Channel Assignment using Chaotic Simulated Annealing Enhanced Hopfield Neural Network

Amir massoud Farahmand Mohammad javad Yazdanpanah

Abstract—**Channel assignment problem in cellular communication is a difficult combinatorial optimization problem. There is no exact polynomial-time solution for it and searching the whole solution space is infeasible for large problems. By defining the problem's cost function as the energy function of a chaotic Hopfield neural network, we devise a framework for finding competitive suboptimal or even optimal solutions for combinatorial optimization problem in general, and channel assignment problem in particular. In our architecture, we inject chaotic noise in order to help the network escape from local minima of the energy function while we enforce problem constraints by external inputs of neurons. Experimental results show the superiority of our method to other methods.**

I. Introduction

Combinatorial Optimization Problems (COP) arise in many application domains of science and technology like job scheduling, VLSI connection optimization, and channel assignment in cellular communication. In this paper, we introduce a Chaotic Simulated Annealing-enhanced Hopfield Neural Network (CSA-HNN) meta-heuristic to effectively solve the channel assignment problem. Channel assignment problem (CAP) is a sample of COP that arises in cellular telecommunication. The region we want to cover by our telecommunication system is divided to many smaller geographical subregions called cell. We have a limited number of frequency channels that must be assigned to cells. Nearby cells have interference with each other, so we need to assign distant frequencies (in frequency domain) to nearby cells. The channel assignment problem is assigning our channels to cells in order to reduce the interference.

Combinatorial optimization problem is a 0-1 optimization problem than can be considered as the generalization of graph coloring problem, which is NP-Hard. It is evident that searching the entire solution space is not possible for big problems. Therefore, having a bunch of heuristics and meta-heuristics for finding appropriate sub-optimal solutions seems inevitable [1]-[3]. These heuristic methods may not find the optimal solution in every case, but they can find a good solution most times. Depending on the situation, a good heuristic might be considered as the one that results in solution closer to the optimal one, or the one that can find an acceptable solution in a reasonable time, or whole range of methods between these two. In this paper, we propose a chaotic simulated annealing-enhanced Hopfield Neural Network (HNN) that not only can find very good and even optimal solutions, but also can find them very fast. This latter issue is mainly due to the possibility of massively parallel hardware implementation of the Hopfield neural network base of CSA-HNN.

The use of HNN for solving COP emerged after demonstration of its ability in solving traveling salesman problem (TSP) by Hopfield and Tank [4]. The technique is based on the energy minimizing property of HNN. Unfortunately, it was shown that the simple HNN often yields infeasible solutions for TSP [5]. Although there were some efforts to cure this problem (e.g. changing energy function or confine solutions to constraint plane), results were not comparable to ones obtained from traditional techniques. The main reason of this inefficiency is the structure of energy function in HNN, which has many local minima in which the network get stuck in one of them due to its strictly energy reducing behavior [2].

To overcome this difficulty we can modify the behavior of the network in order to let it get out of local minima. For instance, in [1] and [2], they use the idea of simulated annealing and permit HNN to occasionally change its state in ascending direction of Lyapunov energy function while reducing this probability as network iterates. As another approach to satisfy constraints and prohibit infeasible solutions, [3] and [6] change the external input of each neuron and penalize the neuron proportional to the amount of deviation from the constraint in which the neuron participates. By doing so, they solve channel assignment problem. We benefit from this idea in our architecture.

There is a new trend in using chaotic neural networks in order to solve COP [7]-[11]. Adding chaotic noise or similarly letting the network to behave chaotically seems to enhance the ability of HNN in searching the global minimum of its Lyapunov energy function. In this paper, we use a modified version of [11] to solve channel assignment problem.

The organization of this paper is as follows: The COP is formulated in Section II. Afterward, the relation between the COP and the HNN and the formulation of HNN with

A. M. Farahmand was with the Electrical and Computer Engineering Department, Control and Intelligent Processing Center of Excellence, University of Tehran, Tehran, Iran. He is now with the Department of Computing Science, University of Alberta, Canada (e-mail: amir@cs.ualberta.ca).

M. J. Yazdanpanah is with the Electrical and Computer Engineering Department, Control and Intelligent Processing Center of Excellence, University of Tehran, Tehran, Iran (e-mails: yazdan@ut.ac.ir).

decaying chaotic noise is presented in Section III. The channel assignment problem in described in Section IV and the experimental results that show the superiority of our method will be presented in Section V.

I. Combinatorial optimization problem

In this section, a brief formulation of a class of COP is provided. Although in this paper we provide experimental results only for CAP, the method should work in similar problems too. Quadratic form of 0-1 COP is defined as follows [2]:

$$\begin{aligned} &\min_X F(X) = \\ &\min_X \sum_{i=1}^{N}\sum_{j=1}^{M} X_{i,j} \sum_{p=1}^{N}\sum_{q=1}^{M} Q(i,j,p,q) X_{p,q} + \sum_{i=1}^{N}\sum_{j=1}^{M} C(i,j) X_{i,j} \\ &\text{subject to} \quad \begin{array}{l} \sum_j X_{i,j} = D_i \quad \forall i = 1,\ldots,N \\ \sum_i X_{i,j} = 1 \quad \forall i = 1,\ldots,M \end{array} \\ &X_{i,j} \in \{0,1\} \end{aligned} \tag{1}$$

where X is the solution matrix, $F(X)$ is a quadratic cost function, and Q, C, and D are constants that define the problem. The first term (quadratic) of $F(X)$ indicates the cost of being "on" (i.e. 1) of two elements of the solution together and the second one (linear) indicates being "on" state of a single one. The first constraint is called *demand* or *transportation* constraint and the second one is named *assignment* constraint [2]. These constrains are commonly encountered in many combinatorial optimization problems. For example, the TSP's constraints are $D_k = 1$.

II. Hopfield Neural Network with Decaying Chaotic Noise

Hopfield neural network comprises a fully interconnected system of *n* neurons. Neuron i has internal state u_i and external (to the network) input signal I_i. The dynamical equation of the system is described by the following equations [12]:

$$\frac{du_i}{dt} = -\frac{u_i}{\tau} + \sum_j w_{ij} x_j + I_i \tag{2}$$

$$x_i(t) = f\big(u_i(t)\big) \; . \tag{3}$$

A common activation function is sigmoidal function:

$$x(t) = f\big(u(t)\big) = \frac{1}{1 + e^{-u(t)/\varepsilon}} \, . \tag{4}$$

Assuming that $\mathbf{w}$ is symmetric and $f^{-1}(u)$ is continuous and monotonously increasing function of u, it can be shown that the system described by (1) and (2) converges to a local minimum of the following Lyapunov energy function:

$$\begin{aligned} E_{Hop}(t) = &-\frac{1}{2}\sum_i\sum_j w_{ij} x_i(t) x_j(t) - \\ &\sum_i I_i x_i(t) + \frac{1}{\tau}\sum_i \int_0^{x_i(t)} f^{-1}(\upsilon)\, d\upsilon \end{aligned} \, . \tag{5}$$

Without going further, we know that the stable points of the very high-gain, continuous deterministic Hopfield model corresponds to the stable points of the discrete stochastic Hopfield model with the following Lyapunov energy function [12]:

$$E_{Hop}(t) = -\frac{1}{2}\sum_i\sum_j w_{ij} x_i(t) x_j(t) - \sum_i I_i x_i(t) \, . \tag{6}$$

Comparing (6) with the cost function of COP (1), one can easily see the similar quadratic and linear terms in both of them. Therefore, whenever there is a COP in the form of (1), one can construct a HNN that has the same energy function (6) as the cost function $F(X)$. Benefiting from the fact that the dynamics of HNN tries to minimize (6), we can use HNN to solve (1). However, before that, we need a method to incorporate constraints of problem (1) into the dynamics of the HNN. Including the constraints in the cost function as a penalty term ([3] and [11]) or enforcing solution vector to lie in the constraints plane ([1] and [2]) would do so.

The chaotic Hopfield neural network used in this paper is mainly in the form of [11], which is defined as

$$u_i(t+1) = \alpha\left(\sum_j w_{ij} x_j + I_i\right) + \gamma \eta_i(t) \tag{7}$$

$$x_i(t) = f\big(u_i(t)\big) = \frac{1}{1 + e^{-u_i(t)/\varepsilon}} \tag{8}$$

where (i=1,2,…,n) is the neuron index, $u_i(t)$, $x_i(t)$ and I_i are the internal state, output and input bias of neuron i. α and γ are positive scaling parameters for input and chaotic noise and ε defines the gain of the output (activation) function. $\eta_i(t)$ is an external chaotic noise for neuron i.

We may use different mechanisms for generating chaotic noise with different properties, but here, a modification of logistic map is used

$$\eta_i(t) = z_i(t) - h \tag{9}$$

$$z_i(t+1) = a(t) z_i(t)\left(1 - z_i(t)\right) \tag{10}$$

$$a(t+1) = (1-\beta) a(t) + \beta a_0 \tag{11}$$

$$h = z^* = 1 - \frac{1}{a_0} \tag{12}$$

where $0 < \beta < 1$. Initializing a to a large value makes the process chaotic and decreasing it will lead to a stable solution. The idea of chaotic simulated annealing (CSA) is starting from a large a (i.e. $a(1) >> 1$) and then decreasing it through time. Large values of a force the HNN to behave chaotically; therefore, it explores the solution space with a considerable randomness. Gradually decreasing a change the behavior of the injected noise from chaotic to stable fixed-point (i.e. constant) through reversed period-doubling bifurcations. After becoming stable, the noise does not affect the network at all and the system remains in the same local basin of attraction afterwards. If the previous exploration was sufficient, we may hope that we would find the right basis of attraction through the chaotic to stable dynamics transition.

Before formulating channel assignment problem, it is necessary to point out that this chaotic model of HNN is not the only way of making a chaotic simulated annealing (CSA). In this paper, we used external chaotic noise, but there are other methods, which have internal chaotic dynamics ([7] and [8]), namely decaying self-coupling of [7] and decaying time step of [8]. Despite these differences, in [9] they introduced a unified framework for representing the energy function of different methods of making a chaotic HNN. The modified energy function is defined as

$$E(t) = E_{Hop}\left(\mathbf{x}, \mathbf{w}, \mathbf{I}\right) + H\left(\mathbf{x}, \mathbf{w}, \mathbf{I}\right) \tag{13}$$

in which H is a new term that modifies the energy landscape.

In different methods of CSA, the way the energy function is modified is different and in the one we use here, it can be expressed as

$$H = -\sum_i \eta_i x_i \tag{14}$$

in that η_i is defined as [13].

III. Channel Assignment Problem

These days, there are increasing demands for cellular mobile communication systems. Like any other engineering problem, optimizing our resources in this problem is crucial. In cellular mobile communication, the frequency channels are our resources and it is a need to use them wisely.

The cellular mobile communication problem is as follows ([1] and [3]): In cellular communication, the geographical area is divided into many smaller regions, which are called cells. Each cell has its own caller traffic and according to that, a demand for the cell is defined. Some cells' demand is high (e.g. business centers) and some cells' demand is low (e.g. low-population regions in the area). In cellular communication, we have a limited number of frequency channels. Each caller occupies each channel when s/he uses the network. The main problem is interference between these frequency channels. The closer frequencies of two callers are, the more interference between those channels exists. This interference depends on the distance between two callers. If they are far from each other (i.e. two cells are far in the network), the interference is low and vice versa. In order to use the frequency channels efficiently, one needs to assign close frequencies to faraway cells.

There are three sources of interference and thus three constraints in this problem: interference between two callers in the same cell (*co-site interference*), interference between two callers using the same channel within some spatial range (*co-channel interference*), and interference between two callers in adjacent cells using adjacent channels (*adjacent channel interference*). The amount and severity of these interferences can be calculated considering the topological and physical nature of the region. The demands can also be estimated considering the population and other related factors in each cell. The objective is assigning channels for each cell in order to minimize the interference while keeping demands satisfied. It may happen that one has enough frequency channels, but due to incorrect cell/channel assignment, s/he cannot use all of her/his channels, so the efficiently of the network would be degraded.

The channel assignment problem can be divided into static and dynamic cases. In static channel assignment (SCA), we assign channels only once in the network design phase, but in dynamic channel assignment (DCA) we change the assignment as the traffic's demand of the network changes. Only the solution of SCA is considered in this paper. However, if we can implement our method efficiently and have access to the latest information about the demand of each cell, we can re-run the SCA continually to reassign channels during the network run.

Suppose there are N cells and M channels. The demand of the i^{th} cell is $D_i (i = 1, ..., N)$. Define compatibility matrix, $\mathbf{C}_{N \times N}$ as the matrix that shows the severity of interference between cells. C_{ii} is the minimum required frequency distance between two assigned channels in cell i and C_{ij} indicates the required distance between cell i and j. We can define the COP as follow [1]

$$\min \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{M} X_{i,j}\sum_{p=1}^{N}\sum_{q=1}^{M} C_{ip}X_{p,q}$$

$$\text{subject to: } \sum_{j} X_{i,j} = D_k \quad \forall k = 1,...,N. \tag{15}$$

$$X_{i,j} \in \{0,1\}$$

This minimization problem is the same as the COP we defined before - except that it has not any linear term (the other constraint is included in C matrix). In order to include constraints, we use penalty term method as [3]. After comparing this new problem with Lyapunov energy function of HNN, we find the following weight matrix and inputs for the network [3]:

$$W_{ijpq} = -\delta_{ip}\alpha_{jq}(d_{\text{csc}}) - |1-\delta_{ip}|\alpha_{jq}(C_{ip}) - \delta_{ip}|1-\delta_{jq}| \tag{16}$$

where δ_{ij} is the Kronecker delta function, $d_{\text{csc}} = C_{ii}$ (which is considered to be constant among all cells) and

$$\alpha_{ij}(x) = \begin{cases} 1, & \text{if } |\text{i - j}| < x \\ 0, & \text{othewise} \end{cases} \tag{17}$$

and $W_{ijij} = 0$ (self-feedback is not allowed). The external input I_i is defined as

$$I_i = (D_i - 1). \tag{18}$$

In order to force the network to converge to feasible solutions, following the idea of Kim ([3]) we define another external input that enforces demand constraint to be satisfied as

$$I_{E_{ij}} = \left(D_i - \sum_{k=1}^{M} X_{ik} \right). \tag{19}$$

In addition, we inject our chaotic noise (9) to increase the chance of escaping from local minima. Now, we have

$$u_{ij}(t+1) = \alpha\left(\sum_{p}\sum_{q} w_{ijpq}x_{pq} + I_i + I_{E_{ij}}\right) + \gamma\eta_{ij}(t) \tag{20}$$

$$x_{ij}(t) = f(u_{ij}(t)) = \frac{1}{1+e^{-u_{ij}(t)/\varepsilon}} \tag{21}$$

$$\eta_{ij}(t) = z_{ij}(t) - h \tag{22}$$

$$z_{ij}(t+1) = a(t)z_{ij}(t)(1 - z_{ij}(t)). \tag{23}$$

There are some remarkable points. First is the modification we have made in CSA. In [11], they use monotonically decreasing $a(t)$ so that the network has chaotic behavior only in the beginning (although setting β a small value, increase the number of iterations with the chaotic behavior). Assigning β not so small (around 0.05) may deteriorate the ability of network to converge to global minima and assigning it a very small number will increase the number of iterations needed for convergence. Therefore, we have decided to let β be not so small (so noise would change fast from chaotic to stable behavior), but reset $a(t)$ to its initial value ($a(0)$) whenever the value of cost function which we calculate in each iteration does not change anymore. Thus, the network's dynamical behavior might change from chaotic to stable several times during the run. The second note is that the network's output, by construction, is not a binary value, but a continuous value between 0 and 1. Remembering that a solution of COP should be 0 or 1, we need to devise a method to infer 0/1 solution from the continuous-value outputs. Setting ε in (21) a small value makes sigmoid behave like a threshold function (so, it outputs close to zero or one) but it was not as satisfactory as our approach. Our method is simply selecting the D_i greatest value of i^{th} row of X as follows

$$x_{ik}^{d}(t) = \begin{matrix} 1 & \text{if } \mathrm{x_{ik}}(t) \geq \mathrm{D_i}^{\text{th}} \text{ value in } \mathrm{i}^{\text{th}} \text{ row} \\ 0 & \text{otherwise} \end{matrix} \tag{24}$$

After convergence, the method is the same as others (like relying on networks high-gain feature) but it works better in early iterations. The third note is that we update HNN asynchronously and each iteration in this paper means the whole updating cycle.

IV. EXPERIMENTS

In this section, we apply our method to two different SCA dataset. One of them is small artificial one (adopted from [1]) and we call it as *EX* dataset (Table I). These two problems have interference-free solutions while satisfying constraints. The other problem is a real-world telecommunication problem that has larger number of cells and channels than the previous problem. We call it as *KUNZ* dataset. *KUNZ* dataset is derived from an actual 24x21km area around Helsinki, Finland as used by Kunz [13]. In its original form, the region is divided into 25 cells while 73 channels are available. There are some modified versions, which use only a subset of the whole cells and channels. They are important because there is no interference-free solution for them, thus comparing different methods on those cases can show the ability of different methods to find the best solution. The size of *KUNZ* problems is defined in

TABLE I
PROBLEM DESCRIPTION FOR *EX DATASET*

EX 1	$N=4$	$M=11$	$D=\begin{pmatrix}1\\1\\1\\3\end{pmatrix}$	$C=\begin{pmatrix}5&4&0&0\\4&5&0&1\\0&0&5&2\\0&1&2&5\end{pmatrix}$
EX 2	$N=5$	$M=17$	$D=\begin{pmatrix}2\\2\\2\\4\\3\end{pmatrix}$	$C=\begin{pmatrix}5&4&0&0&1\\4&5&0&1&0\\0&0&5&2&1\\0&1&2&5&2\\1&0&1&2&5\end{pmatrix}$

TABLE II
PROBLEM DESCRIPTION FOR KUNZ DATASET

Problem	N	M	C	**D**
KUNZ1	10	30	$[C_3]_{10}$	$[D_3]_{10}$
KUNZ2	15	44	$[C_3]_{15}$	$[D_3]_{15}$
KUNZ3	20	60	$[C_3]_{20}$	$[D_3]_{20}$
KUNZ4	25	73	$[C_3]_{25}$	$[D_3]_{25}$

Table II. For the exact definition of C and D see [1].

For our simulations, we used the following parameters:

$$\begin{aligned}&\alpha=0.015\\&\varepsilon=0.004\\&a(0)=3.9\\&\beta=0.05\\&\gamma=0.5\end{aligned}\qquad(25)$$

and the simulation lasted at most 500 steps for all problems. For gaining more insight into the effect of CSA, Fig. 1, 2, and 3 show a typical time evolution of energy function for *EX2* problem using CSA (CSA-HNN), Hill-Climbing method (HC-HNN) of [3], and simple HNN, respectively. It is seen in Fig. 1 that CSA-HNN can find the best solution after exploring the state space guided by chaotic noise. The noise decreases gradually and the network converges to a stable fixed-point. In Fig. 2, the result of HC-HNN for this problem is depicted. It escapes a few times from local minima, but it cannot find the optimum value. The simple HNN decreases the energy function monotonically, so it cannot escape any local minimum. The same behavior is observed for other examples too. An interference-free solution of *EX2* is depicted in Fig. 4. Based on the parameters, the speed of convergence can be low or high. We use the same set of parameters for all problems of this paper. It is expected that this specific parameters are not the best possible set for this *EX2* and faster convergence rate would be possible (as this problem is much smaller than KUNZ, so it is probable that faster deteriorating noise would work too).

Results of the simulation and comparison with other method for all problems are listed in Table III. GAMS/MINOS, Steepest Descent (SD) heuristics, Simulated Annealing (SA), Simple Hopfield network (HN) and hill-climbing Hopfield network (HCHN) are taken from [1] and TCNN is taken from [10]. Note that HCNN is a Hill-Climbing HNN that is different from Kim's ([3]) and is enhanced by using simulated annealing-like behavior. TCNN is a chaotic HNN, which uses self-coupling decaying method of [7] for being chaotic. Placing 'N/A' instead of a value may mean two things: there is no data available (TCNN case) or we have not yet run that experiment enough times to have an estimation of the average cost (*KUNZ2* and *KUNZ4* with CSA-HNN*)*. When average values are available, we did the experiment for at least ten times except in *KUNZ3* that we did it for five times. However, as in the *KUNZ3* case the average cost is equal to the minimum cost, we infer that our method will find the best solution in every trial with high probability. It is seen that the performance of CSA-HNN is the best for all problems. Also its average performance (for the cases that this information is available) is equal to the best performance. No other method shows this property.

We have done some robustness tests on parameters of the method and changed γ and β in order to investigate their effect on feasibility of solutions (whether it can find a solution with global minimum cost or not) and convergence speed. Our preliminary investigation shows that the performance of the system is somehow robust to its parameters' variations. More investigation is needed for a precise statement.

The computation time of these experiments ranges from a few minutes to a few hours. For instance, *KUNZ4* takes about seven hours in our MatLab implementation. We did not benefit from the capability of Matlab in performing matrix calculations. Therefore, it is expected that if we had implemented our method in a language like C, we would have gotten results much faster. However, as we argue in the conclusion, this is not a restricting issue.

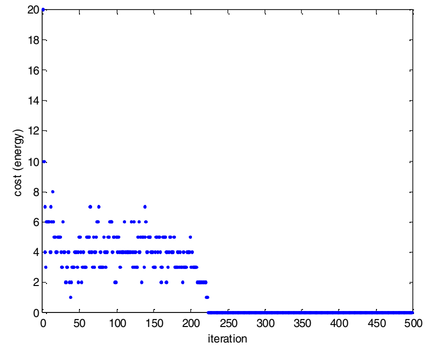

Fig 1. Energy (cost) function changes for CSA-HNN through iterations of the network. It finds the optimal solution of the problem.

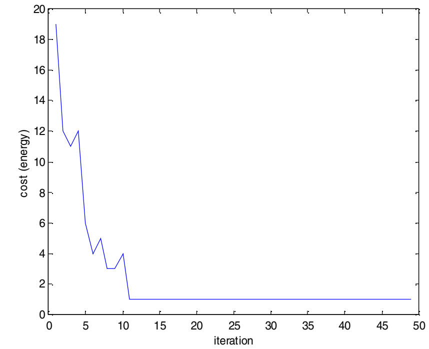

Fig 2. Energy (cost) function changes for HC-HNN through iterations of the network. It cannot find the optimal solution of the problem.

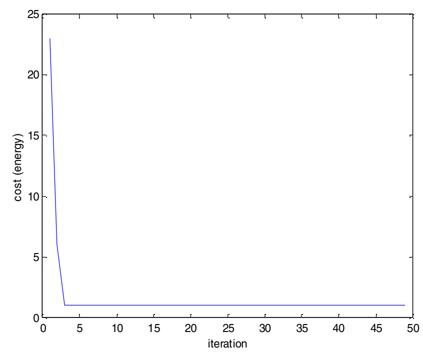

Fig 3. Energy (cost) function changes for simple HNN through iterations of the network. It cannot find the optimal solution of the problem.

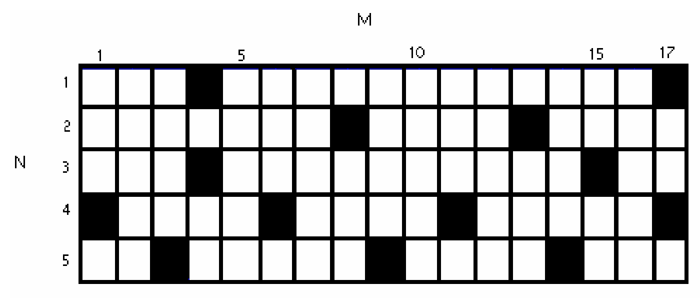

Fig 4. An interference-free assignment for EX2 problem.

V. Conclusion

In this paper, we applied chaotic simulated annealing-enhanced Hopfield neural network in order to solve a difficult COP. The results show the effectiveness of the framework in solving COP. Adding chaotic noise enhances the capability of HNN to escape from local minima and thus increases the chance of finding the global minimum. One may argue that this method is slower comparing to simple HNN or other heuristics. The important issue is that the speed of convergence is not a big issue for many problems like channel assignment problem if the computation time is not exponential in the number of states (which is of course not in this case). One reason is that no one needs to find a solution to SCA in real-time as this optimization is done in network design phase in which a few minutes or even a few hours are not important. The other reason is the special massively parallel structure of HNN that lets us implement the method in hardware. This way, we can even reach near real-time performance, as a few hundreds of iterations are not a big deal for a parallel hardware.

Our future research is focused on four issues:

1) Investigations on the effect of parameters' variations in the problem solving ability (like what [9] did) must be made. Our early simulations show that the performance is rather robust to parameters' changes.

2) Comparing this method of making chaotic network with other ones (specifically with [7] and [8]).

3) Use a progress estimator for changing the amount of chaotic noise injected in the network depending on the quality of the solution instead of using a fixed schedule.

4) Applying this methodology to other combinatorial optimization problems.

References

[1] K. Smith and M. Palaniswami, "Static and dynamic channel assignment using neural networks," *IEEE J. Selected Areas in Comm.*, vol. 15, no. 2, 1997, pp. 238-249.

[2] K. Smith, M. Palaniswami, and M. Krishnamoorthy, "Neural techniques for combinatorial optimization with applications," *IEEE Trans. Neural Network*, vol. 9, no. 6, 1998, pp. 1301-1381.

[3] J. S. Kim, S. H. Park, P. W. Dowd, and N. M. Nasrabadi, "Cellular radio channel assignment using a modified hopfield network," *IEEE Trans. Veh. Technol.*, vol. 46, no. 4, 1997, pp. 957-967.

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

[5] G. V. Wilson and G. S. Pawley, "On the stability of the tsp algorithm of hopfield and tank," *Biol. Cybern.*, vol. 58, 1988, pp. 63–70.

[6] M. J. Yazdanpanah, E. Madanian, and A. M. Farahmand, "Channel assignment in cellular communications using a new modification on hopfield network," accepted for publication in Iranian Journal of Science and Technology, Transaction B: Engineering, 2005.

TABLE III

TEST PROBLEMS FOR GAMS/MINOS-5, STEEPEST DESCENT (SD), SIMULATED ANNEALING (SA), SIMPLE HOPFIELD NETWORK (HN), HILL-CLIMBING HOPFIELD NETWORK (HCHN), TRANSIENT CHAOTIC NEURAL NETWORK (TCNN), AND CHAOTIC SIMULATED ANNEALING – HOPFIELD NEURAL NETWORK (CSA-HNN) (THIS WORK).

	GAMS	**SD**		**SA**		**HN**		**HCHN**		**TCNN**		**CSA-HNN**	
Problem	**Min**	**Av.**	**Min**	**Av.**	**Min.**	**Av.**	**Min**	**Av.**	**Min**	**Av.**	**Min**	**Av.**	**Min**
EX1	2	0.6	0	0.0	0	0.2	0	0.0	0	N/A	N/A	0.0	0
EX2	3	1.1	0	0.1	0	1.8	0	0.8	0	N/A	N/A	0.0	0
KUNZ1	28	24.4	22	21.6	21	22.1	21	21.1	20	20.6	20	20.0	20
KUNZ2	39	38.1	36	33.2	32	32.8	32	31.5	30	31.2	30	N/A	30
KUNZ3	13	17.9	15	13.9	13	13.2	13	13.0	13	13.0	13	13.0	13
KUNZ4	7	5.5	3	1.8	1	0.4	0	0.1	0	0.5	0	N/A	0

[7] L. Chen and K. Aihara, "Chaotic simulated annealing by a neural network model with transient chaos," *Neural Networks*, 8 (6), 1995, pp. 915-930.

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

[9] T. Kwok and K. A. Smith, "Experimental analysis of chaotic neural network models for combinatorial optimization under a unifying framework," *Neural Network* (13), 2000, pp. 731-744.

[10] Y. Zhang, Z. He, Ch. Wei, and L. Yang, "Parametric controlled transient chaotic neural network for the channel assignment problem," *IEEE Asia-Pacific Conf. on Circuits and System*, 2000, pp. 331-334.

[11] Y. He, "Chaotic simulated annealing with decaying chaotic noise," *IEEE Trans. Neural Networks*, vol. 13, no. 6, 2002, pp. 1526-1531.

[12] S. Haykin, *Neural Networks: A Comprehensive Foundation*, 2nd edition, Prentice Hall International, 1999, pp. 680-696.

[13] D. Kunz, "Channel assignment for cellular radio using neural networks," *IEEE Trans. Veh. Technol.,* vol. 40, no. 1, 1991, pp. 188–193.