# Hybrid Cross-Entropy Method/Hopfield Neural Network for Combinatorial Optimization Problems

Emilio G. Ortiz-García† and Ángel M. Pérez-Bellido†

†Department of Signal Theory and Communications, Universidad de Alcalá, 28871 Alcalá de Henares, Madrid, Spain.

**Abstract.** This paper presents a novel hybrid algorithm for combinatorial optimization problems based on mixing the cross-entropy (CE) method and a Hopfield neural network. The algorithm uses the CE method as a global search procedure, whereas the Hopfield network is used to solve the constraints associated to the problems. We have shown the validity of our approach in several instance of the generalized frequency assignment problem.

## 1 Introduction

The hybridization of heuristic algorithms with local search methods has attracted the attention of many researchers in the last few years [1]-[3]. The interest in these methods is mainly due to the fact that they allow a robust solving of a huge number of difficult problems, in which finding a solution is very hard (impossible in some cases) using traditional optimization methods.

In the literature different ways of hybridizing algorithms can be found. In every hybrid algorithm two sub-procedures coexist: a global search procedure, which is the fundamental part of the algorithm, and a local search one which complements the global search. A classification of hybrid algorithms can be done, depending on the role played for the local search procedure, that includes hybrid [1] and memetic [3] algorithms. The first ones include local search procedures in order to solve the problem constraints. In this way, the global search works only with feasible solutions. However, in the memetic approach, the local search procedure is used to improve the performance of the global search. In both methods, the hybridization of the local procedures can be in Lamarckian mode or in Baldwin mode. In Lamarckian mode, the resulting individual of the local search replaces to the previous individual in the global search. In Baldwinian mode, the individual is not replaced in the global search, but only the objective function value of the solution obtaining after the local search procedure is considered in the global search.

In this paper we focus our work in hybrid algorithms (which use the local search to solve constraints problem). Specifically, we study a new hybrid algorithm which arises of the hybridization of a global search procedure based in the Cross-Entropy method (CE) [5] and a binary Hopfield neural network (HNN)

[6] as local search procedure. Firstly, we carry out an introduction on the main characteristics of two methods mentioned, and later on, we describe how the hybrid algorithm we propose is organized. Finally, we prove that this hybrid algorithm improves the results obtained with a genetic algorithms as global search procedure in a well known combinatorial optimization problem with constraints.

The rest of the paper is structured as follows: Section 2.1 presents the binary Hopfield neural network used in our approach. Section 2.2 presents a introduction to the Cross-Entropy method including a brief summary of its origin and a discussion of the algorithm in combinatorial optimization problems with binary encoding. Section 3 presents the main characteristics of the HNN-CE algorithm proposed, which is applied to solve the Generalized Frequency Assignation problem in Section 4. Finally, the section 5 finishes the paper giving some conclusions.

## 2 Background

### 2.1 The binary Hopfield neural network

In order to satisfy the COP's constraints of a combinatorial optimization problem with binary encoding, we use a binary Hopfield neural network (HNN). The HNN used belongs to a class of binary HNNs [8], [9] where the neurons can only take values 1 or 0.

The structure of the HNN can be described as a graph, where the set of nodes are the neurons, and the set of edges define the connections between the neurons. We map a neuron to an element in the solution $\mathbf{X}$. In order to simplify our notations, we shall use $\mathbf{X}$ to denote the neurons in the HNN. The HNN dynamics can then be described as follows: After a random initialization of every neuron with binary values, the HNN operates in serial mode. This means that only one neuron is updated at a time. Denoting by $x_i(t)$ the state of a neuron on time $t$, the updating rule is described by

$$x_i(n+1) = \phi(u_i(n)) \tag{1}$$

where $\phi$ is the updating rule, and $u_i(n)$ is the input equation of the neuron $i$ at time $n$, which depends on the output equations of the rest of neurons through the COP's constraints.

In this updating rule, neurons $x_{ik}$ are updated in their natural order, i.e., $i = 1, 2, \ldots, n$, $k = 1, 2, \ldots, m$. A modification of this rule can be introduced by performing the updating in a random ordering of the rows (variable $i$). This way the variability in the feasible solution found is increased. Let $\pi(i)$ be a random permutation of $i = 1, 2, \ldots, n$. The new updating rule of the HNN will be

$$x_i(n+1) = \phi(u_{\pi(i)}(n)) \tag{2}$$

We can define a *cycle* as the update of all neurons in a certain order. In a *cycle* each neuron is updating in the order given by $\pi$. After two cycles, we consider that the HNN has converged (in that moment no neuron will change its state). In this way, the final HNN state is a potential solution of the optimization problem considered which fulfills the COP's constraints.

### 2.2 The Cross-Entropy method for optimization

The CE method is an elegant optimization method which comes from the simulation of rare events field [4], [5]. The idea is that the simple algorithm of cross-entropy used for estimating probabilities of rare events could be adapted to solve difficult problems of optimization as well. This adaptation is done by translating the "deterministic" optimization framework to a "stochastic" one, and then using rare event simulation techniques similar to the ones described in [7].

The CE method can be split in two phases:

- Generate a random data sample for the problem, according to a specified mechanism.
- Update the parameters of the method, mainly the probabilities associated to the variables of the solution.

The formal description of the CE for optimization can be done following [7]: Consider an optimization problem consisting in finding the solution (*state*) $\mathbf{x}$ which maximizes a given real-valued, objective function $S$. Mathematically:

find $\mathbf{x}$ such that,

$$\max_{\mathbf{x}\in\chi} S(\mathbf{x}). \tag{3}$$

where $\chi$ is a finite set of states (search space).

The CE method associates an *estimation problem* with the optimization problem described by (3). To do this, a collection of indicator functions $\{I_{S(\mathbf{x})\geq\gamma}\}$ on $\chi$ is defined, for various levels $\gamma \in \mathbb{R}$. Then, let $\{f(.;\mathbf{v})\}$ be a family of discrete pdfs on $\chi$, parameterized by a real-valued vector $\mathbf{v} \in [0,1]^n$. We can associate then the optimization problem defined in (3) with the problem of estimating the number

$$l(\gamma) = \mathbb{P}_{\mathbf{v}}(S(X) \geq \gamma) = \sum_{\mathbf{x}} I_{\{S(\mathbf{x})\geq\gamma\}} f(\mathbf{x};\mathbf{v}) = \mathbb{E}_{\mathbf{v}} I_{\{S(\mathbf{x})\geq\gamma\}} \tag{4}$$

where $\mathbb{P}_{\mathbf{v}}$ is the probability measure under which the random state $X$ has pdf $f(.;\mathbf{v})$, and $\mathbb{E}_{\mathbf{v}}$ stands for the expectation operator. This problem is usually called *associated stochastic problem* [7].

Following [7], it can be shown that to estimate $l(\gamma)$, the optimal value for $\mathbf{v}$, so called $\mathbf{v}^*$, can be computed as:

$$\mathbf{v}^* = arg\max_{v} \mathbb{E}_{\mathbf{u}} I_{\{S(\mathbf{X})\geq\gamma\}} \ln f(\mathbf{X};\mathbf{v}). \tag{5}$$

This parameter can be estimated as:

$$\mathbf{v}^* = arg\max_{v} \frac{1}{N} \sum_{i=1}^{N} I_{\{S(\mathbf{X}_i)\geq\gamma\}} \ln f(\mathbf{X}_i;\mathbf{v}). \tag{6}$$

where $\mathbf{X_i}$ are generated from pdf $f(.;\mathbf{v})$.

Thus the CE algorithm for optimization can be stated as follows:

**General CE algorithm:**

1: Define an initial value $\mathbf{v_0}$, set $t = 1$.
2: Generate a sample $\mathbf{X_1}, \ldots, \mathbf{X_N}$ from density $f(.; \mathbf{v}_{t-1})$.
3: Calculate $\gamma_t$ as $\gamma_t = \max\{s : \mathbb{P}_{v_{t-1}}(S(\mathbf{X}) \geq s) \geq \rho\}$
4: Calculate $\mathbf{v_t}$ using equation (6).
5: If for some $t \geq d$, $\gamma_t = \gamma_{t-1} = \ldots = \gamma_{t-d}$, then stop;
6: Set t=t+1 and reiterate from step 2.

Several approximations can be done to simplify the general CE algorithm:

First, the value $\gamma_t = \max\{s : \mathbb{P}_{v_{t-1}}(S(\mathbf{X}) \geq s) \geq \rho\}$ can be estimated from the sample $\mathbf{X_1}, \ldots, \mathbf{X_N}$ as follows:

Calculate the performances $S(\mathbf{X}_i)$ for all $i$, and sort them out in increasing order: $S_{(1)} \leq S_{(2)} \leq \ldots S_{(N)}$, and then evaluate the $(1-\rho)$ sample quantile as

$$\hat{\gamma}_t = S_{(\lceil (1-\rho)N \rceil)} \tag{7}$$

A further simplification in the general CE algorithm can be obtained for 0-1 combinatorial optimization problems, if we consider the solution vector $\mathbf{X} = (X_1, \ldots, X_n)$ to be Bernoulli random variables, $\mathbf{X} \sim Ber(\mathbf{p})$. In this case the pdf is

$$f(\mathbf{X}; \mathbf{p}) = \prod_{i=1}^{n} p_i^{X_i} \left(1 - p_i\right)^{1 - X_i}, \tag{8}$$

since each $X_j$ can only be 0 or 1,

$$\frac{\partial}{\partial p_j} \ln f(\mathbf{X}; \mathbf{p}) = \frac{X_j}{p_j} - \frac{1 - X_j}{1 - p_j} = \frac{1}{(1 - p_j)p_j} \left(X_j - p_j\right). \tag{9}$$

using this result, we can calculate the maximum in Equation (6) by setting the first derivative with respect to $p_j$ to zero for all $j$:

$$\frac{\partial}{\partial p_j} \sum_{i=1}^{N} I_{\{S(\mathbf{X}_i \geq \gamma)\}} \ln f(\mathbf{X}_i; \mathbf{p}) = \frac{1}{(1 - p_j)p_j} \sum_{i=1}^{N} I_{\{S(\mathbf{X}_i \geq \gamma)\}} (X_{ij} - p_j) = 0. \tag{10}$$

and finally we obtain the following expression for the estimation of $p_j$:

$$p_j = \frac{\sum_{i=1}^{N} I_{\{S(\mathbf{X}_i \geq \gamma)\}} X_{ij}}{\sum_{i=1}^{N} I_{\{S(\mathbf{X}_i \geq \gamma)\}}}. \tag{11}$$

The CE algorithm for optimization of 0-1 problems can be stated now as follows:

**CE algorithm for combinatorial optimization:**

1: Define an initial value $\mathbf{p_0}$, say $\mathbf{p_0} = (\frac{1}{2}, \frac{1}{2}, \ldots, \frac{1}{2})$; set $t = 1$.
2: Generate a sample $\mathbf{X_1}, \ldots, \mathbf{X_N}$ of bernoulli vectors with success probability $\mathbf{p}_{t-1}$.

3: Calculate the performances $S(\mathbf{X}_i)$ for all $i$.
4: Sort the performances in increasing order $S_{(1)} \leq S_{(2)} \leq \ldots S_{(N)}$.
5: Estimate $\hat{\gamma}_t$ using Equation (7).
6: Estimate $\hat{\mathbf{p}}_t = (\hat{p}_{t,1}, \hat{p}_{t,2}, \ldots, \hat{p}_{t,n})$ using equation (11).
7: If for some $t \geq d$, $\gamma_t = \gamma_{t-1} = \ldots = \gamma_{t-d}$, then stop;
8: Set t=t+1 and reiterate from step 2.

Several improvements have been proposed to this algorithm, like the *smoothed updating* [7], consisting in estimating vector $\hat{\mathbf{p}}_t$ using Equation (11) followed by a smoothing step:

$$p_{t,j} = \alpha p_{t,j} + (1 - \alpha)\, p_{t-1,j} \tag{12}$$

## 3 The hybrid CE-HNN algorithm

As we describe in the section 2.1, the binary HNN repairs the unfeasible solutions of a combinatorial optimization problem with binary encoding, obtaining feasible solutions that fulfill the COP's constraints. However the HNN cannot ensure that these feasible solutions are optimum with regard to a given objective function. For this reason, many hybrid heuristics have been proposed using a HNN as local heuristic. As an example, Figure 1 shows the search space of a optimization problem where only a small portion are feasible solutions due to the inherent problem constraints. In this example, the HNN acts as a "filter" of unfeasible solutions, producing solutions in the correct region of the search space. Many algorithms have been used as global search heuristics to be mixed with HNNs, for example Genetic Algorithms [10]-[13] or Simulated Annealing [14], [15].

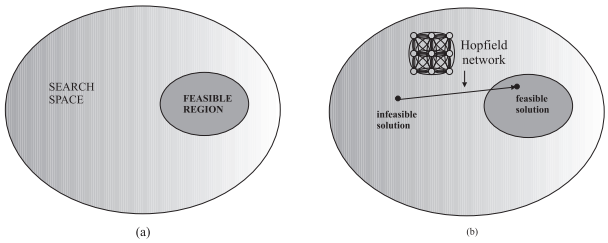



**Fig. 1.** Example of the effect of a Hopfield network in reducing the search space size: the unfeasible solutions are "filtered", producing feasible ones.

In our case, we present a new hybrid heuristic algorithm formed by the binary HNN described in Section 2.1 and the CE method described in the section 2.2, in order to solve optimization problems with binary encoding and constraints

among the components of the solution. In this algorithm the solutions are encoded by means of a vector of bits with $0-1$ values, which we denominate $x$.

Given a solution $x$, the CE method considers each component of the sequence as a random variable with Bernoulli pdf. Firstly, the CE initializes the vector $x$ to $x = \{\frac{1}{2}, \frac{1}{2}, \ldots, \frac{1}{2}\}$. Then, using the vector $x$ as probabilities, the CE generates a sample of possible solutions. Note that usually these solutions are not feasible solutions. Thus, these solutions are repaired by means of the binary HNN, obtaining feasible solutions which replace the original ones. Following to this step, the objective function is evaluated for each feasible solution and the obtained values are sorted in ascending order (maximization problem). Then the parameter $\gamma$ of the expression (7) is re-calculated. Finally, the CE also recalculates the probabilities of the vector $x$ by means of the Equation (11) and applying la Equation (12). This procedure is recursive, and finalizes when the parameter $\gamma$ has the same value $k$ times in a row ($k$ is a design parameter). Figure 2 shows an schema of the hybrid HNN-CE. Note that the Hopfield network repairs the solutions generated by means of the probabilities vector $x$, being the generated feasible ones used to update the search parameters of the CE method, i.e. $\gamma$ and $x$.

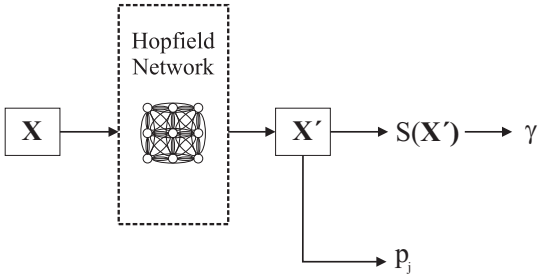



**Fig. 2.** Schema of the hybrid HNN-CE algorithm.

## 4 Experiments in the generalized frequency assignment problem

The Generalized Frequency Assignment Problem (GFAP hereafter) in a mobile communications network can be stated as follows: given a mobile communication network formed by $N$ cells, and a set of $M$ available frequencies, achieve an assignment of frequencies to every cell, in such a way that the system is free from interferences, and the number of assigned frequencies is maximized.

The minimum distance between frequencies in the system in order to avoid interferences is given by an integer matrix $D$. The solution of the problem can

be represented as an $N \times M$ binary matrix $X$, where the rows represents the cells and the columns represents the available frequencies. An element $x_{ij} = 1$ means that the frequency $j$ has been assigned to the cell $i$. Some articles which have dealt with this problem before are [16], [18].

Mathematically the problem can be defined as follows:

Find an assignment $X$, which maximizes

$$\max \left( f(X) = \sum_{i=1}^{N} \sum_{j=1}^{M} x_{ij} \right) \tag{13}$$

and such as, if $x_{ik} = 1$ and $x_{jl} = 1$,

$$|k - l| \geq d_{ij} \tag{14}$$

Note that the definition of GFAP coincides with the type of problem which is possible to solve directly by means of the hybrid HNN-CE, i.e, binary encoding and constraints among the components of the solution. In this case, the vector $x$ which represents the solution in the HNN-CE is formed by the elements of the matrix of frequencies $X$ through the next equation:

$$x_{(i-1)\cdot m+j} = X_{ij}, \quad i = 1, \ldots, n; j = 1, \ldots, m \tag{15}$$

in such a way that exists a equivalence between the matrix and the vector and the Hopfield network can be represented as a matrix with the neurons $x_{ij}$. Furthermore, the dynamics of the Hopfield network depends on the matrix $D$ which define the minimum distance between a neuron with state "1" and the next neuron in the same column with state "1" too. In consequence, the equation of input to the neuron and the updating function (see equation 2) depends on the matrix $D$ according to the following equations:

$$u_{ik}(t) = isgn \left( \sum_{\substack{p=1 \\ p \neq i}}^{n} \sum_{\substack{q=max(1,k-D_{i,p}+1) \\ q \neq k}}^{min(m,k+D_{i,p})} x_{pq} \right) \ \forall \ i, \ k, \tag{16}$$

$$\phi(u_{ik}) = \begin{cases} 0 & if \ \ u_{ik} > 0 \\ 1 & otherwise. \end{cases} \tag{17}$$

In this way the HNN gives feasible solutions, i.e. solutions which fulfill the Equation (14), while the CE method searches for the optimal among these feasible solutions.

Four problems of different sizes have been studied, though the evolution of the algorithm has only been analyzed in instance 4 defined in [16]. We will compare our algorithm with other hybrid approach which uses a genetic algorithm in order to optimize the problem (HNN-GA). The Hopfield network is the same

in the two algorithms so we can verify the performance of both algorithms in similar simulation conditions.

The parameters which configure the hybrid approaches are: in the CE algorithm we use the parameter $1 - \rho$ (quantile) of 0.95, $N$ (number of samples in each iteration) of 100 and the stop parameter fixed to 5 [7]. In the genetic algorithm we use a population of 100 individuals for 200 generations, roulette wheel selection, 2-point cross and mutation by change of bit [17].

Table 1 shows the main characteristics of the problems GFAP which we use. The first problem instance considered is a small one, the second can be considered a medium one and others ones are large GFAP instances. The fourth instance (marked by a *) has been considered previously in several articles about frequency assignment [16], [18] and [19] and it will be used to analyze the convergence properties of the algorithms.

**Table 1.** Main characteristics of GPAG instances considered.

| Problem | N | M |
|---|---|---|
| 1 | 10 | 20 |
| 2 | 20 | 40 |
| 3 | 30 | 60 |
| 4* | 25 | 73 |

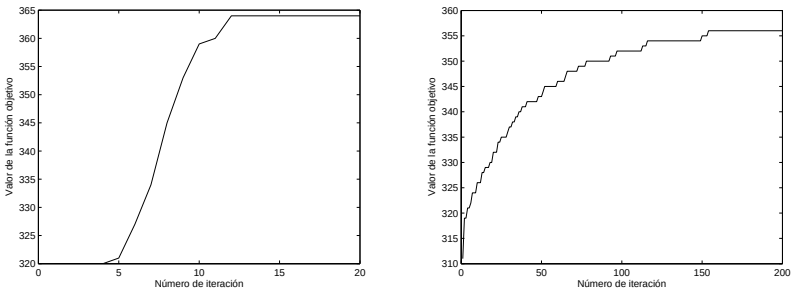
Table 2 shows the results of 30 simulations of each problem by means of the average value and the best fitness obtained. It can be verified how in instances 1 and 2 both algorithms present similar behavior, but when the size of the problem is incremented (instances 3 and 4) the CE algorithm performs better than the HNN-GA. This shows that the CE algorithm presents better scalability than the HNN-GA.

**Table 2.** Average and best results for 30 experiments in each instances.

| Problem | HNN-EC (average) | HNN-EC (best) | HNN-GA (average) | HNN-GA (best) |
|---|---|---|---|---|
| 1 | 77.4 | **80** | 78.0 | **80** |
| 2 | 194.1 | **200** | 197.4 | **200** |
| 3 | 357.6 | **363** | 338.6 | 349 |
| 4* | 359.6 | **364** | 350.3 | 356 |

Other important point is the convergence of the algorithms. In order to analyze it we use the fourth instance. Figures 3 (a) and (b) shows an example of the convergence of both algorithms. First, note that, whereas in the HNN-GA the calculation of the average is very simple because the number of generations is constant in the 30 experiments, the HNN-CE algorithm is stopped when the

parameter $\gamma$ has the same value 5 times in a row. This is why we only show the experiment in which the best solution was obtained. We can see that the HNN-CE algorithm has a faster convergence than the HNN-GA. Note that the HNN-GA needed more than 150 generations in order to converge (100 individuals by generation) whereas the HNN-CE needed only 13 iterations (100 samples by iteration).




(a) (b)

**Fig. 3.** Example of convergence of HNN-CE and HNN-GA approaches for problem 4; (a) HNN-CE convergence; (b) HNN-GA convergence.

In Figure 3 is possible to see how the two hybrid algorithms follow the typical dynamics of the global algorithms that form them. For example, the genetic algorithm has an evolution with small improvements each few generations, typical characteristic of this algorithm. However, the hybrid HNN-CE has a trajectory with a larger slope, i.e. good improvements in a short time. The dynamics of a HNN-CE rare time extends beyond 30 iterations, though this point depends on the values of the parameters used in the algorithm.

## 5 Conclusions

In this paper we have presented a hybrid algorithm based on the Cross-entropy method and a Hopfield neural network. The description of the hybrid algorithm and its main properties has been discussed. The performance of this approach has been shown by solving a combinatorial optimization problem known as the generalized assignment problem, where the HNN-CE algorithm outperforms another hybrid approach which includes a genetic algorithm.

## References

1. E. Talbi, "A taxonomy of hybrid metaheuristics," *Journal of Heuristics*, vol. 8, pp. 541-564, 2002.

2. C. Blum and A. Roli, "Metaheuristics in combinatorial optimization: overview and conceptual comparison," *ACM Computing Surveys*, vol. 35, pp. 268-308, 2003.
3. N. Krasnogor and J. Smith, "A tutorial for competent memetic algorithms: model, taxonomy, and design issues," *IEEE Trans. Evol. Comput*, vol. 9, no. 5, pp. 474-488, 2005.
4. R. Y. Rubinstein, "The simulated entropy method for combinatorial and continous optimization," *Methodology and computing in applied probability*, vol 2, pp. 127-190, 2002.
5. R. Y. Rubinstein and D. P. Kroese, *The cross-entropy method: a unified approach to combinatorial optimization, Monte-carlo simulation and machine learning*, Springer-Verlag: NY, 2004.
6. J. Hopfield, "Neurons and physical systems with emergent collective computational abilities," *Proc. Natl. Acad. Sci. USA*, vol. 79, pp. 2554-2558, 1982.
7. P. De Boer, D. P. Kroese, S. Mannor and R. Y. Rubinstein, "A tutorial on the cross-entropy method", *Annals of Operations Research*, vol. 134, pp. 19-67, 2005.
8. Y. Shrivastava, S. Dasgupta, and S. Reddy, "Guaranteed convergence in a class of Hopfield networks," *IEEE Trans. Neural Networks*, vol. 3, pp. 951-961, 1992.
9. S. Salcedo-Sanz, and C. Bousoño-Calzón, "A portable and scalable algorithm for a class of constrained combinatorial optimization problems," *Computers & Operations Research*, vol. 32, pp. 2671-2687, 2005 .
10. J. Balicki, A. Stateczny and B. Zak, "Genetic algorithms and Hopfield neural networks to solve combinatorial optimization problems," *Applied Mathematics and Computer Science*, vol. 10, no. 3, pp. 568-592, 1997.
11. Y. Watanabe, N. Mizuguchi and Y. Fujii, "Solving optimization problems by using a Hopfield neural network and genetic algorithm combination," *Systems and Computers in Japan*, vol. 29, no. 10, pp. 68-73, 1998.
12. S. Salcedo-Sanz and C. Bousoño-Calzón, "A hybrid neural-genetic algorithm for the frequency assignment problem in satellite communications," *Applied Intelligence*, vol. 22, pp. 207-218, 2005.
13. C. Bousoño-Calzón and A. R. Figueiras-Vidal "Emergent techniques for dynamic frequency assignment: merging genetic algorithms and neural networks," In *Proc. of the RTO IST Symposium on Frequency Assignment, Sharing and Conservation in Systems (Aerospace)*, vol. 32, pp. 12/1-12/5, 1998.
14. S. Salcedo-Sanz, R. Santiago-Mozos and C. Bousoo-Calzn, "A hybrid Hopfield network-simulated annealing approach for frequency assignment in satellite communications systems," *IEEE Trans. Systems, Man and Cybernetics, part B*, vol. 34, no. 2, pp. 1108-1116, 2004.
15. C. Calderón-Macías, M. K. Sen and P. L. Stoffa, "Hopfield neural networks, and mean field annealing for seismic deconvolution and multiple attenuation," *Geophysics*, vol. 62, no. 3, pp. 992-1002, 1997.
16. N. Funabiki and Y. Takefuji, "A neural network parallel algorithm for channel assignment problems in cellular radio networks," *IEEE Trans. Veh. Technol.* vol. 42, no. 4, pp. 430-437, 1992.
17. D. Goldberg, *Genetic algorithms in search, optimization and machine learning*, Addison-Wesley, Reading:MA, 1989.
18. W. K. Lai and G. C. Coghill, "Channel assignment through evolutionary optimization," *IEEE Trans. Veh. Technol.*, vol. 55, no. 1, pp. 91-95, 1996.
19. L.M. San Jos-Revuelta, "A new adaptive genetic algorithm for fixed channel assignment," *Information Sciences*, vol. 177, no. 13, pp.2655-2678, 2007.