Minimizing Interference in Satellite Communications Using Chaotic Neural Networks

Wen Liu, Haixiang Shi and Lipo Wang
College of Information Engineering, Xiangtan University,
Xiangtan, Hunan, China.
School of Electrical and Electronic Engineering,
Nanyang Technological University, Block S1,
50 Nanyang Avenue, Singapore 639798.
{liuw0004, pg02782641, elpwang}@ntu.edu.sg

Abstract

We solve the frequency assignment problem (FAP) in satellite communications with transiently chaotic neural networks (TCNN). The objective of this optimization problem is to minimize cochannel interference between two satellite systems by rearranging the frequency assignments. For an N-carrier-M-segment FAP problem, the TCNN consists of $N \times M$ neurons. The performance of the TCNN is demonstrated through solving a set of benchmark problems, where the TCNN finds comparative if not better solutions compared to the existing algorithms.

1 Introduction

Wireless communication has received a lot of attention these years due to its various applications including mobile systems, broadcasting, and satellite communications. One important research direction in wireless communication is interference minimization, so as to guarantee a desired level of quality of service. The frequency rearrangement is an effective complement alongside with the technique to reduce the interference itself. Among diverse formulations and objectives of frequency assignment problems (FAP) [9, 8, 15], we focus on frequency assignments in satellite communications in this paper.

Mizuike and Ito [7] proposed segmentation of frequency band and presented a method based on the branch-and-bound approach for the FAP in satellite communications. The carrier is uniformly divided to unit segments with arbitrary width. The application of their method showed that the method was effective in the interference minimization. Funabiki and Nishikawa [4] proposed a gradual neural network (GNN), where the cost optimization is achieved by a gradual expansion scheme and a binary neural network is in charge of constraints. Salcedo-Sanz *et al.* combined the Hopfield network with simulated annealing (HopSA) [14] and the genetic algorithm (NG) [13] to solve the FAP in satellite communications. As a kind of hybrid algorithms, there is an increase in the computational cost of the HopSA and the NG compared with the GNN [4, 14].

Chaotic neural networks were presented by Nozawa [10, 11] through adding negative self-feedback connections into Hopfield networks. The simulation on several combinatorial optimization problems showed that chaotic search is efficient in approaching the global optimum or sub-optima. Chen and Aihara [2] proposed a transiently chaotic neural network (TCNN) by introducing a decaying negative self-feedback. The TCNN, which is also known as chaotic simulated annealing (CSA) [16], is a powerful tool for combinatorial optimization problems [3, 17, 5]. In this paper, we solve the FAP in satellite communications through the TCNN, and simulation results show that the performance of the TCNN is comparative with existing heuristics.

This paper is organized as follows. We review the TCNN in Section 2. The formulation of the TCNN on the FAP is described in Section 3. Parameters settings and simulation results are presented in Section 4. Finally, we conclude the contribution of this paper in Section 5.

2 Transiently chaotic neural networks

The TCNN [2] model is described as follows:

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

$$y_{ij}(t+1) = ky_{ij}(t) + \alpha[\sum_{p=1,p\neq i}^{N} \sum_{q=1,q\neq j}^{M} w_{ijpq}x_{pq}(t)$$

Third International Conference on Natural Computation (ICNC 2007)
0-7695-2875-9/07 $25.00 © 2007 IEEE

$$+I_{ij}] - z(t)\left[x_{ij}(t) - I_0\right] \quad (2)$$

$$z(t+1) = (1-\beta)z(t) \quad (3)$$

where the variables are

y_{ij} = internal state of neuron ij;
x_{ij} = output of neuron ij;
ε = the steepness parameter of the transfer function ($\varepsilon \geq 0$);
k = damping factor of the nerve membrane ($0 \leq k \leq 1$);
α = the positive scaling parameter for inputs;
w_{ijpq} = the connection weight from neuron ij to neuron pq;
I_{ij} = input bias of neuron ij;
$z(t)$ = self-feedback neuronal connection weight ($z(t) \geq 0$);
I_0 = positive parameter;
β = damping factor for the time-dependent neuronal self coupling ($0 \leq \beta \leq 1$).

w_{ijpq} is confined to the following conditions [6]:

$$\sum_{p=1,p\neq i}^{N}\sum_{q=1,q\neq j}^{M} w_{ijpq}x_{pq}(t) + I_{ij} = -\partial E/\partial x_{ij} \quad (4)$$

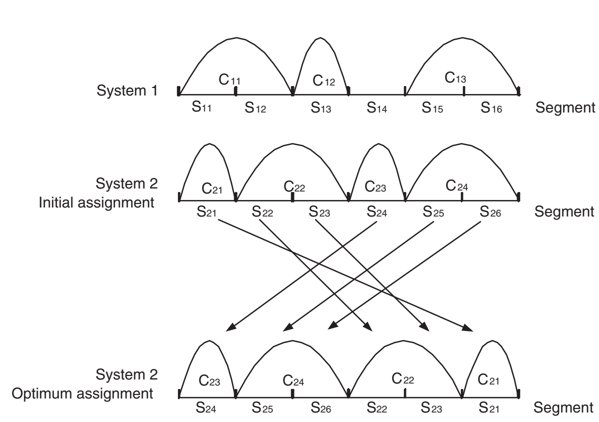
where E denotes the energy function, which is designed to have the minimum value at the optimal solution of the combinatorial optimization problem. Connection weights between neurons (w_{ijpq}) are derived by Equation (4) so that the energy function will decrease monotonously as neurons update after the self-feedback interaction vanishes ($z = 0$).

3 Problem formulation

We use the FAP formulation given by [7] and used in [4, 14, 13]. As indicated in [7], the objective of the FAP includes two parts, i.e., minimization of the largest interference after reassignment and minimization of the total accumulated interference between systems. Fig.1 shows the segmentation method for a 4-carrier-6-segment FAP and an example of interference matrix $E^{(I)} = (e_{ij})$.

We continue using the neural network formulation given by Funabiki and Nishikawa [4]. An N-carrier-M-segment FAP between two systems is formulated on an $N \times M$ neural network. If the neuron output $x_{ij} = 1$, then carrier i is assigned to segment j, and no assignments are made if $x_{ij} = 0$.

Figure 1. Segmentation and interference matrix [4]. C_{ij} denotes the j-th carrier in system i. S_{ij} stands for the j-th segment in system i. (a) Segmentation of the two systems and the optimum assignment corresponds to the followed interference matrix. (b) the interference matrix $E^{(I)}$ for a 4-carrier-6-segment system. * denotes infinity.



(a)

		C11		C12		C14	
		S11	S12	S13	S14	S15	S16
C21	S21	20	20	40	0	25	25
C22	S22	50	10	30	0	55	*
	S23	*	50	30	0	15	55
C23	S24	30	30	45	0	35	35
C24	S25	45	5	25	0	50	*
	S26	*	45	25	0	10	50

(b)

The energy function for the TCNN of the FAP is defined as [7, 4, 12]:

$$E = \frac{W_1}{2}\sum_{i=1}^{N}(\sum_{j=1}^{M} x_{ij} - 1)^2 + \frac{W_2}{2}\sum_{i=1}^{N}\sum_{j=1}^{M}\sum_{\substack{p=1\\p\neq i}}^{N}\sum_{q=max(j-c_p+1,1)}^{min(j+c_i-1,M)} x_{ij}x_{pq} + \frac{W_3}{2}\sum_{i=1}^{N}\sum_{j=1}^{M} x_{ij}(1-x_{ij}) + \frac{W_4}{2}\sum_{i=1}^{N}\sum_{j=1}^{M} d_{ij}x_{ij} \quad (5)$$

where $W_i, i = 1, \ldots, 4$, are weighting coefficients. The W_1 and W_2 terms are designed to guarantee that the solu-

Third International Conference on Natural Computation (ICNC 2007)
0-7695-2875-9/07 $25.00 © 2007 IEEE

Table 1. Specifications of the FAP instances used in the simulation.

#	Number of carriers N	Number of segments M	Range of carrier length	Range of interference
1	4	6	1 - 2	5 - 55
2	4	6	1 - 2	1 - 9
3	10	32	1 - 8	1 - 10
4	10	32	1 - 8	1 - 100
5	10	32	1 - 8	1 - 1000
6	18	60	1 - 8	1 - 50
7	30	100	1 - 8	1 - 100
8	50	200	1 - 8	1 - 1000

tion is a feasible one. The W_3 term is used to force neuron outputs approach 0 or 1 [12]. The W_4 term optimizes the interference after the frequency rearrangement. $max(x, y)$ returns the larger one between (x, y) and $min(x, y)$ finds the smaller value between (x, y). c_i denotes the carrier length of carrier i. Cost matrix $D = (d_{ij}, i = 1, \ldots, N; j = 1, \ldots, M)$ is computed from the interference matrix $E^{(I)}$ [4]. d_{ij} is the element on row i column j of the cost matrix D.

The neuron output is continuous between 0 and 1. We convert the continuous output x_{ij} to discrete neuron output x^b_{ij} as follows [1]:

$$x^b_{ij} = \begin{cases} 1, & \text{if} x_{ij} > \frac{1}{NM} \sum_{p=1}^{N} \sum_{q=1}^{M} x_{pq}(t); \\ 0, & \text{otherwise.} \end{cases} \quad (6)$$

4 Simulation results

We simulate the TCNN for the FAP on 8 benchmarks. The specifications of the 8 benchmarks are listed in Table 1. Benchmarks 1-5 are from [4] and 6-8 are from [14]. Once the difference of the energy function value between two iteration steps is smaller than a threshold (0.00001) in 3 consecutive steps or the number of iteration steps exceeds a predefined number (15000 in our simulation), the iteration is terminated.

Initial inputs of neural networks $y_{ij}(0)$, $(i = 1, \ldots, N, j = 1, \ldots, M)$ are randomly generated from $[-1, 1]$. Parameters for the neural network are chosen as follows [2]:

$$\varepsilon = 0.004, \quad k = 0.999, \quad \alpha = 0.0015,$$
$$\beta = 0.001, \quad z(0) = 0.1.$$

The weight coefficients of the energy function W_i, $i = 1, \ldots, 4$ are chosen as follows :

$$W_1 = 1.0,\ W_2 = 1.0,\ W_3 = 0.7,\ W_4 = 0.0002.$$

Table 2. The performance of the TCNN on 8 instances. # denotes the instance number. I_L is the best largest interference and I_T is the best total interference. “Opt rate” stands for the rate that the TCNN reached the optimum in the 1000 runs. “Ave. error” denotes the average error from the optimum. T is the average number of iteration steps. η is the convergence rate. “SD” stands for “standard deviation”.

#	I_L			I_T	T	η
	Best	Opt rate %	Ave. error		mean±SD	(%)
1	30	32.1	5.4	100	426.5 ± 81	100
2	4	48.8	0.8	13	829.4 ± 117	100
3	7	21.6	1.3	85	2485 ± 153	96.4
4	64	16.7	10.5	919	2383 ± 264	87.5
5	697	18.6	53.8	7574	2342 ± 280	61.0
6	49	26.7	28.3	963	2651 ± 391	53.7
7	98	21.3	1.2	3889	3716 ± 389	67.1
8	994	27.4	3.9	60587	4839 ± 447	52.2

It is necessary to tune these coefficients to obtain better performance of the TCNN. Along with the growing problem size, the value of the W_4 term increases, so does the differences between numerical values of the W_1 term and W_2, W_3 terms. Hence, we slightly decrease W_2 and W_3, but increase W_4 as the problem size grows.

The algorithm is run 1000 times with different randomly generated initial neuron states on each of 8 benchmarks. Table 2 shows results, including the best largest interference I_L, the rate to reach the optimum (Opt rate), the average error from the optimal result (Ave. error), the convergence rate η (the ratio at which the neural network finds a feasible solution in 1000 runs), and the total interference I_T when the optimum of the largest interference is found. The average iteration steps T and standard deviations are also shown in this Table. The convergence rate denotes the ratio that the neural network finds a feasible solution at the end of iterations. The results show that the TCNN is effective in reducing the largest interference and total interference by rearranging the frequency assignment.

Comparison of the TCNN with the GNN [4] and the HopSA [14] is shown in Table 3. Results of the GNN and the HopSA are from references [4, 14]. As authors in [14] did not publish the average value, only the best result is included in Table 3. We show that the TCNN is comparable with the GNN and the HopSA in terms of the largest inter-

Third International Conference on Natural Computation (ICNC 2007)
0-7695-2875-9/07 $25.00 © 2007 IEEE

Table 3. Comparison of simulation results (largest interference and total interference) obtained by the TCNN, GNN and HopSA for instances 1 to 8. "SD" stands for "standard deviation".

#	GNN [4]				HopSA [14]		TCNN			
	Largest		Total		Largest	Total	Largest		Total	
	Best	mean	Best	mean			Best	mean±SD	Best	mean±SD
1	30	31.5	100	100.8	N/A	N/A	30	35.4±4.6	100	112.6±13.7
2	4	4.9	13	15.4	N/A	N/A	4	4.8±0.4	13	15.4±0.8
3	7	8.1	85	99.4	N/A	N/A	7	8.4±1.3	85	90.6±16.7
4	64	77.1	880	982.0	84	886	64	74.1±12.5	880	1045±125
5	640	766.8	8693	9413.9	817	6851	697	749±73.8	7574	8527±238
6	49	N/A	1218	N/A	48	1002	49	77.3±9.3	963	1126±132
7	100	N/A	4633	N/A	98	4093	98	99.2±1.2	3889	4722±282
8	1000	N/A	70355	N/A	988	61330	994	998±3.9	60587	70340±2295

ference and outperforms in terms of the total interference, especially on large-size problems.

5 Conclusion

In this paper, we solve the FAP in satellite communications through the TCNN. With rich and complex dynamics, the TCNN has more chance to reach the global optimum compared with the HNN. Simulation results on 8 benchmark problems show that the TCNN can find better solutions compared to the previous methods with very low computational cost.

References

[1] L. Chen and K. Aihara. *Transient chaotic neural networks and chaotic simulated annealing*, pages 347–352. Amsterdam: Elsevier Science Publishers, 1994.

[2] L. N. Chen and K. Aihara. Chaotic simulated annealing by a neural network model with transient chaos. *Neural Networks*, 8(6):915–930, 1995.

[3] Z. Ding, H. Leung, and Z. Zhu. A study of the transiently chaotic neural network for combinatorial optimization. *Mathematical and computer modelling*, 36(9):1007–1020, 2002.

[4] N. Funabiki and S. Nishikawa. A gradual neural-network approach for frequency assignment in satellite communication systems. *IEEE Trans. Neural Networks*, 8(6):1359–1370, 1997.

[5] Y. He. Chaotic simulated annealing with decaying chaotic noise. *IEEE Transactions on Neural Networks*, 13(6):1526 – 1531, 2002.

[6] J. J. Hopfield. Neurons with graded response have collective computational properties like those of two-state neurons. *Proc. Nat. Academy Science*, 81:3088–3092, May. 1984.

[7] T. Mizuike and Y. Ito. Optimization of frequency assignment. *IEEE Trans. Communications*, 37(10):1031–1041, 1989.

[8] R. Montemanni, J. N. Moon, and D. H. Smith. An improved tabu search algorithm for the fixed-spectrum frequency-assignment problem. *IEEE Transactions on vehicular technology*, 52(3):891–901, May 2003.

[9] R. A. Murphey, P. M. Pardalos, and M. G. C. Resende. *Frequency assignment problems*, volume Supplement Volume A. Kluwer Academic Publishers, 1999.

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

[11] H. Nozawa. Solution of the optimization problem using the neural-network model as a globally coupled map. *Physica D*, 75:179–189, 1994.

[12] H. E. Rauch and T. Winarske. Neural networks for routing communication traffic. *IEEE Cont. Syst. Mag.*, 8(2):26–31, 1988.

[13] S. Salcedo-Sanz and C. B-Calzón. A hybrid neural-genetic algorithm for the frequency assignment problem in satellite communications. *Applied Intelligence*, 22(3):207–217, May 2005.

[14] S. Salcedo-Sanz, R. Santiago-Mozos, and C. B-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(2):1108–1116, 2004.

[15] L. P. Wang, S. Li, F. Y. Tian, and X. J. Fu. A noisy chaotic neural network for solving combinatorial optimization problems: stochastic chaotic simulated annealing. *IEEE Transactions on System, Man, and Cybernetics-Part B: Cybernetics*, 34(5):2119–2125, 2004.

[16] L. P. Wang and K. Smith. On chaotic simulated annealing. *IEEE Transactions on Neural Networks*, 9(4):716–718, July. 1998.

[17] X. Xu, Z. Tang, and J. Wang. A method to improve the transiently chaotic neural network. *Neurocomputing*, 67:456–463, March 2005.

Third International Conference on Natural Computation (ICNC 2007)
0-7695-2875-9/07 $25.00

IEEE COMPUTER SOCIETY