Applied Intelligence 22, 207–217, 2005

A Hybrid Neural-Genetic Algorithm for the Frequency Assignment Problem in Satellite Communications

S. SALCEDO-SANZ*,† AND C. BOUSOÑO-CALZÓN
Department of Signal Theory and Communications, Universidad Carlos III de Madrid, Avda. de la Universidad 30, 28911 Leganés-Madrid, Spain
sancho@tsc.uc3m.es

Abstract. A hybrid Neural-Genetic algorithm (NG) is presented for the frequency assignment problem in satellite communications (FAPSC). The goal of this problem is minimizing the cochannel interference between satellite communication systems by rearranging the frequency assignments. Previous approaches to FAPSC show lack of scalability, which leads to poor results when the size of the problem grows. The NG algorithm consists of a Hopfield neural network which manages the problem constraints hybridized with a genetic algorithm for improving the solutions obtained. This separate management of constraints and optimization of objective function gives the NG algorithm the properties of scalability required.

We analyze the FAPSC and its formulation, describe and discuss the NG algorithm and solve a set of benchmark problems. The results obtained are compared with other existing approaches in order to show that the NG algorithm is more scalable and performs better than previous algorithms in the FAPSC.

Keywords: combinatorial optimization, frequency assignment, genetic algorithms, Hopfield neural networks, satellite communications

1. Introduction

Frequency assignment problems (FAPs) arise in many different situations in the context of wireless communications. Mobile telephony, TV broadcasting or satellite communications are some examples [1, 2]. These applications lead to different models, types of instances and solving techniques for FAPs, and include problems like:

(1) Planning models for permanent spectrum allocation, which maximize the utilization of all available spectra [3, 4].
(2) On-line algorithms for dynamically assigning frequencies to users in a established network (mainly wireless communication networks) [5, 6].
(3) Planning models for network and systems design, like mobile, broadcast or satellite networks [2, 7].

In this paper we focus on planning the design of satellite communication systems, in order to reduce the system cochannel interference. This reduction of the cochannel interference has arisen as one major factor for determining satellite systems design [1, 8]. In addition, with the increase of geostationary satellites, this interference reduction has become an even more important issue, due to the necessity of accommodating as many satellites as possible in geostationary orbit. To cope with interference reduction, the rearrangement of frequency assignments is considered an effective measure in practical situations [9].

Frequency rearrangement can be formulated as a combinatorial optimization problem known as Frequency Assignment Problem for Satellite Communications (FAPSC hereafter). FAPSC belongs to a class of optimization problems with constraints, in which a goal

*This work was supported in part by CICYT under grant TIC 1999-0216.
†Author to whom all correspondence should be addressed.

function must be optimized and a set of constraints have to be fulfilled for a solution to be feasible. In this kind of problem, scalability is a major factor of the algorithm's design, due to the poor performance of non-scalable algorithms when the size of the problem grows. In this context, FAPSC has been solved before by using emerging methods such as branch and bound [9] and Hopfield neural networks [8]. Both techniques have the problem of lack of scalability, which leads to poor quality solutions in large, difficult problems.

In this paper we propose a hybrid Neural-Genetic algorithm (NG) for the FAP, in which a fast serial Hopfield neural network (HNN) manages the problem's constraints and a simple Genetic Algorithm (GA) searches for high-quality solutions. We show that our algorithm is more scalable, due to the separate management of constraints and goal function, and achieves better results than existing algorithms for the FAPSC.

The rest of the paper is organized as follows: in the next section we define and analyze the FAPSC. In Section 3 the hybrid Neural-Genetic algorithm is described, by studying the Hopfield neural network and the GA which form it. Section 4 shows the performance of the NG algorithm, by solving a set of benchmark problems and comparing the results obtained with previous algorithms for the FAPSC. In this section some discussion about the design of the NG algorithm is provided. Finally, Section 5 ends the paper with some concluding remarks.

2. Problem Formulation

Given two adjacent satellite systems as in Fig. 1, the FAPSC consists of reducing the inter-system cochannel interference by rearranging the frequency assignment on carriers in system #2, while the assignment in system #1 remains fixed.

Due to the fact that each carrier usually occupies a different length in a frequency band [9], introduced the segmentation of carriers, in such a way that every carrier can be described by a collection of consecutive unit segments.

In a system with M segments, an interference $M \times M$ matrix E is defined, in which the e_{ij} element stands for the cochannel interference when segment #i in system #2 uses a common frequency with segment #j in system #1 (see Fig. 2 as an example).

The constraints of the FAPSC are:

(C1) Every segment in system #2 must be assigned to a segment in system #1.

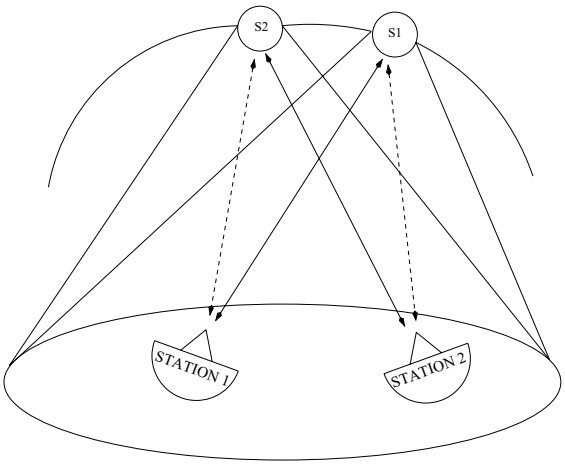


Figure 1. Outline of cochannel interference.

		C11		C12		C13	
		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	20	45	0	15	35
C24	S25	45	5	25	0	50	*
	S26	*	45	25	0	10	50

Figure 2. Example of interference matrix. Symbols * stand for an infinity interference.

(C2) Every segment in system #1 can be assigned by at most one segment in system #2.

(C3) All the segments of each carrier in system #2 must be assigned to consecutive segments in system #1 in the same order.

The frequency reassignment in a system with N carriers and M segments can be represented by a square $M \times M$ matrix F, called the reassignment matrix, in such a way that $f_{ij} = 1$ means that the segment #i in system #2 has been reassigned to segment #j in system #1. Following [8], the management of constraint C3 is performed by using a mixed representation to solve the

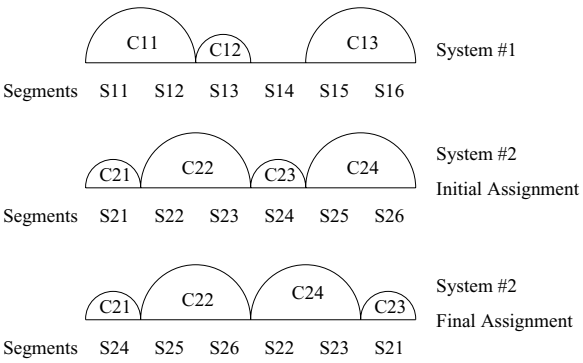


Figure 3. Segmentation of the system defined by Figs. 1 and 2.

problem: another matrix $\tilde{F}$, $N \times M$, derived from F is defined, such that $\tilde{f}_{ij} = 1$ means that first segment of carrier #i in system #2 has been reassigned to segment #j in system #1, and the following segments of the carrier go behind consecutively. Figure 3 shows an example of assignment matrices $\tilde{F}$ and F for the interference matrix E in Fig. 2. Note that matrix F can be obtained in a straight forward manner from matrix $\tilde{F}$ knowing the number of carriers of system #2 and their lengths in segments.

In addition, we also need to define a matrix C, $N \times N$, in which every element c_{ij} stands for the minimum separation in segments between two carriers #i and #j.

Taking into account the definitions above, we can mathematically formulate the frequency assignment problem as follows:

Achieve an assignment $\tilde{F}$ (recall that F is obtained from $\tilde{F}$) such that:

$$\min(\gamma(E, F)) \tag{1}$$

subject to:

$$\sum_{i=1}^{N} \tilde{f}_{ij} = 1 \quad j = 1, \ldots, M \tag{2}$$

and in such a way that the assignment $\tilde{F}$ fulfils the constraints in C, i.e., if $\tilde{f}_{ij} = 1$ and $\tilde{f}_{pq} = 1$ then $|j - q| \geq c_{ip}$.

Where $\gamma(E, F)$ represents an objective function depending on the interference matrix E and assignment matrix F. In this paper we consider two objective functions.

First, we consider an objective function for the FAP which requires that the total interference of the systems to be minimum:

$$\gamma_1(E, F) = \sum_{i=1}^{M} \sum_{j=1}^{M} e_{ij} \cdot f_{ij}. \tag{3}$$

and second, we introduce an objective function which minimizes the maximum peak of interference between the systems (largest interference):

$$\gamma_2(E, F) = \max(e_{ij} \cdot f_{ij}) \quad \forall\, i, j \tag{4}$$

2.1. *An Example*

An example of a small FAPSC instance may clarify concepts. First, consider the two systems (satellite-station) depicted in Fig. 1. Imagine that the interference matrix between the two systems, E, is the one in Fig. 2. Both systems have $M = 6$ segments, and system #2 has $N = 4$ carriers. Note that the *range of carrier* (length in segments of the minimum and the maximum carrier in the system) in this example is 1–2 (the smaller carrier has 1 segment and the larger carrier has 2 segments). The *range of interference* is defined as the minimum (not 0) and maximum (not infinity) values in matrix E. In this example the range of interference is 5–55.

The FAPSC consists of reassigning carriers of system #2, whereas system #1 is fixed. Figure 4 illustrates the segmentation of the systems, and a possible reassignment when interference matrix in Fig. 2 is considered.

Figure 3 shows this assignment in the mixed representation we use to solve the problem. Figure 3(a) shows matrix $\tilde{F}$. Note that this matrix fulfils the constraint in Eq. (2) (one "1" per row in $\tilde{F}$), and also fulfils the constraints in C (separation in segments between one "1" and the following in $\tilde{F}$ is at least equal to the length of the carrier first "1" belongs). In Fig. 3(b) we can see how to get matrix F from $\tilde{F}$, only knowing the carrier's length. This matrix F will be used to calculate the objective function associated to the problem.

3. The Neural-Genetic Algorithm

The algorithm we propose for solving the FAPSC consists of a hybrid global-local scheme, where a global algorithm looks for the minimization of the objective function, and a local procedure manages the fulfilment of FAPSC's constraints. We use a a standard Genetic Algorithm (GA) as a global algorithm, due to several

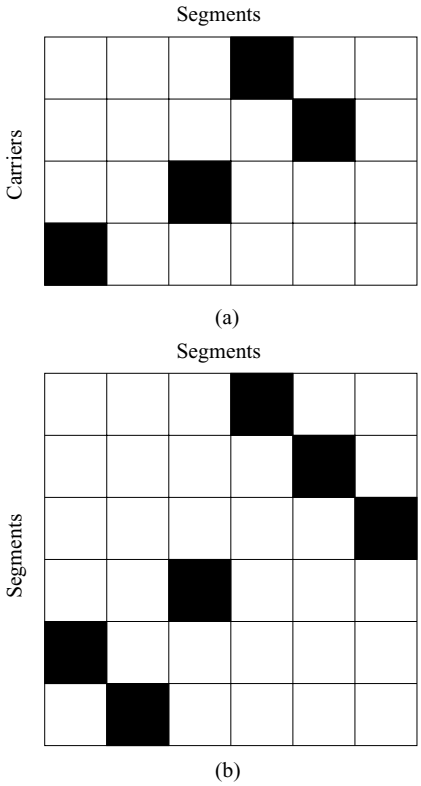


Figure 4. (a) Example of matrix $\tilde{F}$ for the interference matrix of Fig. 2 (shaded squares represent "1s" and whites squares "0s"). (b) Matrix F obtained from $\tilde{F}$.

reasons. First, they are known to be robust search algorithms [10]; second, the standard GA codify the solutions as binary strings, so the solution matrix $\tilde{F}$ can be codified in a GA as a binary string. This allows the construction of hybrid search schemes, mixing global and local algorithms. The separation of the global search from the constraints fulfilment gives our algorithm the desired properties of scalability, outperforming the existing algorithms in large difficult problems, as we will show in the experiments Section.

3.1. The Local Search Procedure

As was mentioned before, the local search procedure we use consists of a kind of Hopfield Neural Network (HNN) whose dynamics depends on the matrix C, and, of course, on the initial state of the neurons. This Hopfield network belongs to a class of digital Hopfield networks, where the neurons only can take the values 1 or 0, see [11] for further details. The structure of the HNN can be described as a graph, where the set of vertices are the neurons, and the set of edges defines the connections between the neurons. We map a neuron to every element in the solution matrix $\tilde{F}$. In order to simplify notation, we shall also use matrix $\tilde{F}$ to denote the neurons in the Hopfield network. The HNN dynamics can be described then in the following way: After a random initialization of every neuron with binary values, the HNN operates in serial mode. This means that only a neuron is updated at a time, while the rest remain unchanged. Denoting by $\tilde{f}_{ij}(t)$ the state of a neuron on time t, the update rule is described by:

$$\tilde{f}_{ij}(t) = i\mathrm{sgn}\left(\sum_{\substack{p=1\\p\neq i}}^{N}\sum_{\substack{q=\max(1,c_{i,p+1})\\q\neq j}}^{\min(M,j+c_{i,p})}\tilde{f}_{pq}\right)\forall i,\ j. \quad (5)$$

where the isgn operator is defined by:

$$i\mathrm{sgn}(a) = \begin{cases} 0 & \text{if} \quad a > 0 \\ 1 & \text{otherwise} \end{cases}$$

The discussion of this updating rule is important to understand how the digital Hopfield network proposed works: consider that a neuron $\tilde{f}_{ij}$ has to be updated.[1] Its state $\tilde{f}_{ij}$ depends only on the state of other neurons $\tilde{f}_{pq}$ within a distance of c_{ip} in columns. If the state of any of these $\tilde{f}_{pq}$ neurons in the neighbourhood of $\tilde{f}_{ij}$ is equal to 1, the solution given by the network would be infeasible, and therefore the neuron $\tilde{f}_{ij}$ must be updated to 0. In the case that all neurons $\tilde{f}_{pq}$ within a distance in columns of c_{ip} from $\tilde{f}_{ij}$ are 0, the neuron $\tilde{f}_{ij}$ will be updated to 1.

Following [11], the convergence of this neural network to a feasible solution is always guaranteed if the number of neurons is finite. In addition, it is expected that this neural network converges much faster to a feasible solution than traditional Hopfield networks defined by means of energy function terms, what makes this network suitable to be hybridized with global search heuristics like Genetic Algorithms.

It is important to note that in updating rule (5), the neurons $\tilde{f}_{ij}$ are updated in their natural order, i.e., $i = 1, 2, \ldots, N$, $j = 1, 2, \ldots, M$. We introduce a modification of this rule by performing the updating of the neurons in a random ordering of the rows (variable i). This way the variability in the feasible solution found increases. Let $\pi(i)$ be a random permutation of $i = 1, 2, \ldots, N$. The new updating rule of the HNN

results:

$$\tilde{f}_{\pi(i)j}(t) = i\mathrm{sgn}\left(\sum_{\substack{p=1\\p\neq\pi(i)}}^{N}\sum_{\substack{q=\max(1,c_{\pi(i),p+1})\\q\neq j}}^{\min(M,j+c_{\pi(i),p})}\tilde{f}_{pq}\right)\forall i,\ j. \tag{6}$$

This updating rule runs over the rows of $\tilde{F}$ in the order given by the permutation $\pi(i)$, but the columns are updated in its natural order $j = 1, 2, \ldots, M$. Note that the discussion carried on before on the performance of rule (5) is valid for this new rule (6), since only the order of updating has been modified.

We can define a *cycle* as the set of $N \times M$ successive neuron updates in a given order. In a cycle, every neuron is updated once following the given order $\pi(i)$, which is fixed during the execution of the algorithm. After every cycle, the convergence of the HNN is checked. The HNN is considered converged if none of the neurons have changed their state in the cycle. The final state of the HNN dynamics is a feasible solution for the FAPSC, which fulfils the constraints of the matrix C.

3.2. The Global Search Algorithm

The optimization of the objective function is performed in our algorithm by a Genetic Algorithm (GA), in the following way: A solution $\tilde{F}$ is codified in the GA as a $(N \times M)$-length binary string. The GA population is formed by a fix number of $(N \times M)$-length strings, χ, which codify several solutions to the problem. These solutions are called individuals of the population. The population is then evolved through successive generations by means of the application of the genetic operators: selection, crossover and mutation [10].

Selection is the process by which individuals are randomly sampled with probabilities inversely proportional to their fitness values (in order to minimize the objective function). In this case, values of fitness are given by the problem's objective function:

$$\gamma_1(E, F) = \sum_{i=1}^{M}\sum_{j=1}^{M} e_{ij} \cdot f_{ij}. \tag{7}$$

or

$$\gamma_2(E, F) = \max(e_{ij} \cdot f_{ij})\ \forall i, j \tag{8}$$

An elitist strategy, consisting in always passing the highest fitness string to the next generation, is applied in order to preserve the best solution encountered thus far in the evolution. The selected set, of the same size of the initial population, χ, is subjected to the crossover operation. First, the binary strings are coupled at random. Second, for each pair of strings, an integer position along the string is selected uniformly at random. Two new strings are composed by swapping all bits between the selected position and the end of the string. This operation is applied to the couples with probability P_c less than one.

By means of the mutation operation, every bit in every string of the population may be changed from 1 to 0, or vice versa, with a very small probability, P_m.

Finally, since crossover and mutation operators may cause the new string to be infeasible, this string is set as the initial state of the HNN, and the result of the neural algorithm substitutes it in the new population.

3.3. The Complete Algorithm

The complete algorithm for the FAPSC is formed by mixing the GA and the HNN, and performs in the following way: First, the GA population is initialized at random. At this stage, the individuals are not feasible solutions, so, every individual is passed to the HNN, which obtains a feasible one, and the fitness value associated to the individual is calculated. Once the GA population is formed only by feasible solutions to the problem, the genetic operators, selection, crossover and mutation, are applied as was explained in Section 3.2. Since the genetic operators modify the individuals, some of them might become infeasible. Thus, the individuals are again passed to the HNN and a new step of the NG algorithm begins.

Figure 5 shows a flowchart for the NG, and in pseudocode, it can be written as follows:

NG Algorithm
Initialize GA population at random
while (max. number of generations not reached) **do**
  for (every individual $\tilde{F}$)
    Run the HNN to obtain a feasible $\tilde{F}$.
    Obtain F from $\tilde{F}$ and calculate the fitness value of the individual through it.
    Substitute the individual in the GA by the feasible $\tilde{F}$.
  endfor
    selection
    crossover
    mutation
end while

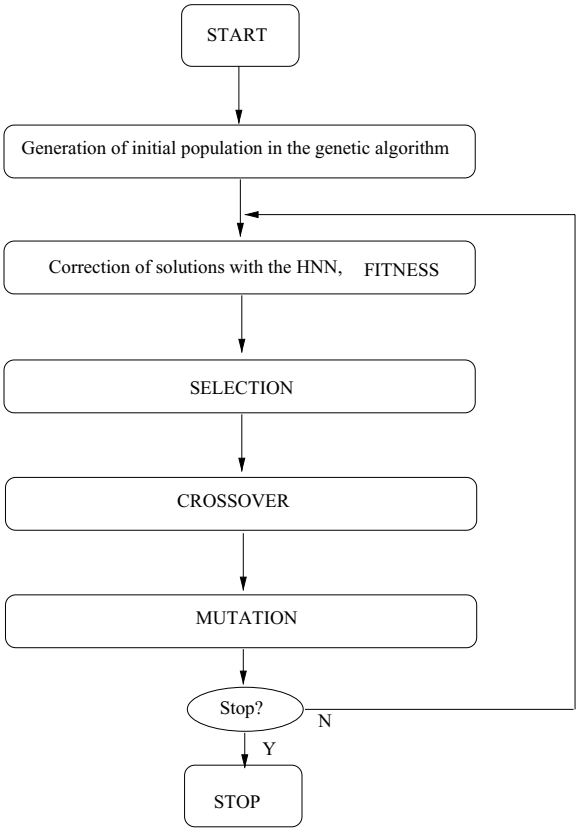


Figure 5. Flowchart for the NG algorithm.

4. Experiments

In order to test the performance of our Neural-Genetic algorithm, a set of benchmark problems from [8] have been selected. There are two easy problems, #1 and #2, one problem of medium difficulty, #3 and two hard problems, #4 and #5. A further description of these problems and their interference matrices can be found in [8]. In addition, four new hard problems have been added to the set of benchmark problems, and the algorithm in [8] has been programmed and applied to them for comparison purposes. The main characteristics of the benchmark problems are shown in Table 1. Note that problem #1 has been used as an example in Section 2.1.

The population of the GA was fixed to 50 individuals ($\chi = 50$), with a stop criterion based on the number of generations (maximum 1000). The best individual was passed to the next generation as has been pointed out in Section 3.2. Probabilities of crossover and mutation were fixed to $P_c = 0.6$ and $P_m = 0.01$, respectively.

Table 1. Main features of the set of benchmark problems.

Problem #	Carriers	Segments	Range of carrier	Range of interfer.
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	20	100	1–8	1–100
8	15	50	1–7	1–1000
9	50	200	1–8	1–1000

4.1. Results

Table 2 shows the results obtained by our NG algorithm and a comparison with other algorithms results for the benchmark problems considered, when total interference (objective function γ_1) is considered. The results shown in this table correspond to the best solutions encountered by the algorithms. In low difficulty problems, our NG algorithm achieves equal solution than the best existing algorithm, whereas in harder problems, #4 to #9, NG algorithm improves the results of other existing methods. The solution achieved by NG algorithm in problem #4 is 5% better than the best solution found by the best existing algorithm, 21% in problem #5, 12% in #6, 17% in #7, 21% in #8 and 11% better in problem #9. These results show that NG algorithm performs well in difficult problems, achieving better results and being much more scalable than existing algorithms.

Table 2. Comparison of the results obtained by the NG algorithm with previous approaches (total interference, γ_1). Problems #1 to #9.

Problem #	Mizuike and Ito [9]	Funabiki and Nishikawa [8]	NG
1	100	100	100
2	13	13	13
3	100	85	85
4	929	880	838
5	10330	8693	6851
6	–	1218	1075
7	–	4633	3860
8	–	16192	12860
9	–	70355	62853

Table 3. Comparison of the results obtained by the NG algorithm with previous approaches (largest interference, γ_2). Problems #1 to #9.

Problem #	Mizuike and Ito [9]	Funabiki and Nishikawa [8]	NG
1	30	30	30
2	4	4	4
3	8	8	8
4	67	64	64
5	803	640	640
6	–	49	43
7	–	100	96
8	–	919	687
9	–	1000	938

Table 3 shows the results obtained by the NG algorithm when the objective function γ_2 is considered (largest interference in the assignments). Our algorithm achieves equal or better results than other algorithms and the differences are again notorious in the hardest problems #6 to #9.

Figure 6(a) and (b), shows the best evolution of the genetic algorithm for problems #4 and #5, respectively (objective function γ_1 is considered). In these figures it is possible to check the performance of the GA: in generation 0 (random initialization of GA population) the best individuals obtained for Problems #4 and #5 have a value of total interference about 1250 and 11000, respectively. In the last generation of the GA, the quality of the solution found is much better: 838 and 6851, respectively. These results shown that the GA has a good performance in both problems.

In Fig. 7 the best solutions obtained by the NG algorithm for problems #4 (a) and #5 (b), are displayed as an example (objective function γ_1). Note that both solutions are feasible, fulfilling the problem's constraints given in Section 2. In the design of real systems, we would utilize these solutions in the case that the total interference of the systems must be controlled. In a system where the primal objective in its design is to control peaks of interference, the solution provided by objective function γ_2 should be used.

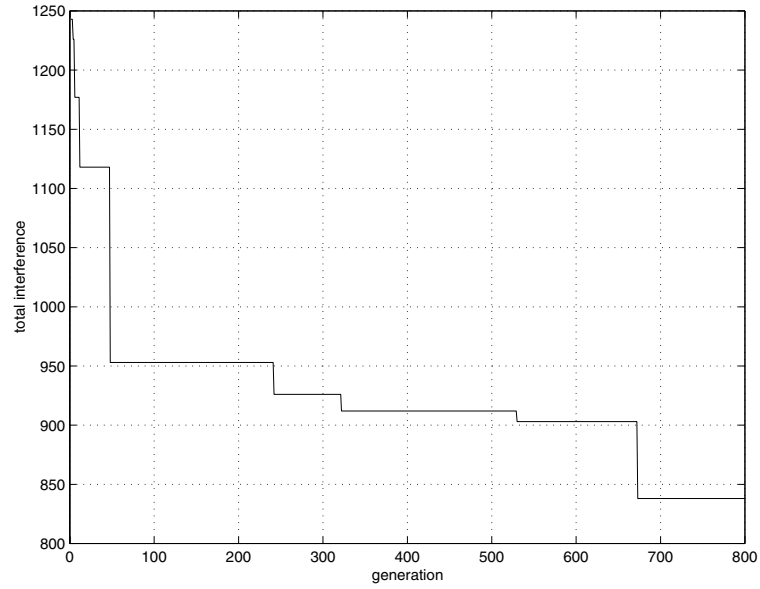
4.2. *NG and its Components: Effects of the Hybridization*

As has been shown before, the hybridization of the HNN and the GA produces a novel algorithm which is able to improve previous approaches for the FAPSC. However, it would be also interesting to compare the results obtained by the NG algorithm with the results obtained using the HNN on its own. Note that the binary HNN used in this paper is designed for managing the FSCRP constraints, without taking into account the objective function. We can include the objective function minimization into the binary HNN by adding a local search heuristic at the end of its dynamics. Once the HNN is considered converged, a number κ of swaps between carriers of the same length are applied. If a better solution is found, it is consider the current assignment, and the swapping process continues from it. Worse solutions than the current one are discarded. Figure 8 shows the process of swapping between carriers of the same length. This figure represents a problem with 6 carriers and 9 segments, in which the length of the carriers varies between 1 and 2. Two possible swaps between carriers of the same length are illustrated. Note that swapping between carriers of different length would be a much more complex process, which would provided unfeasible solutions in the majority of swaps.

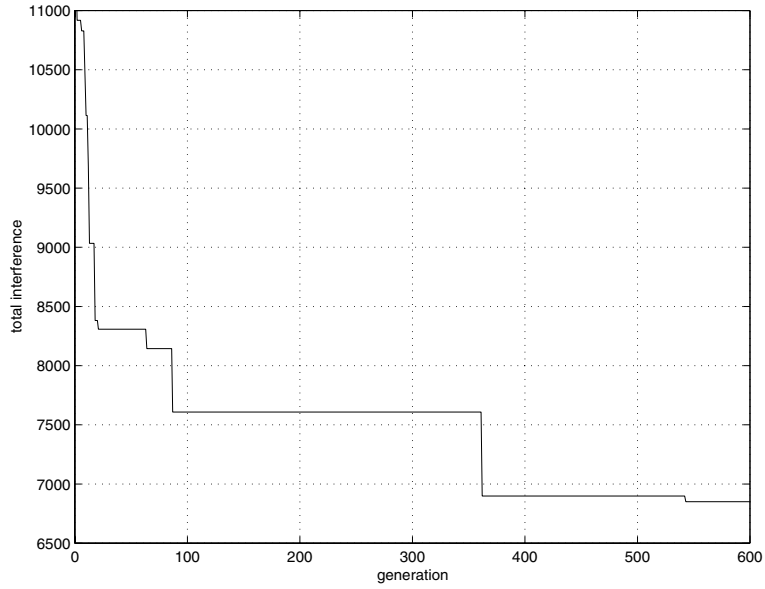
Table 4 shows a comparison between the results obtained by the HNN, the HNN with the local search heuristic ($\text{HNN}_{\text{local}}$ hereafter) and the NG algorithm. We launched 5000 HNN and $\text{HNN}_{\text{local}}$, fixing parameter $\kappa = 100$. Function γ_1 was used as objective function. Note that the HNN obtains poor results in the FAPSC. This result is expected, since the network is designed for providing feasible solutions without optimizing the objective function. The inclusion of the local search

Table 4. Comparison of the results obtained with the HNN, the HNN with local search heuristic incorporated and the NG algorithm.

Problem #	HNN	$\text{HNN}_{\text{local}}$	NG
1	100	100	100
2	13	13	13
3	112	87	85
4	1009	903	838
5	9740	7362	6851
6	1390	1263	1075
7	5023	4425	3860
8	18172	16202	12860
9	78558	69831	62853



(a)



(b)

Figure 6. (a) Evolution of the best individual in a run of the GA for problem #4; (b) Evolution of the best individual in a run of the GA for problem #5.

based on swapping carriers improves the HNN performance, as can be seen in this table. The results obtained by the HNN_{local} are better than the results obtained by the HNN in all cases, but the two first easy problems, where both networks obtain the same result. Note that the NG algorithm obtain results which are better than the HNN, with and without local search heuristic incorporated. There is a large difference between the HNN results and the NG results, which means that the hybridization of the HNN with the GA is the key for obtaining a powerful hybrid approach for the FAPSC.

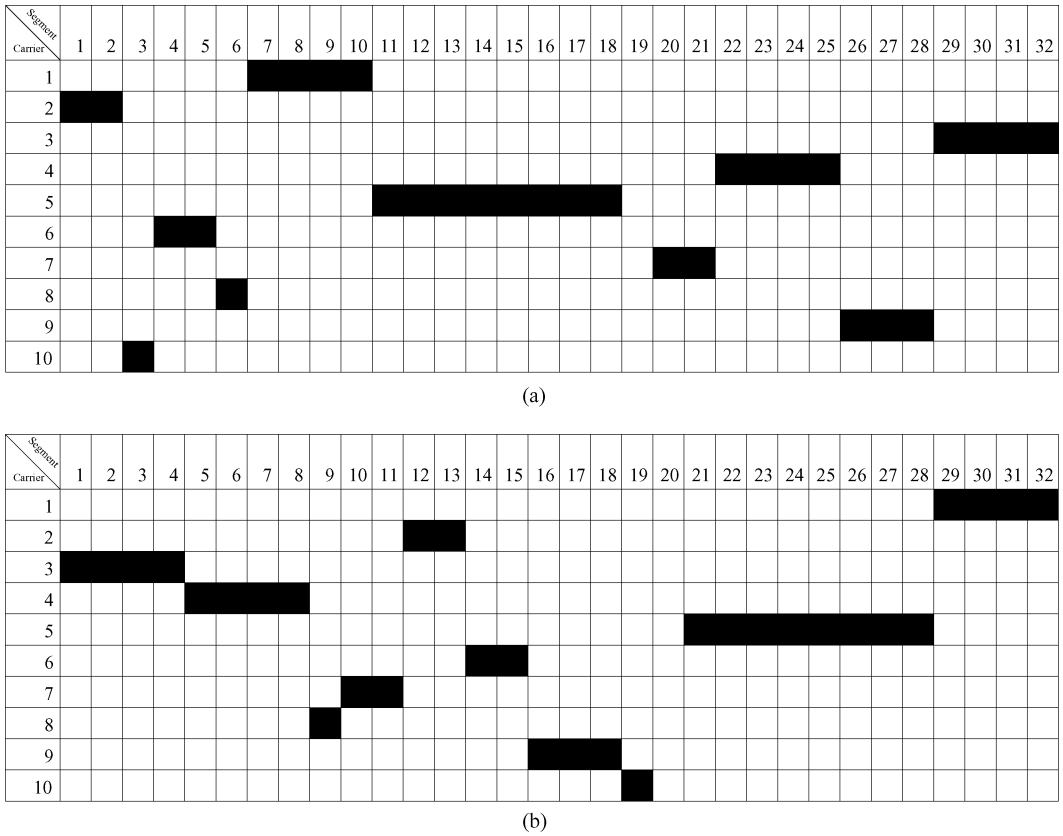


Figure 7. Best assignments ($\tilde{F}$) achieved by the NG algorithm in Problems #4 (a) and #5 (b), using objective function γ_1.

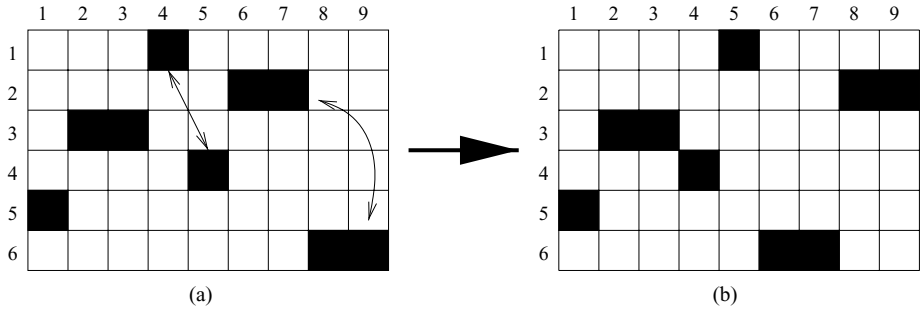


Figure 8. Example of local heuristic included in the HNN dynamics: (a) initial assignment, (b) final assignment.

4.3. Some Comments About the NG Algorithm's Design

The increasing of computational cost is the main drawback when using a hybrid algorithm in a combinatorial optimization problem as the FAPSC. Thus, the design of the local and global algorithms to be mixed must be as accurate as possible, taking into account the computational cost as a primary factor. The design of the NG algorithm follows the above hints: first, the GA uses standard genetic operators, as was shown in Section 3.2. Second, the HNN used is a fast digital network, with very good properties of convergence. We found that the HNN always achieves a feasible solution starting from an unfeasible one codified in the GA. In addition, the speed of convergence of the network is

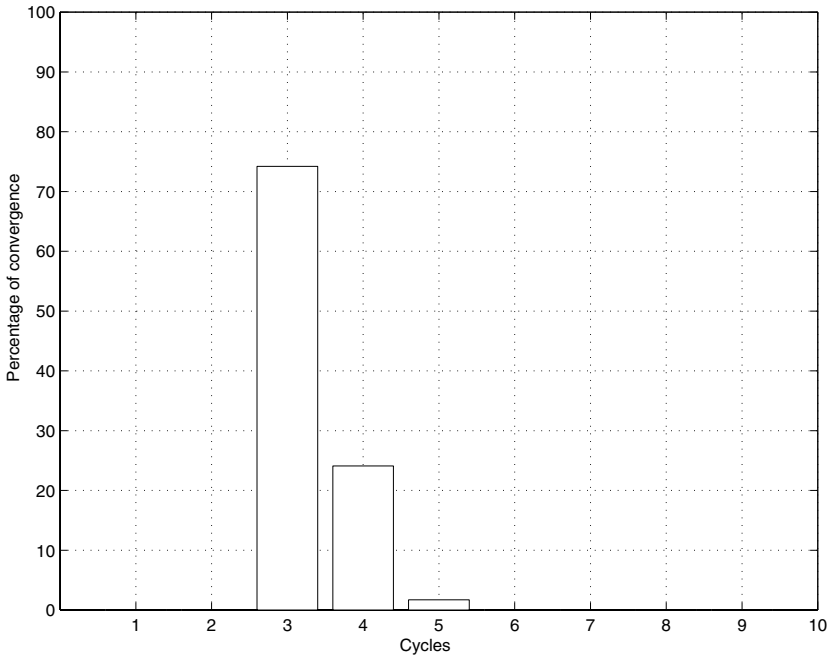


Figure 9. Number of cycles needed for the HNN convergence, in problem #5.

very good: Fig. 9 shows that the 75% of the networks launched for solving problem #5 converge in 3 cycles (see Section 3.1 for the definition of a cycle). Over the 23% of the networks launched converge in 4 cycles, and only about the 2% of the networks run converge in 5 cycles. The updating rule involving a random permutation of rows (see Section 3.1), does not modify these values, due to it only changes the order of updating, not the structure of the HNN dynamics. These data show that the HNN used in this paper fast enough to allow its mixing with a GA.

5. Conclusions

In this paper, a hybrid Neural-Genetic algorithm for the frequency assignment problem in satellite communications has been presented. The algorithm consists of a $N \times M$ Hopfield neural network (N-carriers, M-segments) which manages the problem's constraints, hybridized with a genetic algorithm which improves the solution obtained from the network. This approach for the FAPSC is more scalable than previous algorithms due to the separate management of constraints and goal function.

Simulations in a set of benchmark problems have shown very good performance of the algorithm, obtaining better solutions in terms of largest and total interference than existing algorithms, and showing the differences in scalability between the NG and the other algorithms.

Acknowledgment

The authors would like to thank the anonymous referees for their useful comments and hints.

Note

1. Recall that the updating of neuron $\tilde{f}_{ij}$ will only involves states 1 or 0 (digital network).

References

1. K.I. Aardal, S.P.M. Van Hoesel, C. Mannino, and A. Sassano, "Models and solution techniques for frequency assignment problems," ZIB Report 01-40, Dec. 2001, http://www.zib.de/PaperWeb/abstracts/ZR-01-40.
2. R.A. Murphey, P.M. Pardalos, and M.G.C. Resende, "Frequency assignment problems," *Handbook of Combinatorial Optimization*, Kluwer Academic Press, 1999.
3. S. Salcedo-Sanz, C. Bousoño-Calzón, and A.R. Figueiras-Vidal, "A mixed neural-genetic algorithm for the broadcast scheduling

problem," *Trans. Wireless Commun.*, vol. 2, no. 2, pp. 277–283, 2003.
4. G. Wang and N. Ansari, "Optimal broadcast scheduling in packet radio networks using mean field annealing," *IEEE Journal on Selec. Areas Commun.*, vol. 15, pp. 250–259, 1997.
5. H.G. Sandalidis, P.P. Stavroulakis, and J. Rodriguez-Tellez, "An efficient evolutionary algorithm for channel resource management in cellular mobile systems," *IEEE Tran. Evol. Comput.*, vol. 2, no. 4, pp. 125–137, 1998.
6. C. Voudouris and E.P.K. Tsang, "Solving the radio link frequency assignment problem using guided Local search," in *Proc. NATO Symposium on Radio Length Frequency Assignment, Sharing and Conservation Systems (Aerospace)*, Aalborg, Denmark, Oct. 1998.
7. D.H. Smith, S.M. Allen, S. Hurley, and W.J. Watkins, "Frequency assignment: Methods and algorithms," In *Proc. NATO Symposium on Radio Length Frequency Assignment, Sharing and Conservation Systems (Aerospace)*, Aalborg, Denmark, October 1998.
8. N. Funabiki and S. Nishikawa, "A gradual neural-network approach for frequency assignment in satellite communication systems," *IEEE Trans. Neural Networks*, vol. 8, no. 6, pp. 1359–1370, 1997.
9. T. Mizuike and Y. Ito, "Optimization of frequency assignment," *IEEE Trans. Commun.*, vol. 37, no. 10, pp. 1031–1041, 1989.
10. D.E. Goldberg, *Genetic Algorithm in Search, Optimization and Machine Learning*, Addison Wesley, 1989.
11. Y. Shrivastava, S. Dasgupta, and S.M. Reddy, "Guaranteed convergence in a class of Hopfield networks," *IEEE Trans. Neural Networks*, vol. 3, pp. 951–961, 1992.
12. R. Acosta, R. Bauer, R.J. Krawczyk, R.C. Reinhart, M.J. Zernic, and F. Gargione, "Advanced communications technology satellite (ACTS): Four-year system performance," *IEEE Journal on Selec. Areas Commun.*, vol. 17, no. 2, pp. 193–203, 1999.
13. N. Ansari, S.H. Hou, and Y. Youyi, "A new method to optimizate the satellite broadcast schedules using the mean field annealing of a Hopfield neural network," *IEEE Trans. Neural Networks*, vol. 6, no. 2, pp. 470–482, 1995.
14. F.Q. Bac and V.L. Perov, "New evolutionary genetic algorithm for NP-complete combinatorial optimization problems," *Biol. Cybern.*, vol. 69, no. 3, pp. 229–234, 1993.
15. J. Balicki, A. Stateczny, and B. Zak, "Genetic algorithms and Hopfield neural networks for solving combinatorial problems," *Appl. Math and Comp. Sci.*, vol. 7, no. 3, pp. 568–592, 1997.
16. C. Bousoño-Calzón and A.R. Figueiras Vidal, "Emerging techniques for dynamic frequency assignment: Merging genetic algorithms and neural networks," In *Proc. of Information Systems Technology Symposium*, Aalborg, Denmark, 1998, pp. 12.1–12.5.
17. W. Crompton, S. Hurley, and N.M. Stephens, "Applying genetic algorithms to frequency assignment problems," in *Proc. of the SPIE, the International Society for Optical Engineering*, 1994.
18. A. Koster, C. Van Hoesel, and A. Kolen, "Lower bounds for minimum interference frequency assignmet problems," *Ricerca Operativa*, vol. 30, no. 94–95, pp. 101–116, 2001.
19. V. Maniezzo and R. Montemanni, "An exact algorithm for the Min-interference frequency assignment problem," Tech. Report wp-cO0003, Scienze dell'Informazione, University of Bologna, 2000.
20. H. Okinaka, Y. Yasuda, and Y. Hirata, "Intermodulation interference-minimum frequency assignment for satellite SCPC systems," *IEEE Trans. Commun.*, vol. 4, pp. 462–468, 1984.
21. S. Salcedo-Sanz and C. Bousoño-Calzón, "A hybrid neural-genetic algorithm for frequency assignment optimization in satellite communications," in *Proc. of the 5th International Conference of Optimization, Techniques and Applications*, ICOTA '01, Hong-Kong, 2001.
22. 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.