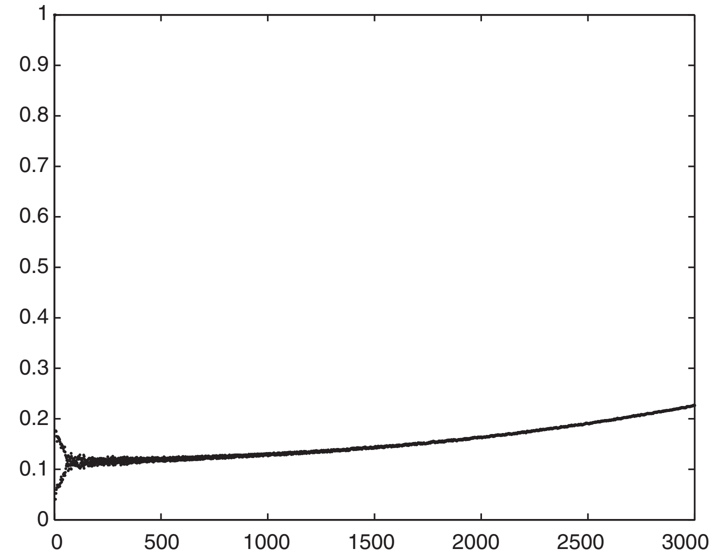

Fig. 1. The single neuron dynamics of the noisy chaotic neural network with variable threshold, $I_0 = 0.1$

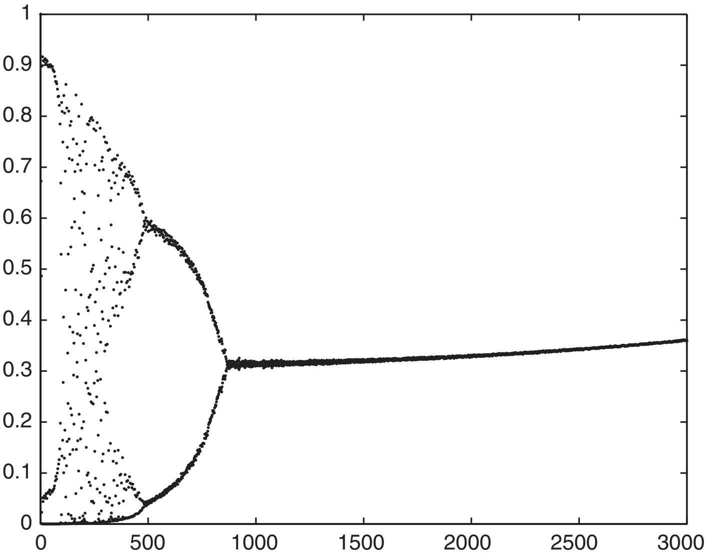

Fig. 2. The single neuron dynamics of the noisy chaotic neural network with variable threshold, $I_0 = 0.3$

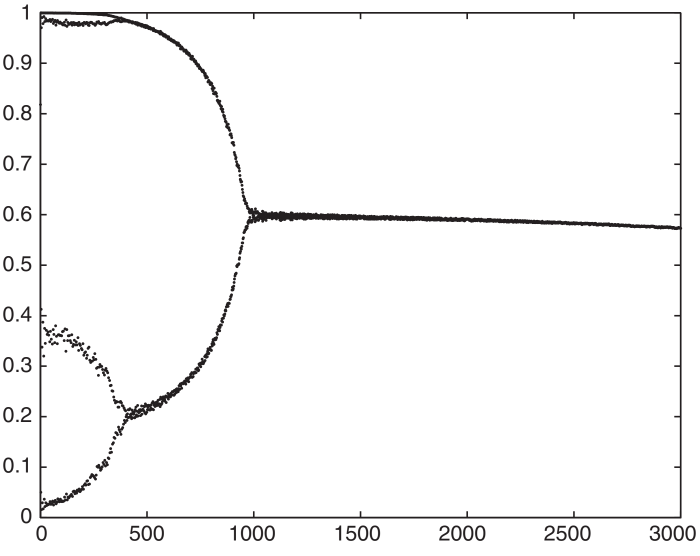

Fig. 3. The single neuron dynamics of the noisy chaotic neural network with variable threshold, $I_0 = 0.6$

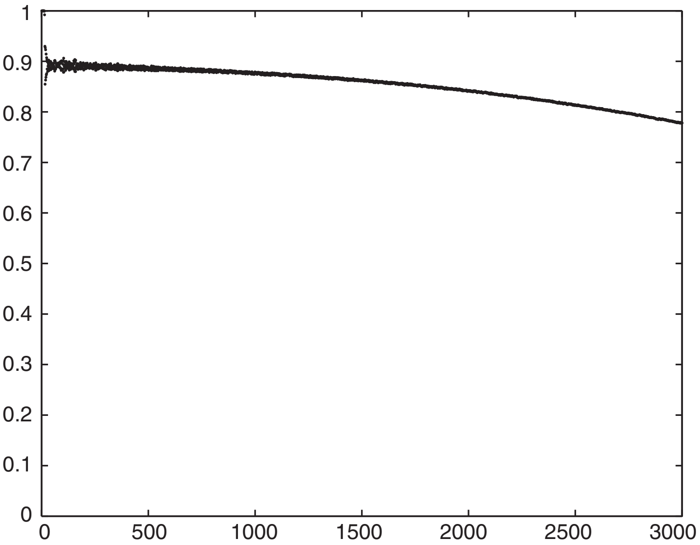

Fig. 4. The single neuron dynamics of the noisy chaotic neural network with variable threshold, $I_0 = 0.9$

$$y_{jk}(t+1) = ky_{jk}(t) + \alpha \left\{ \sum_{i=1, i\neq j}^{N} \sum_{l=1, l\neq k}^{M} w_{jkil} x_{jk}(t) + I_{ij} \right\}$$
$$-z(t) \left[x_{jk}(t) - I_{jk} \right] + n(t) \,. \tag{6}$$

where I_{jk} is not a constant parameter but a variable which determines the selection of firing of each neuron. The value of I_{jk} is related to the problem optimization term.

4.3 Mapping Functions

The mapping function connects the problem's cost with the parameter I_{ij}.

$$I_{jk} = f(c_{jk}), \quad j = 1, 2, \ldots, N; k = 1, 2, \ldots, M \tag{7}$$

The mapping function can be a linear or nonlinear function which transform the cost into the probability value between 0 and 1. Normally, a linear mapping is adopted:

$$f(c_{jk}) = 1 - \frac{c_{jk} - c_{min}}{c_{max} - c_{min}} \,. \tag{8}$$

where c_{jk} is the jk element in the cost matrix C, c_{max} is the maximum value among the cost matrix and c_{min} is the minimum value in the cost matrix.

From the mapping functions in eqn. (8), it can be seen that the neuron ij with smaller cost d_{ij} will have larger or higher probability to fire while the ones with larger cost will be inhibited to fire.

5 Using G-NCNN to Solve the Broadcast Scheduling Problem

5.1 Problem Introduction

In a time-division-multiple-access (TDMA) network, time is divided into frames and each TDMA frame is a collection of time slots. A time slot has a unit time length required for a single packet to be communicated between adjacent nodes. When nodes transmit simultaneously, conflicts will occur if the nodes are in a close range. Therefore, adjacent nodes must be scheduled to transmit in different time slots, while nodes some distance away may be arranged to transmit in the same time slot without causing conflict [39]. The goal of the broadcast scheduling problem (BSP) is to find an optimal TDMA frame structure that fulfills the following two objectives. The first is to schedule transmissions of all nodes in a minimal TDMA frame length without any conflict. The second is to maximize channel utilization or total conflict-free transmissions.

5.2 Energy Function Formulation

For the two objectives, a two-stage methods using the NCNN are used to solve the problem. The first stage aims to find the minimal TDMA frame cycle length (M), whereas the objective in the second stage is to maximize the total node transmissions in order to fulfill the channel utilization.

The G-NCNN consists of $M \times N$ neurons. M is initially set as its lower bound value L_m, which can be easily obtained using graph theory [21]. The network can be formulated to a graph G = (V, E). The graph G can be transformed into $G' = (V, E')$, where E in G stands for one-hop-away edges, and E' in G' stands for one-hop-away and two-hop-away edges. The lower bound is:

$$L_m = \omega(G') \,. \tag{9}$$

where $\omega(G')$ is the maximal cardinality of a clique in G' [33].

The energy function E_1 for the first stage is given as follows [9]:

$$E_1 = \frac{W_1}{2}\sum_{j=1}^{N}(\sum_{k=1}^{M} x_{jk} - 1)^2 + \frac{W_2}{2}\sum_{j=1}^{N}\sum_{i=1}^{M}\sum_{k=1,k\neq i}^{N} d_{jk}x_{ij}x_{ki} \,, \tag{10}$$

where W_1 and W_2 are weighting coefficients. The W_1 term represents the constraint that each of the N nodes in the PRN must transmit exactly once during each TDMA cycle. The W_2 term indicates the constraint that any pair of nodes which is one-hop-away or two-hop-away must not transmit simultaneously during each TDMA cycle.

From eqn. (2), eqn. (5), and eqn. (10), we obtain the dynamics of the G-NCNN as follows:

$$y_{jk}(t+1) = ky_{jk}(t) + \alpha \left\{ -W_1(\sum_{k=1}^{M} x_{jk} - 1) - W_2 \sum_{k=1,k\neq j}^{N} d_{jk}x_{ki} \right\} \\ -z(t)\left[x_{jk}(t) - I_0\right] + n(t)\,. \tag{11}$$

The G-NCNN stops when the it finds a feasible assignment and the current number of time slots together with its transmission assignments are the optimal results for phase I of the BSP. In this chapter, different from the GNN in [9], where the neurons are expanded gradually at every P iterations during the iterative computation of the neural network, we implement the GES based on a convergence index $\delta(t)$ of the network energy, which we defined as:

$$\delta(t) = \sum_{q=t-4}^{t} \left|E(q) - E(q-1)\right| / E(0)\,. \tag{12}$$

where $E(q)$ is the value of energy function at time step q. If index $\delta(t)$ is less than a very small value, e.g., $\delta(t) < 10^{-4}$ in our simulation, the neural network is considered as having fully converged. If the network has converged but no feasible solutions are found using the current number of time slots, the number of time slots is increased by 1, i.e., $M \to M+1$, and the G-NCNN re-starts to search for optimal solutions with the updated number of neurons.

In the second stage, the minimal TDMA frame length M is found and each node is assigned with one and exactly one time slot. In phase II, we aim at maximizing the channel utilization by adding as many conflict-free transmissions as possible to the TDMA frame. Because in phase I one node is assigned with exactly one slot in order to find a minimal frame length, there are many nodes which can use other time slots without violating the no-conflict constraint. Thus, additional transmissions may be found on some nodes but frame length M and the assigned transmissions in phase I are fixed [9].

$$E_2 = \frac{W_3}{2}\sum_{j=1}^{N}\sum_{i=1}^{M}\sum_{k=1,k\neq i}^{N} d_{jk}x_{ij}x_{ki} + \frac{W_4}{2}\sum_{j=1}^{N}\sum_{i=1}^{M}(1-x_{ij})^2\,, \tag{13}$$

where W_3 and W_4 are weighting coefficients. W_3 represents the constraint term that any pair of nodes which is one-hop-away or two-hop-away must not transmit simultaneously during each TDMA cycle. W_4 is the optimization term which maximizes the total number of firing neurons.

From eqn. (2), eqn. (5), and eqn. (13), we obtain the dynamics of the NCNN for phase II of the BSP as follows:

$$y_{jk}(t+1) = ky_{jk}(t) + \alpha \left\{ -W_3 \sum_{k=1,k\neq j}^{N} d_{jk}x_{ki} + W_4(1 - x_{ij}) \right\}$$
$$-z(t)\left[x_{jk}(t) - I_0\right] + n(t)\,. \qquad (14)$$

The NCNN is updated cyclically and asynchronously, which means we update the neurons in two loops and the neuron is selected to be computed in a fixed order. The new state information of a updated neuron is immediately available for the other neurons in the computation. The iteration is terminated once a feasible transmission schedule is obtained, i.e., the transmissions of all nodes are conflict-free.

5.3 Results Discussions

The benchmark examples are get from other published papers. Each problem is simulated 50 different times and the best and the average values are displayed in Table 1.

Table 1. Comparisons of average delay time η and numbers of time slots M and computation time T obtained by the G-NCNN and other algorithms for the three benchmark problems in 50 runs, where Best/Avg stands for the best value and average value in multiple runs.

Case		BM 1	BM 2	BM 3
		Best/Avg	Best/Avg	Best/Avg
G-NCNN	η	6.8/7.0	9.0/9.5	5.7/6.1
	M	8/8.0	10/10.5	8/8.0
	T	6.0/7.2	16.0/18.3	6.0/6.5
HNN-GA	η	7.0/7.0	9.3/9.	6.3/6.5
	M	8/8.0	10/10.0	8/9.0
	T	4.0/4.7	17/19.0	13.0/14.0
SVC	η	7.2/7.4	10.0/12.0	6.8/7.2
	M	8/8.0	10/11.0	8/10.0
	T	2.5/2.8	15.0/15.4	10.0/12.0
GNN	η	7.1/7.2	9.5/10.0	6.2/6.5
	M	8/8.0	10/10.5	8/8.5
	T	15.0/16.4	18.0/20.0	17.0/19.5
MFA	η	7.2/7.5	10.5/12.5	6.9/8.2
	M	8/9.0	12/13,5	9/10.0
	T	25.0/7.2	32.5/38.5	28.0/29.0

From the results, we can see that our G-NCNN method can find shorter frame length than previous methods. In addition, our proposed method can find the smallest average time delay η among all methods in all three cases. The computation time (T in the table) needed by the G-NCNN is relatively lower than the previous MFA and comparable to other methods.

6 Applying the NCNN-VT to the Frequency Assignment Problem

6.1 Problem Introduction

Due to the economic effect on the average person in everyday life, there is an increasing number of satellites in geostationary orbits. In order to accommodate the crowded satellites in the same orbit, optimal design of satellites are necessary in order to provide high quality transmissions. In satellite communication systems, the major impairments in transmission design include thermal noise, rain attenuation, inter-modulation, and co-channel interference, among which, co-channel interference dominates because it seriously affects system design and operation [28]. Hence, the reduction of the co-channel interference has arisen as a major problem in satellite communications with the dramatic increase of geostationary satellites in orbits.

In order to reduce the interference, re-arrangements of frequency assignments, which take advantage of carrier interleaving, is thought as an effective way in practical situations. Early efforts have focused on various analytical methods for evaluations of co-channel interference [31, 18] and very few systematic methods have been adopted to optimize frequency assignments to reduce co-channel interference. The later work of Muzuike and Ito [28] revealed the importance of mathematical models for reduction of co-channel interference. They proposed a basic mathematical model to formulate the co-channel interference reduction problem as the "assignment problem". The assignment problem aims to minimize the largest interference among carriers. Fig. 6 shows the co-channel interference model for the system in Fig. 5. In the inter-system context, the two sets of carriers share the same frequency band. One set of carriers (C_{11} to C_{13}) is in satellite system 1 and the other set of carriers (C_{21} to C_{24}) corresponds to satellite system 2 in Fig. 5.

In the model shown in Fig. 6, carrier frequencies for one set of carriers are to be rearranged while keeping the other set fixed, i.e., the frequencies for carriers in system 2 are chosen to be re-arranged while the frequencies for system 1 are fixed. Fig. 5 shows the inter-system co-channel interference between two adjacent satellite systems. The communications are assumed to operated between F_a and F_b as showed in Fig. 6, where F_a and F_b are frequency band. The co-channel interference can be evaluated by calculating the each pair of carriers using the same frequency, which varies with different pairs. The objective of this assignment problem is to find the optimal

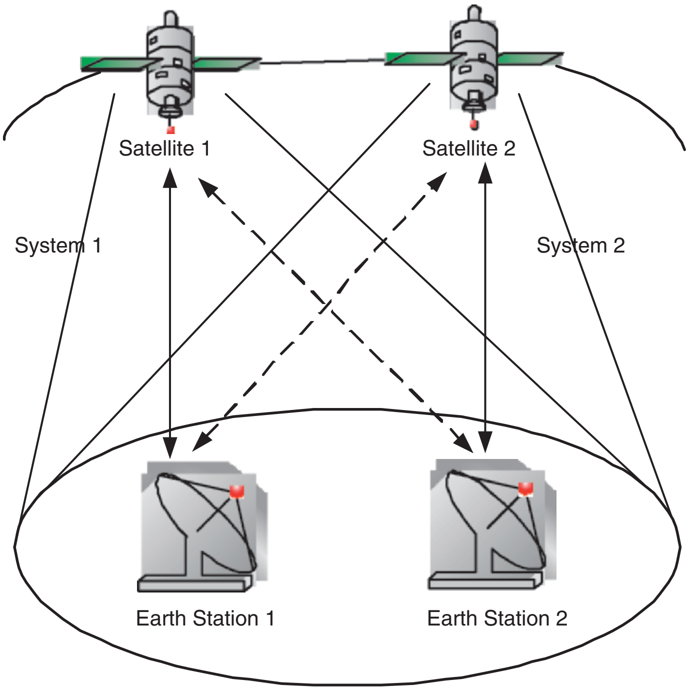

Fig. 5. Inter-system co-channel interference

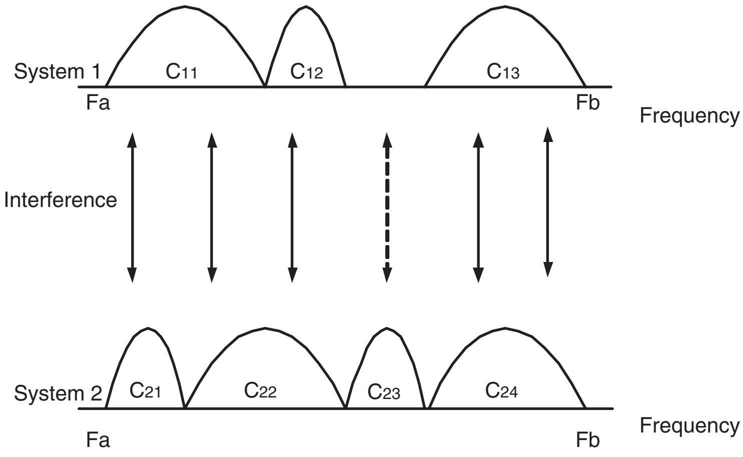

Fig. 6. Co-channel interference model for the system in Fig. 5, where the C_{xy} stands for the carrier y in system x, e.g., C_{23} stands for the carrier 3 in system 2

assignment of frequencies in system 2 in order to reduce the co-channel interference. The largest interference is considered as a limiting factor, and the optimal assignment is the one which can minimize the limiting factor.

In this chapter, we use a two-dimensional neural network which consists of $N \times M$ neurons for the FAP of N carriers and M segments. The output of each neuron V_{ij} will be converted into binary values V_{ij}^d. V_{ij}^d represents whether carrier i is assigned to segment $j-(j+c_i-1),(i=1,\ldots,N;j=1,\ldots,M)$, where c_i indicates the length of carrier i, i.e.:

$$V_{ij}^d = \begin{cases} 1 \text{ carrier } i \text{ is assigned to segment } j-(j+c_i-1), \\ 0 \text{ otherwise.} \end{cases}$$

Fig. 7 shows the neural network formulation for the 4-carrier-6-segment problem. This neural network consists of 24 (= 4×6) neurons as shown in Fig. 7 (a). Fig. 7 (b) is the convergence state, with the black squares stand for the neurons with output $V_{ij}^d = 1$. Fig. 7 (c) shows the full assignment for each carrier. And Fig. 7 (d) is the final frequency assignment for the FAP. Note that it can be easy to expand the convergence state in Fig. 7 (c) to the final assignment in Fig. 7 (d) given the carrier length for each carrier. We provide only the solution format in Fig. 7 (c) to represent the final assignment in this Chapter.

Our objective is to minimize the largest element of the interference matrix selected in the assignment and at the same time minimize the sum of interference of all selected elements. Thus we define the choice of the mapping function of I_{ij} as follows:

$$\begin{aligned} I_{ij} &= 1 - \frac{d_{ij} - d_{i,min}}{d_{i,max} - d_{i,min}} \\ &= \frac{d_{i,max} - d_{ij}}{d_{i,max} - d_{i,min}} . \end{aligned} \tag{15}$$

where d_{ij} is the ij-th element in cost matrix $D = (d_{ij}, i = 1,\ldots,N; j = 1,\ldots,M)$, and $d_{i,max}$[1] is the maximum value in line i of matrix D and $d_{i,min}$ is the minimum value line i of matrix D. We will label $d_{i,max}$ as d_{max} and $d_{i,min}$ as d_{min} for simplicity.

Through the mapping in eqn. (15), not only can we achieve the objectives of FAP, but also separate the objective from the energy function, which will make the tuning of weighting coefficients in the energy function easier without the need to balance the optimization term and constraint term in one energy function. Moreover, it will improve the convergence speed of the noisy chaotic neural network as shown in the result discussion section.

[1] Note that the maximum value of cost does not include the infinity value of in the interference matrix. Actually the neurons corresponding to the infinite interference will never fire due to its inhibitive cost.

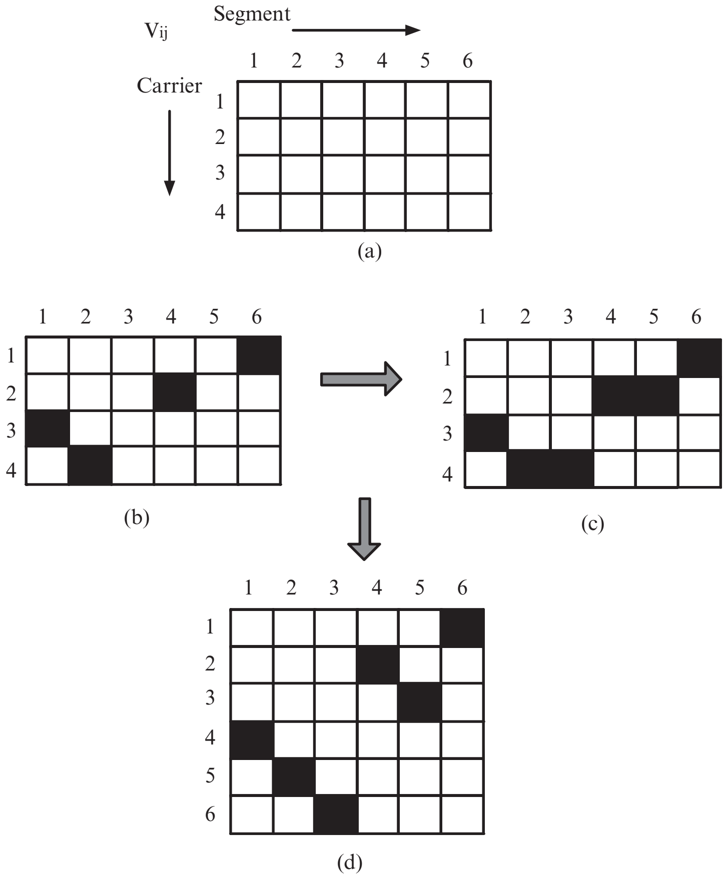

Fig. 7. The neural network formulation for the FAP. (a) The 24 neurons for the 4-carrier-6-segment FAP. (b) the convergence state of the neural network. (c) the full assignment of the neural network formulation. (d) the final assignment of the segments for the FAP through the expansion from the neural network formulation

6.2 Energy Function and Results

Recall that the goal of the FAP has been separated from the constraints. Only the two constraints of the FAP need to be formulated in the energy function. The first constraint of an N-carrier-M-segment problem is that each first segment of the N carriers in system 2 must be assigned to one and only one of the M segments. Hence, one and only one neuron among the M neurons for each carrier has output 1. Then the first constraint can be formulated as [11]:

$$E_1 = \sum_{i=1}^{N}(\sum_{q=1}^{M} V_{iq} - 1)^2 \ . \tag{16}$$

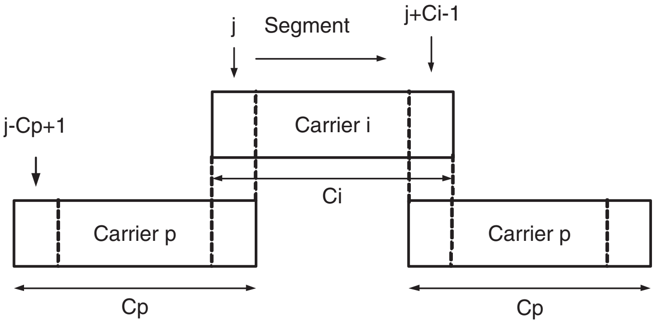

Fig. 8. Violation condition for the second and third constraints of the FAP

The second and third constraints are that each segment in system 1 can be assigned to at most one segment in system 2 and the assignment of carriers in system 2 should be in consecutive segments in system 1 in the same order. If carrier i is assigned to segment $j-(j+c_i-1)$, any other carrier $p(p \neq i)$ must not be assigned to the consecutive segment $(j-c_p+1)-(j+c_i-1))$. The violation condition of the two constraints is shown in Fig. 8, where c_i is the carrier length of carrier i. In other words, if carrier i is assigned with consecutive c_i segments, the segments occupied by any other carrier cannot occupy with the segment $j-(j+c_i-1)$. The first segment of each carrier $p,(p \neq i)$ should be $(j-c_p+1)$ before and $j-(j+c_i-1)$ after the first segment of carrier i.

Hence these two constraints give raise to the second part of the energy function to be minimized, as formulated in [11]:

$$E_2 = \sum_{i=1}^{N} \sum_{j=1}^{M} \sum_{\substack{p=1 \\ p \neq i}}^{N} \sum_{q=j-c_p+1}^{j+c_i-1} V_{ij} V_{pq} \,. \tag{17}$$

Note that because $(j-c_p+1)$ can be negative and $(j+c_i-1)$ can exceed the total number of segments M. The original formulation in eqn. (17) [11] has errors in dealing with the bounds and produces the program bugs when searching the solutions. We formulate the second term in our energy function in a revised version as follows:

$$E_2' = \sum_{i=1}^{N} \sum_{j=1}^{M} \sum_{\substack{p=1 \\ p \neq i}}^{N} \sum_{q=max(j-c_p+1,1)}^{min(j+c_i-1,M)} V_{ij} V_{pq} \,. \tag{18}$$

where function $max(x, y)$ returns the maximum value between (x, y) two numbers and $min(x, y)$ finds the minimum value between (x, y).

We use the following convergence term in our energy function to help the neuron output converge to the corner (0 or 1) of the hypercube:

Table 2. Comparisons of the simulation results (largest interference and total interference) obtained by the NCNN-VT, GNN and HopSA in the benchmark examples.

Instance	GNN[11]		HopSA[34]		NCNN-VT	
	largest	total	largest	total	largest	total
BM 1	30	100	30	100	30	100
BM 2	4	13	4	13	4	13
BM 3	7	85	7	85	7	88
BM 4	64	880	84	886	64	880
BM 5	640	8693	817	6851	640	7246

$$E_3 = \sum_{i=1}^{N}\sum_{j=1}^{M} V_{ij}(1 - V_{ij}) \,, \tag{19}$$

The total energy function of the NCNN-VT is given by the summation of the three parts E_1, E_2', and E_3:

$$E = \frac{W_1}{2}\sum_{i=1}^{N}(\sum_{q=1}^{M} V_{iq} - 1)^2 + \frac{W_2}{2}\sum_{i=1}^{N}\sum_{j=1}^{M}\sum_{\substack{p=1\\p\neq i}}^{N}\sum_{q=max(j-c_p+1,1)}^{min(j+c_i-1,M)} V_{ij}V_{pq}$$

$$+\frac{W_3}{2}\sum_{i=1}^{N}\sum_{j=1}^{M} V_{ij}(1 - V_{ij}) \,. \tag{20}$$

where W_1, W_2, and W_3 are weighting coefficients.

6.3 Result Discussions

Table 2 shows the results obtained by the NCNN-VT and a comparison with other previous methods. For the benchmark problems from BM 1 to BM 5, the NCNN-VT algorithm matches or improves the results of other existing algorithms. The results on the benchmark examples show that the NCNN-VT can find better or similar solution compared with the previous methods.

7 Future Challenges

We have reviewed neural-network-based techniques for solving NP-hard COPs, especially neural networks with chaotic neuro-dynamics. Two applications in telecommunication networks demonstrated that chaotic neural networks have effective search abilities compared to other methods. Despite the computational advantages of these methods, there still exist tremendous challenges from both methodology and applications points of view.

The "no free lunch" theorem proposed by Wolpert and Macready [44] showed that all algorithms that search for an extremum of a cost function perform exactly the same, when averaged over all possible cost functions. They claimed that if algorithm A outperforms algorithm B on some cost functions, then loosely speaking there must exist exactly as many other functions where B outperforms A. The no free lunch theorem also applies for the NCNN and the extensions (G-NCNN and NCNN-VT). Actually, the NCNN-based methods are facing difficities on solving problems besides combinatorial optimizations, for example, they are opt for solving combinatorial optimization problems, other than applications like non-linear or multi-dimensional function optimization. We may ask the question: how do chaotic neural networks compare to other computational intelligence methods, such as genetic algorithms and ant colonies, in solving other practical COPs?

Problem modeling can also cause difficulties when solving COPs using neural networks. Every COP, whether simple or hard, needs to be constructed into an energy function formulation before using the NCNN to solve it. And the form of this energy function is critical for neural-network-based methods to search for optimal solutions. Different formulations may lead to different solution quality and search time. It is common to see various formulations of energy funcations made by different researchers on the same problem, e.g. the TSP problem [4, 43, 38]. In order to improve solution quality, the formulation needs to be revised and upgraded from time to time. The state-of-the-art energy function formulation, if not difficult to formulate, is actually time-consuming to find.

Another tough problem when using neural-network-based methods is the selection of the parameters. The various parameters, including system parameters and weighting coefficients in the energy functions, are influential to solution quality and search efficiency. There are basic guidelines for parameter selections [35]; however, it can be challenging to find the optimal parameters, which in turn will lead to optimal solution quality and search efficiency.

For applications in the communications domain, comparisons performed in the research studies have usually been undertaken in simplified scenarios simulated in servers or desktop PCs. Except for some partial implementations on real hardware [12], algorithm testing using hardware is usually not undertaken. It will be necessary to place a greater emphasis on demonstrating advantages of computational intelligence methods in real communications hardware, in order to convince industrialists to adopt computational intelligence methods in telecommunications companies and equipment manufacturers.

References

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

[2] K. Aihara, T. Takabe, and M. Toyoda. Chaotic neural networks. *Physics Letters A*, 144:333–340, 1990.

[3] Mustafa K. Mehmet Ali and F. Kamoun. Neural networks for shortest path computation and routing in computer networks. *IEEE Trans. on Neural Networks*, 4:9, 1993.

[4] R.D. Brandt, Y. Wang, A.J. Laub, and S.K. Mitra. Alternative network for solving the travelling salesman problem and the list-matching problem. In *Proceedings IEEE International Joint Conference on Neural Networks*, volume 2, pages 333–340, 1988.

[5] L. Chen and K. Aihara. Chaotic simulated annealing by a neural network model with transient chaos. *Neural Networks*, 8:915–930, 1995.

[6] L. Chen and K. Aihara. Global searching ability of chaotic neural networks. *IEEE Trans. Circuits and Systems - I: Fundamental Theory and Applications*, 46(8):974–993, 1999.

[7] G.W. Davis. Sensitivity analysis in neural net solutions. *IEEE Trans. on Systems, Man and Cybernetics*, 19:1078–1082, 1989.

[8] D.E. Van den Bout and T.K. Miller. A traveling salesman objective function that works. In *Proceedings IEEE International Joint Conference on Neural Networks*, volume 2, pages 299–303, 1988.

[9] N. Funabiki and J. Kitamichi. A gradual neural network algorithm for broadcast scheduling problems in packet radio networks. *IEICE Trans. Fundamentals*, E82-A:815–824, 1999.

[10] N. Funabiki and S. Nishikawa. A binary hopfield neural-network approach for satellite broadcast scheduling problems. *IEEE Trans. on Neural Networks*, 8:441–445, 1997.

[11] N. Funabiki and S. Nishikawa. A gradual neural-network approach for frequency assignment in satellite communication systems. *IEEE Trans. Neural Networks*, 8:1359–1370, 1997.

[12] V. Catania G. Ficili S. Palazzo G. Ascia and D. Panno. A VLSI fuzzy expert system for real-time tra.c control in atm networks. *IEEE Transactions on Fuzzy Systems*, 5(1):20–31, 1997.

[13] S. Geman and D. Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Trans. Pattern Analysis and machine Intelligence*, 6:721–741, 1984.

[14] Y. Hayakawa, Marunoto A, and Y. Sawada. Effects of the chaotic noise on the performance of a neural network model for optimization problems. *Physical Review E*, 51:2693–2696, 2002.

[15] S. Hegde, J. Sweet, and W. Levy. Determination of parameters in a hopfield/tank computational network. In *Proceedings IEEE International Conference in Neural Networks*, volume 2, pages 291–298, 1988.

[16] J.J. Hopfield. Neurons with graded response have collective computational properties like those of two-state neurons. In *Proc. Natl. Acad. Sci. USA*, volume 81, pages 3088–3092, 1984.

[17] J.J. Hopfield and D. W. Tank. Neural computation of decisions in optimization problems. *Biological Cybernetics*, 52:141–152, 1985.

[18] M. Jeruchim. A survey of interference problems and applications to geostationary satellite networks. In *Proceedings IEEE*, pages 317–331, 1977.

[19] D.S. Johnson, C.H. Papadimitriou, and M. Yannakakis. Optimzation by simulated annealing: an experimental evalution, part 1, graph partitioning. *Operat. Res.*, 37:865–892, 1989.

[20] D.S. Johnson, C.H. Papadimitriou, and M. Yannakakis. Optimzation by simulated annealing: an experimental evalution, part 2, graph partitioning. *Operat. Res.*, 39:378–406, 1991.

[21] D. Jungnickel. *Graphs, Netwrks and Algorithms.* Springer-Verlag, Berlin, Germany, 1999.

[22] B. Kamgar-Parsi and B. Kamgar-Parsi. Dynamical stability and parameter selection in neural optimization. In *Proceedings IEEE International Joint Conference on Neural Networks*, volume 4, pages 566–571, 1992.

[23] S. Kirkpatrick, C.D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220:671–680, 1983.

[24] T. Kwok and K.A. Smith. A noisy self-organizing neural network with bifurcation dynamics for combinatorial optimization. *IEEE Trans. on Neural Networks*, 15:84–98, 2004.

[25] W.K. Lai and G.G. Coghill. Genetic breeding of control parameters for the hopfield/tank neural net. In *Proceedings IEEE International Joint Conference on Neural Networks*, volume 4, pages 618–632, 1992.

[26] O. Lazaro and D. Girma. A hopfield neural-network-based dynamic channel allocation with handoff channel reservation control. *IEEE Trans. on Vehicular Technology*, 49:1578–1687, 2000.

[27] R.S.T. Lee. A transient-chaotic autoassociative network (tcan) based on lee oscillators. *IEEE Trans. on Neural Networks*, 15:1228–1243, 2004.

[28] T. Mizuike and Y. Ito. Optimization of frequency assignment. *IEEE Trans. Communications*, 37:1031–1041, 1989.

[29] H. Nonaka and Y. Kobayashi. Sub-optimal solution screening in optimization by neural networks. In *Proceedings IEEE International Joint Conference on Neural Networks*, volume 4, pages 606–611, 1992.

[30] H. Nozawa. A neural network model as a globally coupled map and applications based on chaos. *Chaos*, 2(3):377–386, 1992.

[31] B. Pontano. Interference into angel-modulated systems carrying multichannel telephony signals. *IEEE Trans. Communications*, 21, 1973.

[32] C.R. Reeves. *Modern Heuristic Techniques for Combinatorial Problems.* Oxford, Blackwell, 1993.

[33] S. Salcedo-Sanz, C. Bouso no Calzón, and A.R. Figueiras-Vidal. A mixed neural-genetic algorithm for the broadcast scheduling problem. *IEEE Trans. on Wireless communications*, 2:277–283, 2003.

[34] S. Salcedo-Sanz, R. Santiago-Mozos, and C. Bouso no Calzón. A hybrid hopfield network-simulated annealing approach for frequency assignment in satellite communications systems. *IEEE Trans. Systems, Man, and Cybernetics-Part B: Cybernetics*, 34:1108–1116, 2004.

[35] H.X. Shi and L.P. Wang. Broadcast scheduling in wireless multihop networks using a neural-network-based hybrid algorithm. *Neural Networks*, 18:765C771, 2005.

[36] H. Tang, K.C. Tan, and Z. Yi. A columnar competitive model for solving combinatorial optimization problems. *IEEE Trans. on Neural Networks*, 15:1568–1573, 2004.

[37] I. Tokuda, K. Aihara, and T. Nagashima. Adaptive annealing for chaotic optimization. *Phys. Rev. E*, 58:5157–5160, 1998.

[38] A. Varma and Jayadeva. A novel digital neural network for the travelling salesman problem. In *Neural Information Processing, 2002. ICONIP '02*, volume 2, pages 1320–1324, 2002.

[39] G. Wang and N. Ansari. Optimal broadcast scheduling in packet radio networks using mean field annealing. *IEEE Journal on Selected Areas in Communications*, 15:250–260, 1997.

[40] L.P. Wang and F. Tian. Noisy chaotic neural networks for solving combinatorial optimization problems. In *Proc. International Joint Conference on Neural Networks*, volume 4, pages 37–40, 2000.

[41] L.P. Wang and K. Smith. On chaotic simulated annealing. *IEEE Transactions on Neural Networks*, 9:716–718, 1998.

[42] R.L. Wang, Z. Tang, and Q.P. Cao. A hopfield network learning method for bipartite subgraph problem. *IEEE Trans. on Neural Networks*, 15:1458–1465, 2004.

[43] G.V. Wilson and G.S. Pawley. On the stalibility of the travelling salesman problem algorithm of hopfield and tank. *Biol. Cybern.*, 58:63–70, 1988.

[44] David H. Wolpert and William G. Macready. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1:67C82, 1997.

[45] M. Yamaguti, editor. *Solution of the optimization problem using the neural network model as a globally coupled map*, 1994.

[46] M. Yamguti, editor. *Transient chaotic neural networks and chaotic simulated annealing.* Amsterdam: Elsevier Science Publishers, 1994.

[47] L. Zheng, K. Wang, and K. Tian. An approach to improve wang-smith chaotic simulated annealing. *International Journal of Neural Systems*, 12:363–368, 2002.