Computers and Mathematics with Applications 57 (2009) 1024–1029

Contents lists available at ScienceDirect

Computers and Mathematics with Applications

journal homepage: www.elsevier.com/locate/camwa

Minimizing interference in satellite communications using transiently chaotic neural networks

Wen Liu, 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, Singapore

ARTICLE INFO

Keywords:
FAP
Satellite communications
Neural network
Optimization

ABSTRACT

The frequency assignment problem (FAP) in satellite communications is solved 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, we construct a TCNN consisting 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 as compared to the existing algorithms.

1. Introduction

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 the technique for reducing the interference itself. For both intersystem interference and intrasystem interference [1], frequency rearrangements take advantage of carrier interleaving and are effective in practical situations. Early efforts have focused on various analytical methods for evaluations of cochannel interference [2,3], rather than systematic methods for optimizing frequency assignments and for reducing cochannel interference. Among diverse formulations and objectives of frequency assignment problems (FAP) [4,5], we focus on frequency assignments in satellite communications in this paper. The objective of the satellite FAP is to minimize the cochannel interference between two satellite systems by rearranging the frequency assignments. This NP-complete problem is difficult to solve, especially for large-size problems, but is growing in importance, since we increasingly depend on satellites to fulfill our communications needs.

Some seminal work has been done in this area:

- Mizuike and Ito [1] proposed segmentation of the frequency band and presented a method based on the branch-and-bound approach.
- Funabiki and Nishikawa [6] 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 in the problem.
- Salcedo-Sanz et al. combined the Hopfield network with simulated annealing (HopSA) [7] and the genetic algorithm (NG) [8] to solve the problem.

On another hand, in the development history of modern computation technology, bio-inspired neural networks [9], due to their intrinsic operation functions, have played a very important role. Derived from some aspects of neurobiology and

* Corresponding author at: School of Electrical and Electronic Engineering, Nanyang Technological University, Block S1, 50 Nanyang Avenue, Singapore 639798, Singapore.
E-mail addresses: liuw0004@ntu.edu.sg (W. Liu), elpwang@ntu.edu.sg (L. Wang).

0898-1221/$ – see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.camwa.2008.10.026

adapted to integrated circuits, neural networks have been used widely in various fields such as optimization, linear and nonlinear programming, pattern recognition [10–12] and so on.

On the basis of Hopfield neural networks (HNN) [13,14], chaotic neural networks were presented by Nozawa [15,16] through adding negative self-feedback connections into Hopfield networks. The simulation for several combinatorial optimization problems showed that chaotic search is efficient in approaching the global optimum or sub-optima. As a kind of complicated nonlinear dynamics, chaos has been widely investigated by not only mathematicians and physicists, but also engineers, economists, and scientists from various disciplines [17–19]. Chaotic dynamics have several special characteristics, such as:

1. a sensitivity to initial conditions,
2. determinism, as the system function is well defined,
3. long term unpredictability.

Chaotic dynamics [20] is a complex behavior which can be generated by a finite set of deterministic nonlinear equations with a simple system. Chaos is globally stable and locally unstable [21]. A lot of research in recent years has focused on developing techniques to harness chaos when it is undesirable or to generate chaos so that the useful function of a chaotic system can be utilized. Chen and Aihara [22] proposed a transiently chaotic neural network (TCNN) by introducing a decaying negative self-feedback. The dynamics of the new model are characterized as transient chaos. Numerical experiments on the traveling salesman problem (TSP) and the maintenance scheduling problem showed that the TCNN has high efficiency for converging to globally optimal solutions.

The TCNN, which is also known as chaotic simulated annealing [10], is not problem-specific but a powerful general method for addressing combinatorial optimization problems (COPs) [23–25]. With autonomous decreases of the self-feedback connection, TCNNs are more effective in solving COPs compared to the HNN. 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. Parameter 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 [22] model is described as follows:

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

$$y_{ij}(t+1) = k y_{ij}(t) + \alpha \left[\sum_{p=1, p\neq i}^{N} \sum_{q=1, q\neq j}^{M} w_{ijpq} x_{pq}(t) + I_{ij} \right] - z(t)\left[x_{ij}(t) - I_0\right] \tag{2}$$

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

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 weight of connection 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 [14]:

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

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

3. Problem formulation

We use the FAP formulation given by [1] and used in [6–8]. As indicated in [1], 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.

We continue using the neural network formulation given by Funabiki and Nishikawa [6]. An N-carrier–M-segment FAP between two systems is formulated on an $N \times M$ neural network. At the end of the neural computing, neuron outputs will decide the assignment in the following way:

$$x_{ij} = \begin{cases} 1, & \text{then carrier } i \text{ is assigned to segment } j; \\ 0, & \text{no assignments.} \end{cases} \tag{5}$$

The energy function for the TCNN of the FAP [1,6,26] is defined as

$$E_1 = \sum_{i=1}^{N} \left(\sum_{j=1}^{M} x_{ij} - 1 \right)^2. \tag{6}$$

This term is to ensure that every segment in system 2 must be assigned to one and at most one segment in system 1.

$$E_2 = \sum_{i=1}^{N} \sum_{j=1}^{M} \sum_{\substack{p=1 \\ p \neq i}}^{N} \sum_{q=j-c_p+1}^{j+c_i-1} x_{ij} x_{pq}. \tag{7}$$

This E_2 term is to ensure that all segments of one carrier in system 2 should be assigned to consecutive segments in system 1 in the same order. Here c_i denotes the carrier length of carrier i. If carrier i is assigned to segment j, any other carrier must not be assigned to segments from j to $(j + c_i - 1)$. The first segment of carrier $p (p \neq i)$ should be assigned to the segment before $(j - c_p + 1)$ or after $(j + c_i - 1)$. As $(j - c_p + 1)$ may be negative and $(j + c_i - 1)$ may exceed the total number of the segments, i.e., M, the formulation in (7) has errors and produces program bugs during simulations. We revised the second term of the energy function as follows:

$$E_2' = \sum_{i=1}^{N} \sum_{j=1}^{M} \sum_{\substack{p=1 \\ p \neq i}}^{N} \sum_{q=\max(j-c_p+1,1)}^{\min(j+c_i-1,M)} x_{ij} x_{pq}. \tag{8}$$

where $\max(x, y)$ is the larger value between (x, y) and $\min(x, y)$ is the smaller value between (x, y). The W_1 and W_2 terms are designed to guarantee that the solution is a feasible one.

We add the following convergence term into the energy function to force the neuron outputs to approach to 0 or 1 [26]:

$$E_3 = \sum_{i=1}^{N} \sum_{j=1}^{M} x_{ij}(1 - x_{ij}). \tag{9}$$

The E_4 term optimizes the interference after the frequency rearrangement.

$$E_4 = \sum_{i=1}^{N} \sum_{j=1}^{M} d_{ij} x_{ij} \tag{10}$$

where d_{ij} is the element on row i column j of the cost matrix D. Cost matrix $D = (d_{ij}, i = 1, \ldots, N; j = 1, \ldots, M)$ is computed from the interference matrix $E^{(I)}$ [6]. If the carrier length for carrier i is 1, i.e., $c_i = 1$, then line i for carrier i in the cost matrix is the same as that in the interference matrix for carrier i. If $c_i > 1$, then we choose the largest value in the diagonal line for each j.

The total energy function is given by the summation of four parts E_1, E_2', E_3, and E_4:

$$\begin{aligned} E = {} & \frac{W_1}{2} \sum_{i=1}^{N} \left(\sum_{j=1}^{M} x_{ij} - 1 \right)^2 + \frac{W_2}{2} \sum_{i=1}^{N} \sum_{j=1}^{M} \sum_{\substack{p=1 \\ p \neq i}}^{N} \sum_{q=\max(j-c_p+1,1)}^{\min(j+c_i-1,M)} x_{ij} x_{pq} \\ & + \frac{W_3}{2} \sum_{i=1}^{N} \sum_{j=1}^{M} x_{ij}(1 - x_{ij}) + \frac{W_4}{2} \sum_{i=1}^{N} \sum_{j=1}^{M} d_{ij} x_{ij}. \end{aligned} \tag{11}$$

where W_i, $i = 1, \ldots, 4$, are weighting coefficients. The choices of these parameters are based on the rule that all terms in the energy function should be comparable in magnitude, so that none of them dominates [5]. The balance of each term in the energy function is crucial to parameter selection.

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	06	1–2	5–55
2	4	06	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

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 [27]:

$$x^b_{ij} = \begin{cases} 1, & \text{if } x_{ij} > \dfrac{1}{NM}\displaystyle\sum_{p=1}^{N}\sum_{q=1}^{M} x_{pq}(t); \\ 0, & \text{otherwise.} \end{cases} \tag{12}$$

Substituting the energy function in (2) with (4) and (11), we obtain the network dynamics, i.e., the difference equation of the neural network as follows:

$$\begin{aligned} y_{ij}(t+1) = {} & k y_{ij}(t) - W_1\left(\sum_{q=1}^{M} x_{iq} - 1\right) - \frac{W_2}{2}\sum_{\substack{p=1\\ p\neq i}}^{N}\sum_{q=\max(j-c_p+1,1)}^{\min(j+c_i-1,M)} x_{pq} \\ & - \frac{W_3}{2}(1-2x_{ij}) - \frac{W_4}{2}\mathrm{d}_{ij} - z(t)\left[x_{ij}(t) - I_0\right]. \end{aligned} \tag{13}$$

4. Simulation results

We simulate the TCNN for the FAP on eight benchmarks. The specifications of the eight benchmarks are listed in Table 1. Benchmarks 1–5 are from [6] and 6–8 are from [7]. Once the difference of the energy function value between two iteration steps is smaller than a threshold (0.00001) in three consecutive steps or the number of iteration steps exceeds a predefined number (15 000 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 [22]:

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

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

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

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, and so do the differences between the 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.

To investigate the efficiency of the TCNN, the algorithm is run 1000 times with different randomly generated initial neuron states on each of eight benchmarks. Table 2 shows the results, including the best largest interference I_L, the rate for reaching 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 numbers of iteration steps T and standard deviations are also shown in this table. The convergence rate denotes the rate for the neural network finding a feasible solution at the end of the iterations.

The results show that the TCNN is effective in reducing the largest interference and total interference by rearranging the frequency assignment. Take problem 4 for example, which is a 10-carrier–32-segment FAP; the optimal largest interference is 64 with an optimal rate at 16.7%. The best total interference found by the TCNN is 919. The algorithm takes 2383 iteration steps on average, and the TCNN converges to a feasible solution in 87.5% of the 1000 random runs.

The comparison of the TCNN with the GNN [6] and the HopSA [7] is shown in Table 3. Results from the GNN and the HopSA are from references [6,7]. As the authors in [7] 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 interference and outperforms them in terms of the total interference, especially on large-size problems.

Table 2
The performance of the TCNN in eight 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 for the TCNN reaching 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; mean ± SD	η (%)
	Best	Opt. rate (%)	Ave. error			
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	7 574	2342 ± 280	61.0
6	49	26.7	28.3	963	2651 ± 391	53.7
7	98	21.3	1.2	3 889	3716 ± 389	67.1
8	994	27.4	3.9	60 587	4839 ± 447	52.2

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 [6]				HopSA [7]		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	7 574	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	3 889	4722 ± 282
8	1000	N/A	70355	N/A	988	61 330	994	998 ± 3.9	60 587	70 340 ± 2295

5. Conclusion

We solve the FAP in satellite communications through the TCNN. The objective of this NP-complete optimization problem is to minimize cochannel interference between two satellite systems by rearranging frequency assignments. The TCNN model consists of $N \times M$ transiently chaotic neurons for an N-carrier–M-segment problem. With rich and complex dynamics, the TCNN has more chance of jumping out of local optima to reach the global optimum compared with the HNN. Simulation results on eight benchmark problems show that the TCNN can find better solutions compared to the previous methods, with low computational cost.

References

[1] T. Mizuike, Y. Ito, Optimization of frequency assignment, IEEE Transactions on Communications 37 (10) (1989) 1031–1041.
[2] B. Pontano, Interference into angle-modulated systems carrying multichannel telephony signals, IEEE Transactions on Communications 21.
[3] M. Jeruchim, A survey of interference problems and applications to geostationary satellite networks, in: Proceedings IEEE, 1977, pp. 317–331.
[4] R.A. Murphey, P.M. Pardalos, M.G.C. Resende, Frequency Assignment Problems, vol. Supplement Volume A, Kluwer Academic Publishers, 1999.
[5] L.P. Wang, S. Li, F.Y. Tian, 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) (2004) 2119–2125.
[6] N. Funabiki, S. Nishikawa, A gradual neural-network approach for frequency assignment in satellite communication systems, IEEE Transactions on Neural Networks 8 (6) (1997) 1359–1370.
[7] S. Salcedo-Sanz, R. Santiago-Mozos, C. Bousono-Calzón, A hybrid Hopfield network-simulated annealing approach for frequency assignment in satellite communications systems, IEEE Transactions on Systems, Man, and Cybernetics—Part B: Cybernetics 34 (2) (2004) 1108–1116.
[8] S. Salcedo-Sanz, C.B. Calzón, A hybrid neural-genetic algorithm for the frequency assignment problem in satellite communications, Applied Intelligence 22 (3) (2005) 207–217.
[9] S. Haykin, Neural Networks—A comprehensive Foundation, second edition, Prentice Hall International Inc., Hamilton, Canada, 1999.
[10] L.P. Wang, K. Smith, On chaotic simulated annealing, IEEE Transactions on Neural Networks 9 (4) (1998) 716–718.
[11] L.P. Wang, Multi-associative neural networks and their applications to learning and retrieving complex spatio-temporal sequences, IEEE Transactions on Systems, Man, and Cybernetics 29B (1) (1999) 73–82.
[12] S.H. Huang, H.-C. Zhang, Artificial neural networks in manufacturing: concepts, applications, and perspectives, IEEE Transactions on Components, Packaging, and Manufacturing Technology—Part A 17 (2) (1994) 212–228.
[13] J.J. Hopfield, Neural networks and physical systems with emergent collective computational abilities, Proceedings of the National Academy of Sciences 79 (1982) 2554–2558.
[14] J.J. Hopfield, Neurons with graded response have collective computational properties like those of two-state neurons, Proceedings of the National Academy of Sciences 81 (1984) 3088–3092.
[15] H. Nozawa, A neural-network model as a globally coupled map and applications based on chaos, Chaos 2 (3) (1992) 377–386.
[16] H. Nozawa, Solution of the optimization problem using the neural-network model as a globally coupled map, Physica D 75 (1994) 179–189.
[17] L.O. Chua, C.W. Wu, A. Huang, G.-Q. Zhong, A universal circuit for studying and generating chaos–part i: Routes to chaos, IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications 40 (10) (1993) 732–744.
[18] K. Aihara, Chaos engineering and its application to parallel distributed processing with chaotic neural networks, Proceedings of the IEEE 90 (5) (2002) 919–930.

[19] M. Delgado-Restituto, A. Rodriguez-Vazquez, Integrated chaos generators, Proceedings of the IEEE 90 (5) (2002) 747–767.
[20] I. Tokuda, K. Aihara, T. Nagashima, Adaptive annealing for chaotic optimization, Physical Review E 58 (4) (1998) 5157–5160.
[21] S. Sharma, An exploratory study of chaos in human–machine system dynamics, IEEE Transactions on Systems, Man, and Cybernetics—Part A: Systems and humans 36 (2) (2006) 319–326.
[22] L.N. Chen, K. Aihara, Chaotic simulated annealing by a neural network model with transient chaos, Neural Networks 8 (6) (1995) 915–930.
[23] Z. Ding, H. Leung, Z. Zhu, A study of the transiently chaotic neural network for combinatorial optimization, Mathematical and computer modelling 36 (9) (2002) 1007–1020.
[24] X. Xu, Z. Tang, J. Wang, A method to improve the transiently chaotic neural network, Neurocomputing 67 (2005) 456–463.
[25] Y. He, Chaotic simulated annealing with decaying chaotic noise, IEEE Transactions on Neural Networks 13 (6) (2002) 1526–1531.
[26] H.E. Rauch, T. Winarske, Neural networks for routing communication traffic, IEEE Control Systems Magazine 8 (2) (1988) 26–31.
[27] L. Chen, K. Aihara, Transient Chaotic Neural Networks and Chaotic Simulated Annealing, Elsevier Science Publishers, Amsterdam, 1994, p. 347–352.