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:

\[
\begin{align*}
    x_{ij}(t) & = \frac{1}{1 + e^{-y_{ij}(t)/\epsilon}} \\
    y_{ij}(t+1) & = ky_{ij}(t) + \alpha \sum_{p=1, p \neq i}^{N} \sum_{q=1, q \neq j}^{M} w_{ijpq} x_{pq}(t)
\end{align*}
\]
where the variables are

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

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

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

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 \(E^{(t)}\) for a 4-carrier-6-segment system. * denotes infinity.

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

\begin{equation}
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_{p=1}^{N} \sum_{q=1,q \neq j}^{M} \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{equation}

where \(W_i, i = 1, \ldots, 4,\) are weighting coefficients. The \(W_1\) and \(W_2\) terms are designed to guarantee that the solu-
Table 1. Specifications of the FAP instances used in the simulation.

<table>
<thead>
<tr>
<th>#</th>
<th>Number of carriers N</th>
<th>Number of segments M</th>
<th>Range of carrier length</th>
<th>Range of interference</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4</td>
<td>6</td>
<td>1-2</td>
<td>1-55</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>6</td>
<td>1-2</td>
<td>1-9</td>
</tr>
<tr>
<td>3</td>
<td>10</td>
<td>32</td>
<td>1-8</td>
<td>1-10</td>
</tr>
<tr>
<td>4</td>
<td>10</td>
<td>32</td>
<td>1-8</td>
<td>1-100</td>
</tr>
<tr>
<td>5</td>
<td>10</td>
<td>32</td>
<td>1-8</td>
<td>1-1000</td>
</tr>
<tr>
<td>6</td>
<td>18</td>
<td>60</td>
<td>1-8</td>
<td>1-50</td>
</tr>
<tr>
<td>7</td>
<td>30</td>
<td>100</td>
<td>1-8</td>
<td>1-100</td>
</tr>
<tr>
<td>8</td>
<td>50</td>
<td>200</td>
<td>1-8</td>
<td>1-1000</td>
</tr>
</tbody>
</table>

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]:

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

(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]:

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

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

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

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, 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 neural 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 $\eta$ (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-

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. $\eta$ is the convergence rate. “SD” stands for “standard deviation”.

<table>
<thead>
<tr>
<th>#</th>
<th>$I_L$</th>
<th>Opt rate</th>
<th>Ave. error</th>
<th>$I_T$</th>
<th>$T$ mean±SD</th>
<th>$\eta$ (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>30</td>
<td>32.1</td>
<td>5.4</td>
<td>100</td>
<td>426.5 ± 81</td>
<td>100</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>48.8</td>
<td>0.8</td>
<td>13</td>
<td>829.4 ± 117</td>
<td>100</td>
</tr>
<tr>
<td>3</td>
<td>7</td>
<td>19.7</td>
<td>0.1</td>
<td>85</td>
<td>2485 ± 153</td>
<td>96.4</td>
</tr>
<tr>
<td>4</td>
<td>167</td>
<td>10.5</td>
<td>919</td>
<td>2383 ± 264</td>
<td>87.5</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>697</td>
<td>18.6</td>
<td>7574</td>
<td>2342 ± 280</td>
<td>61.0</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>49</td>
<td>26.7</td>
<td>28.3</td>
<td>963</td>
<td>2651 ± 391</td>
<td>55.7</td>
</tr>
<tr>
<td>7</td>
<td>8</td>
<td>21.3</td>
<td>1.2</td>
<td>3889</td>
<td>3716 ± 389</td>
<td>67.1</td>
</tr>
<tr>
<td>8</td>
<td>994</td>
<td>27.4</td>
<td>3.9</td>
<td>60587</td>
<td>4839 ± 447</td>
<td>52.2</td>
</tr>
</tbody>
</table>

1, \ldots, 4 are chosen as follows:

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