A Hybrid Hopfield Network-Simulated Annealing Approach for Frequency Assignment in Satellite Communications Systems

Sancho Salcedo-Sanz, *Member, IEEE*, Ricardo Santiago-Mozos, *Student Member, IEEE*, and Carlos Bousoño-Calzón, *Member, IEEE*

***Abstract*—A hybrid Hopfield network-simulated annealing algorithm (HopSA) is presented for the frequency assignment problem (FAP) in satellite communications. The goal of this NP-complete problem is minimizing the cochannel interference between satellite communication systems by rearranging the frequency assignment, for the systems can accommodate the increasing demands. The HopSA algorithm consists of a fast digital Hopfield neural network which manages the problem constraints hybridized with a simulated annealing which improves the quality of the solutions obtained.**

We analyze the problem and its formulation, describing and discussing the HopSA algorithm and solving a set of benchmark problems. The results obtained are compared with other existing approaches in order to show the performance of the HopSA approach.

***Index Terms*—Combinatorial optimization, frequency assignment, Hopfield neural networks, satellite communications, simulated annealing.**

I. Introduction

In SATELLITE communication systems, the reduction of the cochannel interference has arisen as one major factor for determining system design [1]. 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 [2]. To cope with interference reduction, the rearrangement of frequency assignments is considered an effective measure in practical situations [2].

Frequency rearrangement can be formulated as a combinatorial optimization problem known as frequency assignment problem (FAP) for satellite communications. FAP belongs to a class of optimization problems with constraints, in which a goal function must be optimized and a set of constraints have to be fulfilled for a solution to be feasible.

In this kind of problems, scalability is a major factor of the algorithm design, due to the poor performance of nonscalable algorithms when the size of the problem grows. In this context, FAP has been solved before by using emerging methods such as branch and bound [2] and Hopfield neural networks [1], [3]. Both techniques suffer from lack of scalability, which leads to poor quality solutions in large, difficult problems.

Manuscript received July 11, 2003; revised October 27, 2003.

The authors are with the Department of Signal Theory and Communications, Universidad Carlos III de Madrid, Leganés-Madrid 28911, Spain.

Digital Object Identifier 10.1109/TSMCB.2003.821458

This paper follows the problem formulation on the FAP given by Mizuike *et al.* [2] in 1986 and Funabiki *et al.* [1] in 1997. Fig. 1 illustrates an example of two systems which suffer from interference when they operate in the same frequency band. Fig. 2 shows an example of the cochannel interference model used in [1] and [2]. The communications are assumed to be operated in the frequency band between F_a and F_b. In this example, three and four carriers are utilized in each satellite system, respectively. The cochannel interference is evaluated by each pair of carriers using the same frequency. Thus, in order to secure the communication quality of all carriers, the largest interference must be minimized among all pairs. In addition, the total interference should be also minimized for the improvement of the overall communication quality [1].

In this paper we propose a hybrid Hopfield network-simulated annealing (HopSA) for solving the FAP, in which a fast digital Hopfield neural network (HNN) [4] manages the problem's constraints and a simulated annealing algorithm (SA) [5] searches for high-quality solutions. We show that due to the separated management of constraints and goal function our algorithm is more scalable and achieves better results than existing algorithms for the FAP.

The rest of the paper is organized as follows: in the next section we define and analyze the FAP. In Section III the hybrid Hopfield network-Simulated annealing algorithm is described, by studying the Hopfield neural network and the SA which form it. Section IV shows the performance of the HopSA algorithm, by solving a set of benchmark problems and comparing the results obtained with previous algorithms for the FAP. In this section some discussion about the computational cost of the HopSA algorithm is also provided. Finally, Section V ends the paper with some concluding remarks.

II. Problem Formulation

Given two adjacent satellite systems (Fig. 1), FAP consists in reducing the inter-system cochannel interference by rearranging the frequency assignment on carriers in system #2 (M segments, N carriers), while the assignment in system #1 (M segments) remains fixed. Because each carrier usually occupying a different length in a frequency band, Mizuike *et al.* introduced the segmentation of carriers so that every carrier can be described by a collection of consecutive unit segments. The interference between two M-segment systems is described by a $M \times M$ interference matrix E, in which the ijth element e_{ij} stands for

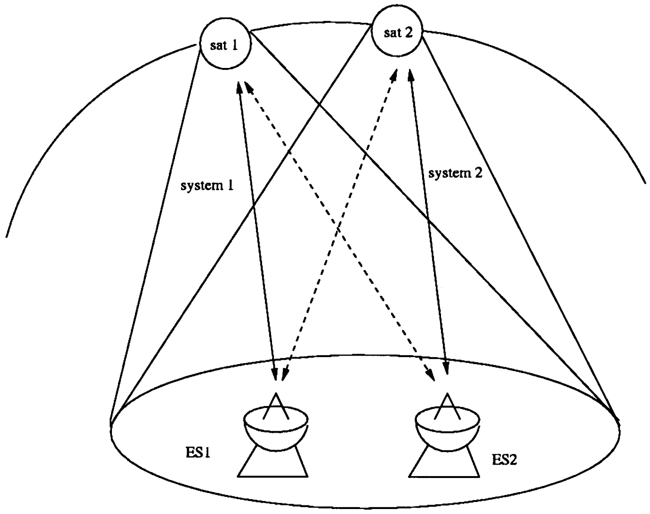

Fig. 1. Outline of cochannel interference.

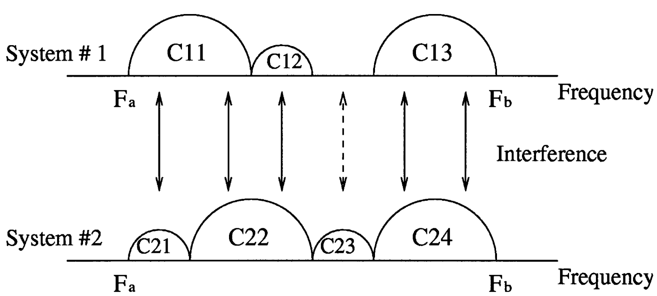

Fig. 2. Cochannel interference model of the system in Fig. 1.

the cochannel interference when segment $\#i$ in system #2 uses a common frequency with segment $\#j$ in system #1.

The constraints of the FAP are the following.

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

C2) Every segment in system #1 can be assigned to 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.

In this paper, we use a mixed representation to solve the problem, which was introduced in [2] and also used in [1]. This representation involves two matrices for completely representing the problem: first an $M \times M$ matrix F is defined, 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. This matrix is called the *reassignment matrix*, and it is used to calculate the objective function associated to the problem. Matrix F can be seen as the most intuitive representation of the problem, where every segment in system #2 is directly assigned to a segment in system #1. However, it is difficult to manage problem's constraint C3 using this representation. Thus, another matrix $\tilde{F}$, $(N \times M)$, 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.[1] The fulfilment of problem's constraints can be better managed using as problem representation matrix $\tilde{F}$. Note that between every two carriers there must be a minimum separation in segments for constraints C1, C2 and C3 to be fulfilled. Thus, a matrix C, $(N \times N)$, can be defined, 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 FAP as follows:

Achieve an assignment $\tilde{F}$ such that

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

subject to

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

and in such a way that the assignment $\tilde{F}$ fulfils the constraints in C: 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.

Note the mixed representation of the problem: matrix F is used for the calculation of the objective function, whereas matrix $\tilde{F}$ is used to perform the reassignment of carriers between the two systems, fulfilling the problem constraints.

A. Example

An example of a small FAP 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. 3. Note that both systems have $M = 6$ segments, and system #2 has $N = 4$ carriers. The FAP consists in reassigning carriers of system #2, whereas system #1 is fixed. Fig. 4 illustrates the segmentation of the systems, and a possible reassignment when interference matrix in Fig. 3 is considered. Fig. 5 shows this assignment in the mixed representation we use to solve the problem. Fig. 5(a) shows matrix $\tilde{F}$. Note that this matrix fulfils the constraint in (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. 5(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.

III. Proposed Approach

The algorithm we propose for solving the FAP consists of a hybrid global-local scheme, where a local procedure (Hopfield neural network) manages the fulfilment of FAP's constraints, and a global algorithm (Simulated Annealing) looks for the minimization of the objective function.

A. Hopfield Neural Network

The Hopfield network we use as local algorithm for solving the FAP constraints belongs to a class of digital Hopfield net-

[1]This new matrix $\tilde{F}$ can be calculated from F in a straightforward manner, knowing the carrier's length.

works [4] where the neurons can only take values 1 or 0. The dynamics of this network depends on the matrix C, and, of course, on the initial state of the neurons, see [4] 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 then be described 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 updating rule is described by

$$\tilde{f}_{ij}(t) = isgn\left(\sum_{\substack{p=1\\p\neq i}}^{N} \sum_{\substack{q=max(1,c_{i,p+1})\\q\neq j}}^{min(M,j+c_{i,p})} \tilde{f}_{pq}\right) \forall i,j \qquad (3)$$

where the $isgn$ operator is defined by

$$isgn(a) = \begin{cases} 0, & if\ a > 0 \\ 1, & otherwise. \end{cases}$$

Note that the updating rule only takes into account neurons $\tilde{f}_{pq}$ equal to 1 and within a distance of c_{ip} in columns of the element $\tilde{f}_{ij}$ being updated. Note also that in this updating rule, 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 is increasing. Let $\pi(i)$ be a random permutation of $i = 1,2,\ldots,N$. The new updating rule of the HNN results

$$\tilde{f}_{\pi(i)j}(t) = isgn\left(\sum_{\substack{p=1\\p\neq \pi(i)}}^{N} \sum_{\substack{q=max\left(1,c_{\pi(i),p+1}\right)\\q\neq j}}^{min\left(M,j+c_{\pi(i),p}\right)} \tilde{f}_{pq}\right) \forall i,j. \qquad (4)$$

The resulting updating rule runs over the rows of $\tilde{F}$ in the order given by the permutation $\pi(i)$, but the columns are updated in natural order $j = 1,2,\ldots,M$. A *cycle* is defined 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 during the cycle.[2] The final state of the HNN dynamics is a potential solution for the FAP, which fulfils the constraints of the matrix C. Note, however, that the solution found may be unfeasible if all the carriers are not assigned.

[2]The convergence of the neural network presented in this section only takes a few cycles, see Section IV-C for a detailed analysis of its convergence in a benchmark problem.

		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	30	45	0	35	35
C24	S25	45	5	25	0	50	*
	S26	*	45	25	0	10	50

Fig. 3. Example of interference matrix for the system in Fig. 1.

B. Simulated Annealing

SA has been widely applied to solve combinatorial optimization problems [5]–[8]. It is inspired by the physical process of heating a substance and then cooling it slowly, until a strong crystalline structure is obtained. This process is simulated by lowering an initial temperature by slow stages until the system reaches to an equilibrium point, and no more changes occur. Each stage of the process consists in changing the configuration several times, until a thermal equilibrium is reached, and a new stage starts, with a lower temperature. The solution of the problem is the configuration obtained in the last stage. In the standard SA, the changes in the configuration are performed in the following way: A new configuration is built by a random displacement of the current one. If the new configuration is better, then it replaces the current one, and if not, it may replace the current one probabilistically. This probability of replacement is high in the beginning of the algorithm, and decreases in every stage. This procedure allows the system to move toward the best configuration. Although SA is not guaranteed to find the global optima, it is still better than others algorithms in escaping from local optima. The solution found by SA can be considered a "good enough" solution, but it is not guaranteed to be the best.

The approach in this paper considers the mixing of a SA and the Hopfield neural network presented in Section III-A. The main idea behind this is that configurations involved in the SA are feasible solutions for the FAP. The SA will then seek for the best feasible solution with respect to a given objective function. There have been similar previous approaches to other optimization problems using a hybrid model SA-HNN [9], and SA hybridized with other optimization procedures [10].

The most important parts in a SA algorithm are: the objective function to be minimized during the process, the chosen representation for solutions and the mutation or configuration change operator. We present these three characteristics in the next subsections.

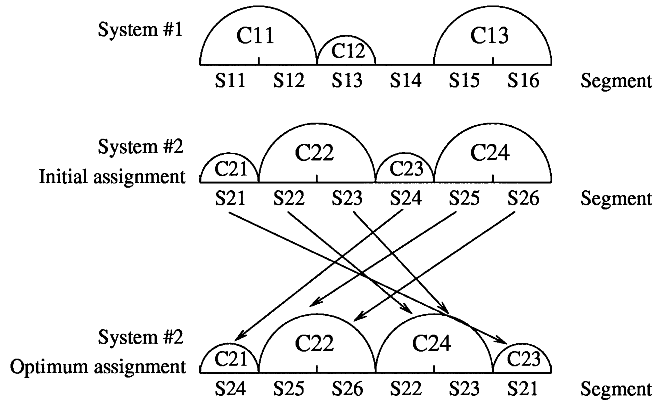

Fig. 4. Segmentation of the system defined by Figs. 1, 2 and 3.

C. Objective Functions for the FAP

We consider three different basic objective functions for the FAP. First, we want the solution to minimize the maximum peak of interference between the systems (Largest interference), so the first objective function will be

$$\gamma_1(E,F) = max(e_{ij} \cdot f_{ij}) \quad \forall i,j. \tag{5}$$

Note that the matrix involved in this calculation is F, which can be obtained from $\tilde{F}$ following the process represented in Fig. 5.

The second objective function of the FAP requires that the total interference of the systems to be minimum, so

$$\gamma_2(E,F) = \sum_{i=1}^{M}\sum_{j=1}^{M} e_{ij} \cdot f_{ij}. \tag{6}$$

Finally, we also consider a third function which takes into account both γ_1 and γ_2

$$\gamma_3(E,F) = \alpha \cdot \gamma_1(E,F) + \beta \cdot \gamma_2(E,F). \tag{7}$$

Note that γ_1 will produce solutions with a very good value of maximum interference, but the value of the total interference may be high. On the contrary, γ_2 minimizes the value of the total interference, but there may be large peaks of interference. Function γ_3 allows a balanced situation, where both the values of maximum and total interference can be controled.

D. Problem Representation

We encode every possible solution of the problem as the binary matrix $\tilde{F}$, $N \times M$. We obtain a feasible solution by running the HNN over an unfeasible $\tilde{F}$ randomly generated (at the beginning of the algorithm) or generated by the mutation operator. Only feasible solutions are considered: if the solution obtained by the HNN is not feasible due to every carrier not having been assigned, the solution is discarded and the mutation operator is reapplied until a feasible solution is obtained by the HNN.

E. Mutation Operator

In order to obtain a new configuration, N_p bits of the binary matrix $\tilde{F}$ are flipped, passing from 1 to 0 or vice-versa. The N_p bits to be changed are randomly chosen among the $N \times M$ possible.

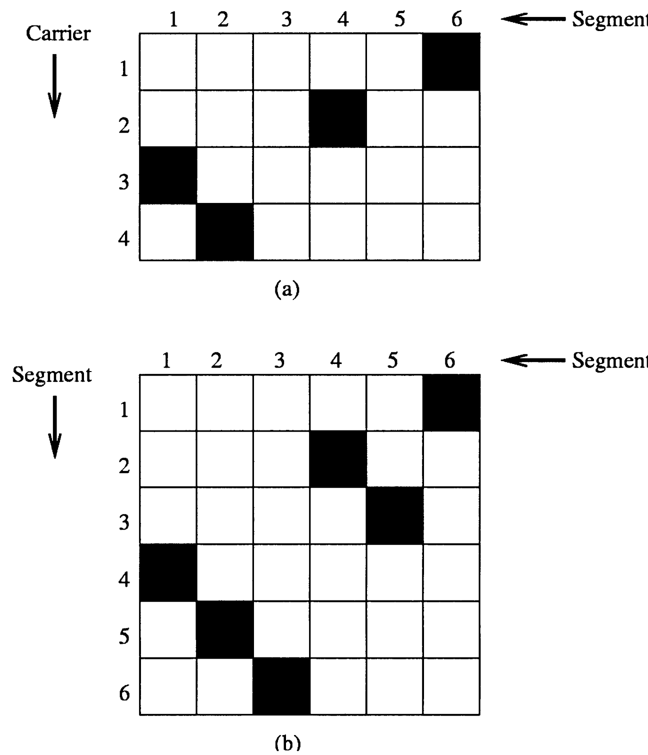

Fig. 5. (a) Example of matrix $\tilde{F}$ for the interference matrix of Fig. 2 (shaded squares represent "1s" and whites squares "0s"). (b) Matrix F obtained from $\tilde{F}$.

TABLE I
MAIN FEATURES OF THE SET OF BENCHMARK PROBLEMS

Problem #	Carriers	segments	Range of carrier	Range of interfer.
1	10	32	1-8	1-100
2	10	32	1-8	1-1000
3	18	60	1-8	1-50
4	30	100	1-8	1-100
5	15	50	1-7	1-1000
6	50	200	1-8	1-1000

F. The Complete Algorithm

The complete algorithm for the FAP is formed by mixing the SA and the HNN, and performs in the following way.

HopSA Algorithm:
$k = 0$;
$T = T_0$;
Initialize a potential solution at random;
do
Run the HNN to obtain $\tilde{F}$;
} **until** (a feasible solution is obtained)
($\tilde{F} \rightarrow F$): **evaluate**(F, $\gamma_i(E,F)$);
(Simulated Annealing)
repeat

0	56	22	75	34	9	53	44	67	12	41	4	83	39	30	29	24	28	60	80	47	63	64	86	0	76	91	45	99	*	*	*
*	84	68	46	84	47	47	13	19	83	49	19	33	35	64	56	30	42	68	43	22	55	94	82	0	61	23	17	84	29	*	*
*	*	68	51	56	95	21	34	65	6	22	26	37	79	5	58	98	55	95	84	92	29	47	68	0	83	53	26	21	81	72	*
*	*	*	61	24	84	2	27	51	27	84	45	37	32	94	11	93	95	87	40	5	48	33	19	0	55	15	76	99	39	92	35
0	65	23	73	45	22	92	8	3	78	67	29	15	68	84	96	25	68	28	54	16	64	50	34	0	9	65	14	40	14	45	*
*	2	14	44	60	63	53	86	24	100	53	55	17	29	45	86	83	32	8	27	71	46	12	56	0	71	79	23	15	8	87	10
0	12	84	8	30	22	89	90	80	77	73	81	54	18	3	29	64	54	26	35	71	66	7	18	0	58	36	48	43	*	*	*
*	3	41	28	36	40	36	89	9	10	91	41	61	99	32	51	40	59	12	52	50	67	19	66	0	10	66	32	73	4	*	*
*	*	67	55	20	38	26	57	73	87	86	49	38	99	30	71	44	35	19	3	37	29	33	13	0	55	49	15	19	7	78	*
*	*	*	17	78	24	82	5	92	96	82	69	13	53	89	67	85	80	55	6	56	97	3	43	0	80	15	69	46	46	50	17
0	89	97	40	21	58	23	15	53	73	14	70	53	67	64	62	64	61	69	68	37	33	20	100	0	96	81	9	79	*	*	*
*	87	18	69	71	93	47	43	47	60	50	45	90	26	43	9	50	21	57	57	31	80	17	64	0	16	100	74	91	45	*	*
*	*	15	94	32	77	4	34	12	20	46	7	89	4	87	93	41	75	68	25	31	67	68	16	0	70	21	96	12	35	62	*
*	*	*	24	49	47	83	62	63	71	21	92	98	46	30	29	98	19	56	89	22	32	94	51	0	20	52	63	77	2	17	98
0	16	28	51	25	27	61	1	56	10	42	25	48	25	65	45	49	43	34	3	38	5	25	67	0	*	*	*	*	*	*	*
*	62	29	92	15	95	64	68	87	57	75	42	76	38	88	81	50	80	17	96	43	68	61	59	0	88	*	*	*	*	*	*
*	*	41	33	91	16	6	18	90	37	67	11	8	55	96	64	64	52	65	32	96	14	100	34	0	79	33	*	*	*	*	*
*	*	*	7	36	69	80	15	67	72	38	81	98	80	95	52	20	55	35	42	24	95	30	78	0	93	53	76	*	*	*	*
*	*	*	*	68	86	93	84	47	36	23	78	2	92	20	5	30	49	92	73	1	16	89	64	0	60	71	99	7	*	*	*
*	*	*	*	*	71	50	73	23	57	67	82	32	58	93	6	32	91	81	33	16	50	99	87	0	25	95	100	60	98	*	*
*	*	*	*	*	*	44	97	92	1	19	41	88	34	38	99	59	8	16	51	74	18	3	1	0	59	52	9	76	83	19	*
*	*	*	*	*	*	*	94	69	86	63	82	20	77	35	48	44	38	35	67	22	33	72	51	0	95	51	92	29	77	12	66
0	93	59	3	91	66	39	48	100	58	21	1	4	65	66	12	80	14	45	91	10	68	75	78	0	95	52	7	34	6	42	2
0	63	19	83	51	8	11	61	25	57	100	94	59	10	92	87	61	20	37	71	49	74	47	65	0	3	59	71	84	40	40	*
*	82	78	59	57	28	40	22	73	76	49	55	39	35	33	28	65	31	74	7	51	95	61	86	0	12	68	63	87	83	31	73
0	32	91	27	28	3	20	59	14	34	27	35	70	49	42	64	41	95	56	16	28	20	28	5	0	89	79	17	79	94	58	*
*	90	91	11	14	29	89	96	83	52	89	20	15	92	55	39	74	91	35	98	75	75	17	24	0	55	98	63	28	61	45	92
0	42	25	86	30	52	16	92	64	88	21	87	94	37	18	52	73	28	21	55	42	35	55	57	0	25	27	31	48	12	81	13
0	46	95	40	100	66	15	6	13	32	9	20	85	96	25	78	80	85	80	36	80	44	39	89	0	33	15	22	97	85	*	*
*	43	73	60	22	50	48	10	30	54	42	66	32	23	74	33	97	25	44	41	64	42	45	38	0	55	51	11	61	6	76	*
*	*	6	26	63	44	97	9	49	74	28	28	96	12	10	46	60	69	30	35	87	38	86	69	0	69	66	23	82	2	15	23
0	6	90	25	11	69	10	29	2	58	51	1	3	65	46	47	91	3	34	34	43	85	47	19	0	29	26	93	11	47	46	70

(a)

0	429	219	546	936	811	45	603	230	165	833	281	402	863	594	274	674	509	559	946	206	333	644	54	0	251	366	781	42	*	*	*
*	834	917	287	462	675	381	821	528	257	202	808	717	856	806	346	210	603	239	833	19	266	504	264	0	443	364	314	932	103	*	*
*	*	41	478	326	183	212	549	145	760	984	384	917	349	990	649	222	726	448	216	913	74	29	775	0	286	142	678	831	957	247	*
*	*	*	637	496	339	206	85	644	139	395	138	445	579	903	19	135	432	596	623	529	859	234	1000	0	550	166	287	444	851	830	316
0	265	709	452	119	560	574	704	783	224	143	80	867	95	736	112	836	73	291	217	887	893	794	764	0	806	534	177	23	283	639	*
*	344	705	655	68	179	808	577	357	476	425	502	175	106	636	30	409	273	354	392	352	888	87	100	0	401	607	988	313	502	399	590
0	518	493	663	190	654	492	98	652	319	729	33	343	175	609	958	663	545	193	378	257	563	730	866	0	490	379	982	300	*	*	*
*	22	362	288	321	122	959	548	482	150	185	67	772	925	729	279	483	167	774	237	811	47	911	959	0	686	123	523	883	661	*	*
*	*	560	965	28	429	565	793	141	686	452	458	499	170	318	885	970	398	204	572	223	142	526	721	0	691	524	665	633	614	635	*
*	*	*	325	192	998	296	953	803	85	782	86	904	334	702	869	178	681	708	921	468	422	464	593	0	449	892	252	422	82	500	208
0	566	307	797	168	637	298	159	996	733	905	441	361	926	388	426	145	938	311	765	33	340	113	192	0	62	885	40	865	*	*	*
*	245	303	663	678	153	819	695	207	952	118	349	616	268	735	596	293	231	487	463	827	613	628	709	0	915	971	458	294	288	*	*
*	*	213	311	937	504	45	199	513	629	761	19	164	979	206	157	273	503	242	268	603	9	557	98	0	250	474	248	646	446	485	*
*	*	*	48	247	663	281	353	798	195	308	468	136	358	503	611	287	917	142	949	638	677	865	566	0	418	760	376	878	804	494	792
0	960	425	677	605	590	164	877	226	518	925	654	577	145	980	932	404	328	901	155	56	171	898	370	0	*	*	*	*	*	*	*
*	313	955	139	999	338	372	785	323	117	942	725	573	598	71	80	810	65	355	689	791	144	666	713	0	806	*	*	*	*	*	*
*	*	236	947	298	771	965	925	528	760	982	734	339	359	37	552	795	223	675	857	925	835	461	351	0	776	15	*	*	*	*	*
*	*	*	151	886	637	682	593	702	984	69	234	844	554	482	175	674	124	445	701	491	725	221	564	0	819	312	579	*	*	*	*
*	*	*	*	984	861	640	245	942	991	596	977	765	744	730	744	884	272	932	576	433	968	911	5	0	404	872	332	377	*	*	*
*	*	*	*	*	3	587	582	514	90	756	828	187	16	816	522	210	662	230	397	464	936	688	851	0	817	193	767	348	475	*	*
*	*	*	*	*	*	375	947	508	920	282	762	117	652	187	924	585	28	902	656	381	473	996	574	0	3	31	642	657	117	792	*
*	*	*	*	*	*	*	949	512	62	335	56	419	17	899	626	185	825	506	71	102	603	246	565	0	937	587	96	920	868	606	192
0	706	481	737	465	648	221	24	586	707	831	394	391	257	727	815	777	589	567	274	392	220	721	760	0	542	383	342	323	271	644	310
0	65	507	945	601	851	160	114	676	629	862	830	306	729	215	311	901	263	274	25	196	586	135	338	0	341	691	488	413	632	410	*
*	158	278	191	272	42	32	884	790	167	785	939	572	726	710	896	905	109	133	290	883	951	823	511	0	196	149	916	546	384	734	910
0	979	749	742	163	231	37	547	976	623	280	606	449	916	902	412	241	852	300	512	155	915	637	875	0	866	933	365	180	516	730	*
*	542	413	341	543	563	418	242	261	307	471	114	801	124	334	456	950	399	994	659	143	51	125	311	0	194	844	951	241	779	798	843
0	354	791	436	383	881	99	328	144	211	961	847	620	703	750	425	723	594	220	495	471	98	300	538	0	657	317	221	736	329	963	250
0	404	632	418	441	375	964	542	506	101	608	58	757	886	391	694	52	254	626	436	962	90	488	740	0	278	957	114	98	456	*	*
*	348	867	372	859	686	837	683	652	224	815	12	141	229	534	60	894	247	110	684	96	289	20	580	0	3	23	648	460	467	909	*
*	*	426	842	461	185	275	285	253	929	102	466	452	697	351	558	255	36	189	203	371	190	528	725	0	227	716	721	493	189	114	960
0	236	179	133	321	964	406	115	871	696	364	415	120	485	473	302	830	719	582	517	5	605	714	952	0	215	182	349	75	611	803	740

(b)

Fig. 6. Interference matrices of problems: (a) #1 and (b) #2. Symbol * stands for an infinity value of interference.

for $j = 0$ to ξ
$\tilde{F}_{mut} = \mathbf{mutate}(\tilde{F})$
do{
Run the HNN to obtain $\tilde{F}$;
} **until** (a feasible solution is obtained)
$(\tilde{F}_{mut} \rightarrow F_{mut})$: **evaluate**$(F_{mut},\ \gamma_i)$;
if$((\gamma_i(E, F_{mut}) < \gamma_i(E, F)))$ **OR** $(random(0,1) < e^{(-a/T)}))$ then
$\tilde{F} = \tilde{F}_{mut}$;
endif
endfor
$T = f_T(T_0, k)$;
$k = k + 1$;
until$(T < T_{min})$;

where k counts the number of iterations performed; T keeps the current temperature; T_0 is the initial temperature; T_{min} is the minimum temperature to be reached; $\tilde{F}$ stands for the current configuration and $\tilde{F}_{mut}$ for the new configuration after the mutation operator is applied; γ_i represents any of the three objective function considered (see Section III-C); ξ is the number of changes performed with a given temperature T; f_T is the freezer function; and a is a previously fixed constant. Parameter a and the initial temperature T_0 are calculated in order to have an initial acceptance probability equal to 0.8, which is the value usually used. The freezer function is defined as

$$f_T = \frac{T_0}{1+k}. \quad (8)$$

The minimum temperature T_{min} is calculated on the basis of the desired number of iterations as

$$T_{min} = f_T(T_0, numIt). \quad (9)$$

IV. Simulations and Results

In order to test the performance of the HopSA algorithm, a set of benchmark problems have been selected. Table I summarizes the main characteristics of the benchmark problems considered. Problems #1 and #2 are taken from Funabiki *et al.* [1], where these are called problem #4 and #5, respectively. Fig. 6 shows the interference matrices of these problems. The interference matrices as well as the carrier lengths (in segments) of all benchmark problems tackled have been attached to this paper in the electronic submission.

Algorithm in [1] has been programmed following the indications in that paper, in order to perform comparison in all test problems.

The parameters of the HopSA algorithm remain unchanged in all simulations performed: the number of iterations $(numIt)$ of the SA was fixed to 300 with $\xi = 50$. Note that tuning the SA parameters is important for ensuring the convergence to a good enough solution of the problem in a reasonable time. Election of $numIt$ or ξ too small may produce that the SA stops in a suboptimal solution. On the other hand, if these parameters are chosen too large, the computational time of the algorithm would be high. The parameters chosen in this paper seem to be a good election in the problems considered. In mutation operator, the value of N_p was fixed to 20. This value has demonstrated to be enough to perform a wide search over the space of matrices $\tilde{F}$. Using function γ_3, the best results were obtained by fixing $\alpha = 0.7$ and $\beta = 0.3$.

A. Results

Table II shows the results obtained by the HopSA algorithm and a comparison with other algorithm results for the benchmark problems considered in largest interference (function γ_1). HopSA algorithm equals or improves the results of other existing methods. These results show that the HopSA algorithm performs well in difficult problems, achieving better results and being much more scalable than existing algorithms.

Table III shows the results for the total interference in the assignments (objective function γ_2). Our algorithm achieves again better results than existing algorithms and the differences are getting larger as the problems become more complicated.

TABLE II
Comparison of the Results Obtained by the HopSA Algorithm With Previous Approaches (Largest Interference, γ_1). Problems #1 to #6

Problem #	Mizuike [2]	Funabiki [1]	HopSA
1	67	64	64
2	803	640	640
3	-	49	41
4	-	100	96
5	-	919	672
6	-	1000	929

TABLE III
Comparison of the Results Obtained by the HopSA Algorithm With Previous Approaches (Total Interference, γ_2). Problems #1 to #6

Problem #	Mizuike [2]	Funabiki [1]	HopSA
1	929	880	792
2	10330	8693	6851
3	-	1218	933
4	-	4633	3745
5	-	16192	11568
6	-	70355	59431

TABLE IV
Results of the HopSA Algorithm With γ_3. Problems #1 to #6

Problem #	γ_3	Total (γ_2)	Largest (γ_1)
1	324.6	886	84
2	2627.2	6851	817
3	334.2	1002	48
4	1296.5	4093	98
5	4584.9	13554	741
6	19091.0	61330	988

Table IV shows the results obtained by our algorithm when function γ_3 is used. Note that using this function there is a balance between the maximum interference and the total interference of the system. In this table we display the values of functions γ_1 and γ_2 associated to the solution obtained using as objective function γ_3.

Fig. 7(a)–(c) shows the best solutions obtained by the HopSA algorithm in problem #1 using objective functions γ_1, γ_2 and γ_3, respectively. Note how different are the solutions achieved by the HopSA algorithm when different objective functions are used.

B. Discussion

We have shown that the HopSA algorithm outperforms existing algorithms for the FAP. However, a more detailed analysis may give some insight about how the HopSA algorithm performs. First, Note that the HopSA algorithm achieves better

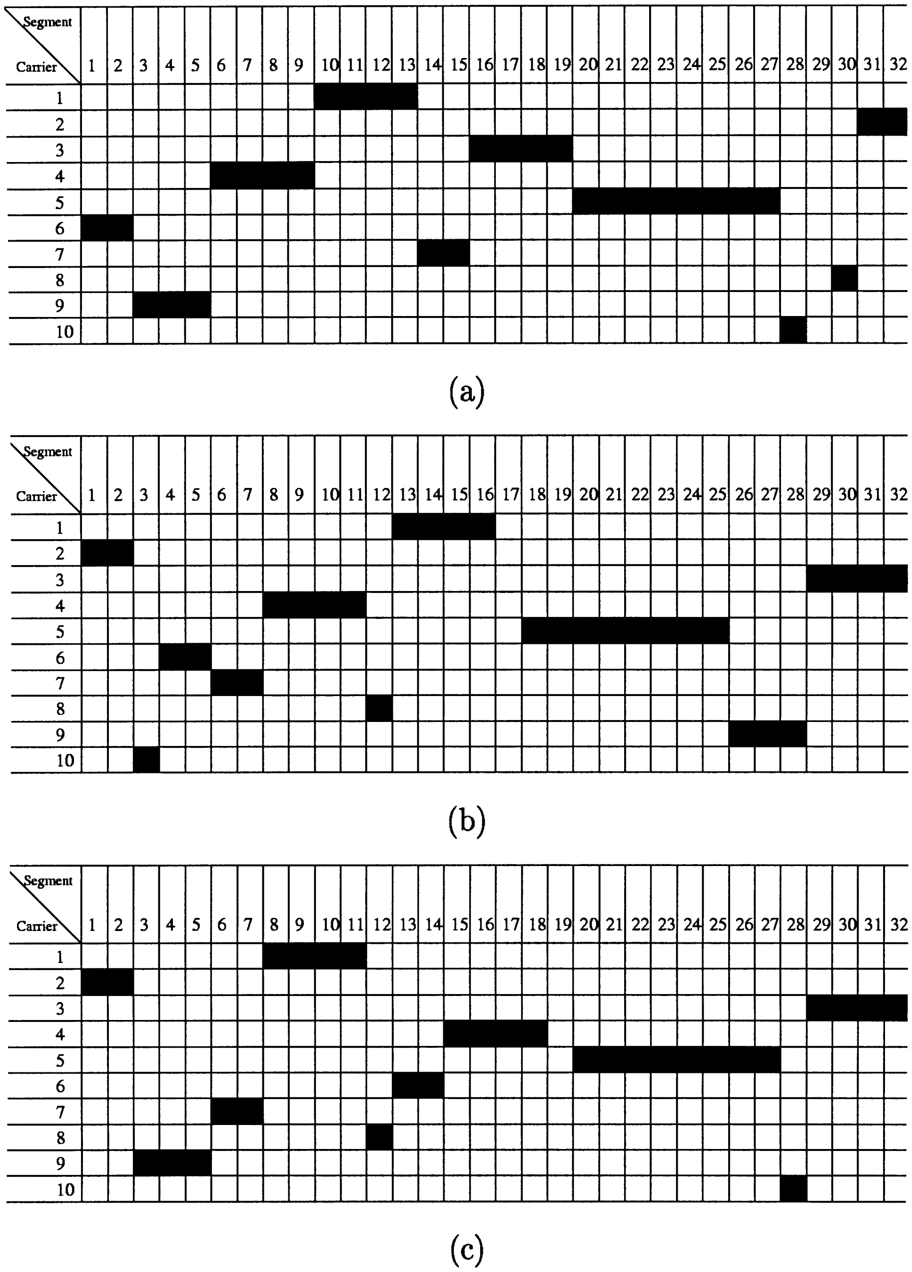

Fig. 7. Best assignment achieved by HopSA algorithm in Problem #1 using objective functions γ_1 (a), γ_2 (b), and γ_3 (c). All the segments in each carrier have been depicted for a clearer explanation, instead of only the first segment of each carrier, as constraint in (2) imposes.

assignments with respect to objective functions γ_1 and γ_2 compared to previous approaches. We interpret that the good performance of the HopSA algorithm is due to the separated management of problem's constraints and optimization function. Note that this makes our algorithm more scalable than other existing algorithms which do not manage separately the problem's constraints and the optimization process (better solutions are found when the size of problem grows).

Second, it is easy to see that our algorithm may be used with different objective functions in a straightforward manner. Depending on the necessities of the satellite systems designer, it may be better to obtain solutions with low total interference or low largest interference. Obtaining a balance between total interference and largest interference is usually the most appealing option. In this sense, objective function γ_3 should be used in the design of the systems. The results obtained in this paper show that the HopSA algorithm using objective function γ_3 provides high quality solutions both in terms of the largest and total interference, as can be seen in Table II. The case of problem #2 is very interesting, since the same solution (shown in Fig. 8) has been achieved with the objective functions γ_2 and γ_3.

C. *Some Comments About the Computational Cost of the HopSA Algorithm*

The increasing of computational cost is the main drawback when using a hybrid algorithm in a combinatorial optimization problem such as the FAP. Thus, the design of the local and global algorithms to be mixed must be as accurate as possible, taking

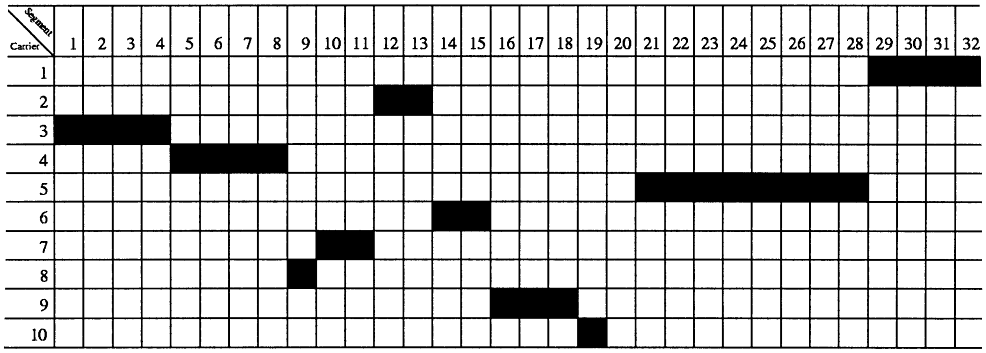

Fig. 8. Best assignment achieved by HopSA algorithm in Problem #2 using objective function γ_2 and γ_3.

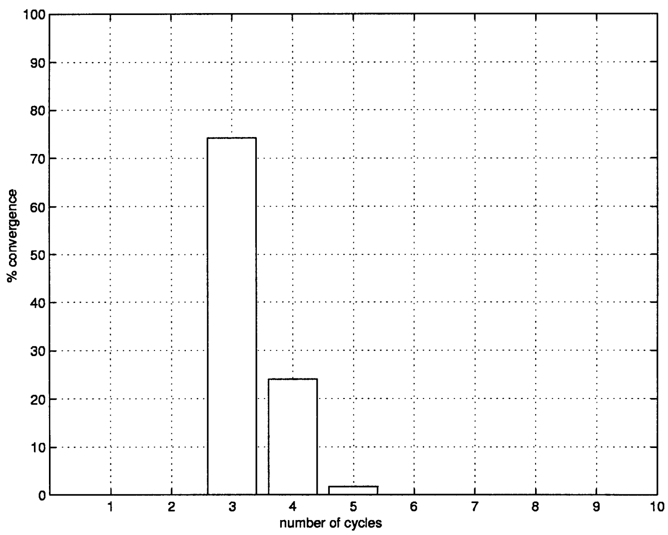

Fig. 9. Number of cycles needed for the HNN convergence, in problem #5.

into account the computational cost as a primary factor. The design of the HopSA algorithm follows the hints bellow: first, the SA algorithm only evolves one potential solution instead of a whole family of potential solutions considered in other techniques (genetic algorithms for example). Thus SA is a more appropriate technique for finding good solutions for the FAP quickly.

Second, the HNN used is a fast digital network, with very good properties of convergence. We found that the HNN achieves a feasible solution in the majority of runs, starting from an unfeasible one. In addition, the speed of convergence of the network is very good: As an example, Fig. 9 shows that 75% of the HNN's launched for solving problem #2 converged in three cycles.[3] Over 23% of the networks launched converged in four cycles, and only about 2% of the networks run converged in five cycles. The new updating rule introduced in Section III-A, does not modify these values, because it only changes the order of updating, not the structure of the algorithm.

In spite of the accurate design of the HopSA algorithm, focused on reducing its complexity, it is expected that its computational cost to be higher than existing approaches for the FAP. We have performed a comparison of computational time between our HopSA algorithm and the gradual neural network programmed following [1], in our simulation platform (a SUN SPARK 2/480 MHz.), for Problems #1 and #2. Funabiki's algorithm needed about 1 minute for solving Problems #1 and also about a minute for Problem #2, whereas HopSA algorithm needed about 4 minutes for solving each problem.

[3]Recall that a cycle is defined as the updating of all the neurons in the Hopfield network (see Section III-A).

V. CONCLUSION

In this paper, a hybrid Hopfield neural network-simulated annealing algorithm (HopSA) 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 Simulated Annealing algorithm which improves the solution obtained from the network. This approach for the FAP is more scalable than previous algorithms due to the separated 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 HopSA and the other algorithms.

REFERENCES

[1] N. Funabiki and S. Nishikawa, "A gradual neural-network approach for frequency assignment in satellite communication systems," *IEEE Trans. Neural Networks*, vol. 8, pp. 1359–1370, Nov. 1997.

[2] T. Mizuike and Y. Ito, "Optimization of frequency assignment," *IEEE Trans. Commun.*, vol. 37, pp. 1031–1041, Oct. 1989.

[3] T. Kurokawa and S. Kozuka, "A proposal of neural network for the optimum frequency assignment problem," *Trans. IEICE*, vol. J76-B-II, no. 10, pp. 811–819, 1993.

[4] 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, Nov. 1992.

[5] S. Kirpatrick, C. D. Gerlatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, pp. 671–680, 1983.

[6] S. Kirpatrick, "Optimization by simulated annealing-quantitative studies," *J. Stat. Phys.*, vol. 34, pp. 975–986, 1984.

[7] J. González, I. Rojas, H. Pomares, M. Salmerón, and J. J. Merelo, "Web newspaper layout optimization using simulated annealing," *IEEE Trans. Systems, Man Cybern. B*, vol. 32, pp. 686–691, Oct. 2002.

[8] G. Wang and N. Ansari, "Optimal broadcast scheduling in packet radio networks using mean field annealing," *IEEE J. Select. Areas Commun.*, vol. 15, pp. 250–259, Feb. 1997.

[9] 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.
[10] H. Kim, Y. Hayashi, and K. Nara, "An algorithm for thermal unit maintenance scheduling through combined use of GA, SA and TS," *IEEE Trans. Power Syst.*, vol. 12, pp. 329–335, Feb. 1997.
[11] 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 J. Select. Areas Commun.*, vol. 17, pp. 193–203, Feb. 1999.
[12] 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, pp. 470–482, Mar. 1995.
[13] C. Bousoño-Calzón and A. R. Figueiras-Vidal, "Emerging techniques for dynamic frequency assignment: merging genetic algorithms and neural networks," in *Proc. Inform. Syst. Technol. Symp.*, Aalborg, Denmark, 1998, pp. 12.1–12.5.
[14] S. Salcedo-Sanz, C. Bousoño-Calzón, and A. R. Figueiras-Vidal, "A mixed neural-genetic algorithm for the broadcast scheduling problem," *IEEE Trans. Wireless Commun.*, vol. 2, pp. 277–283, Feb. 2003.
[15] H. Okinaka, Y. Yasuda, and Y. Hirata, "Intermodulation interference-minimum frequency assignment for satellite SCPC systems," *IEEE Trans. Commun.*, vol. 4, pp. 462–468, Apr. 1984.
[16] K. C. Tan, Y. Li, D. J. Murray-Smith, and K. C. Sharman, "System identification and linearization using genetic algorithms with simulated annealing," in *Proc. 1st IEE/IEEE Int. Conf. Genetic Algorithms Eng. Syst.*, 1995, pp. 164–169.
[17] S. Salcedo-Sanz and C. Bousoño-Calzón, "A hybrid neural-genetic algorithm for frequency assignment optimization in satellite communications," in *Proc. . 5th Int. Conf. Optimization: Techn. Applicat.*, Hong-Kong, China, 2001.
[18] K. Smith, M. Palaniswami, and M. Krishnamoorthy, "Neural techniques for combinatorial optimization with applications," *IEEE Trans. Neural Networks*, vol. 9, pp. 1301–1318, Nov. 1998.
[19] J. Rose, W. Klebsch, and J. Wolf, "Temperature measurement and equilibrium dynamics of simulated annealing placements," *IEEE Trans. Computer Aided Design Integrated Circuits*, vol. 9, pp. 253–259, Mar. 1990.

Sancho Salcedo-Sanz (S'00–M'03) was born in Madrid, Spain, in 1974. He received the B.S. degree in physics from the Universidad Complutense de Madrid, Spain, in 1998, and the Ph.D degree in telecommunications engineering from the Universidad Carlos III de Madrid, Spain, in 2002.

He is currently a Research Fellow in the School of Computer Science, University of Birmingham, U.K. He has coauthored more than 15 international journals and conference papers in the field of genetic algorithms and hybrid algorithms. His current interests deal with optimization in communications, hybrid algorithms, and neural networks.

Ricardo Santiago-Mozos (S'01) was born in San Xenxo, Spain, in 1975. He received the B.S. degree in telecommunications engineering from the Universidad de Vigo, Spain, in 2001 and is currently pursuing the Ph.D degree in telecommunications engineering in the Department of Signal Theory and Communications, Universidad Carlos III de Madrid, Spain.

He is a Research Fellow with the Universidad Carlos III de Madrid. His research interests include optimization in biomedical problems, genetic algorithms, image processing, and signal processing.

Carlos Bousoño-Calzón (M'95) recived the B.S and Ph.D degrees in telecommunications engineering from the Universidad Politécnica de Madrid, Spain, in 1992 and 1996, respectively.

He is now an Associate Professor in the Department of Signal Theory and Communications, Universidad Carlos III de Madrid, Spain. His research interests are focused on optimization in communications, genetic algorithms, and neural networks. He has coauthored more than 30 international journals and conference papers in these areas.