A Mixed Neural-Genetic Algorithm for the Broadcast Scheduling Problem

Sancho Salcedo-Sanz, *Student Member, IEEE*, Carlos Bousoño-Calzón, *Member, IEEE*, and Aníbal R. Figueiras-Vidal, *Senior Member, IEEE*

***Abstract*—The broadcast scheduling problem (BSP) arises in frame design for packet radio networks (PRNs). The frame structure determines the main communication parameters: communication delay and throughput. The BSP is a combinatorial optimization problem which is known to be NP-hard. To solve it, we propose an algorithm with two main steps which naturally arise from the problem structure: the first one tackles the hardest contraints and the second one carries out the throughput optimization. This algorithm combines a Hopfield neural network for the constraints satisfaction and a genetic algorithm for achieving a maximal throughput. The algorithm performance is compared with that of existing algorithms in several benchmark cases; in all of them, our algorithm finds the optimum frame length and outperforms previous algorithms in the resulting throughput.**

***Index Terms*—Broadcast scheduling, genetic algorithms, Hopfield neural networks, packet radio networks (PRN).**

I. Introduction

In packet radio networks (PRNs), a set of geographically dispersed stations shares a single radio channel to send packets to each other by means of a time-division multiple-access (TDMA) protocol. Several stations can simultaneously use the channel if their distance is large enough to avoid interference; otherwise the stations have to transmit their packets in different time slots. If two stations wishing to communicate with each other are far apart, it may be necessary to relay the packets over multiple intermediate stations. Packet radio technology is a good candidate for high-speed wireless data communications, specially over wide area regions when wire connections are not cost effective. PRNs have been deployed since the 60s in military, commercial and academic environments, including Internet [1].

The broadcast scheduling problem (BSP) is defined as the scheduling of the transmisions of all the stations in a minimum number of time slots such that no collision among packets occurs [2]. The final arrangement of the station transmissions into their assigned time slots is called a frame. The frame structure is directly related to the main network performance measures. First, the *frame length* (i.e., the number of time slots needed for at least one transmission for every station) essentially determines the packet average delay. Secondly, for a fixed frame length, the channel utilization is determined by the number of simultaneous transmissions of noninterferring stations. We will refer to this number as *frame throughput* or, simply, throughput. Therefore, the BSP can be stated as to find the minimum length and maximum throughput for a PRN.

The BSP is a combinatorial optimization problem known to be NP-hard [2], [3]. Algorithmic solutions have been proposed based on different approaches such as graph theory (GT) [1], Hopfield neural networks (HNN) [4] or mean field annealing (MFA) [3]; the latter has been shown to outperform the previous ones. However, results reported in [3] do not achieve either minimum frame lengths or maximum throughput. Additionally, although the MFA technique requires much less computational effort than simulated annealing, the discrete-time simulation of the MFA equations may be also a time consuming process. Furthermore, MFA and HNN are quite efficient to take interference constraints into account but may not maximize the number of slots in a frame, as results in similar applications suggest.[1]

In order to jointly consider the interference constraints and maximum throughput, some hybrid algorithms combining hnn and genetic algorithms (GA) have proven better performance [7], [8].

Our approach to solve the BSP is to divide this problem in two subproblems. The first attempt is to find a minimum frame length able to satisfy interference constraints and to guarantee the transmission of every radio station once per frame. The second tackles the maximization of the throughput for a given frame length. The algorithm proposed has two stages corresponding to these two subproblems. For the first stage, a discrete HNN is used and for the second stage, we apply a combination of a HNN and a GA. The performance of the overall algorithm is tested for the problem instances given by [3]. For all these cases, we obtain optimum frame lengths and more throughput than the MFA algorithm. Furthermore, we also give a tight frame length lower bound, which is attained by the two stage algorithm in these instances.

The structure of this paper is as follows. Section II summarizes the formulation for the BSP dividing it in the above mentioned subproblems. In this section, we also discuss the lower bounds for the minimum frame length. The two-stage algorithm is presented and discussed in Section III. The test broadcasting problems and simulation results are presented and analyzed in Section IV. The main conclusions of this work close the paper.

Manuscript received February 22, 2001; revised April 17, 2002; accepted June 21, 2002 . The editor coordinating the review of this paper and approving it for publication is S. Tekinay. This work was supported in part by CYCIT under Grant TIC 1999-0216.

The authors are with the Department of Signal Theory and Communications, Universidad Carlos III de Madrid, Leganés-Madrid, 28911 Spain (e-mail: sancho@tsc.uc3m.es; cbousono@tsc.uc3m.es; arfv@ing.uc3m.es).

Digital Object Identifier 10.1109/TWC.2003.808967

[1]Consider, for example, the results in the frequency assingment problem [5], [6].

II. BSP Formulation and Discussion

Since a PRN operates a single radio channel, the main constraint for the BSP comes from the interference between stations. The interference pattern among radio stations, which is determined by the station physical location, radiation elements, and the radio propagation characteristics, can be represented as a graph: the nodes stand for the stations and there is an undirected arc between two nodes if the corresponding stations interfere with each other. Note that we are assuming a symmetric situation in which if station i interferes with station j, then station j will also interfere with station i. We use $G=(V,E)$ to denote this graph, where V is the set of nodes and E the set of arcs. In the case that $(i,j)\in E$, stations i and j are said to be *one hop apart*. If i and j are not one hop apart but there exists an intermediate station k such that $(i,k)\in E$ and $(k,j)\in E$, stations i and j are said to be *two hops apart*.

For a PRN to work correctly, there are operational constraints. Two stations i and j are not allowed to transmit at the same time if either of them are one hop apart, since their signals would interfer with each other, or if they are two hops apart, since their signals will collide at an intermediate station k. These interference constraints can be represented by a $N\times N$ binary matrix, F, where N is the number of stations in the system. The ith row, jth column element of F, f_{ij}, is one if and only if stations i and j are one or two hops apart and zero, otherwise. Matrix F defined this way is called the *system compatibility matrix* [3].

The frame specifies which stations are allowed to transmit in a definite time slot; so that the definition of the frame structure is the scheduling problem objective. The frame can be represented by a $M\times N$ binary matrix, S, where M is the number of time slots in the frame (i.e., the frame length) and N is the number of stations in the system, as previously defined. Element s_{ti} is one if station i is allowed to transmit in time slot t; if not, s_{ti} is zero. Note that two or more stations can simultaneously transmit if they do not interfere with each other. Furthermore, the more stations transmitting at the same time slot, the better channel utilization for the radio system is. The throughput of the system τ is calculated as $\sum_{tj} s_{tj}$ and is related to the *channel utilization* ρ by $\rho=\tau/NM$.

With the notation introduced in the previous paragraphs, the BSP is formulated as to find frame $S_{\rm opt}$:

1) with the shortest length, $M_{\min}$;
2) that satisfies constraints

$$\sum_{t=1}^{M} s_{tj} \geq 1 \qquad (j=1,2,\ldots,N) \tag{1}$$

$$\sum_{t=1}^{M}\sum_{i=1}^{N}\sum_{j=1}^{N} f_{ij}s_{ti}s_{tj} = 0; \tag{2}$$

3) makes τ maximum.

Constraint (1) forces the frame to allocate at least one time slot per radio station and constraint (2) stands for avoiding interference.

In order to solve this combinatorial optimization problem, it is a common practice to consider M as a parameter, which can be estimated by means of theoretical bounds from GT and to maximize τ subject to constraints (1) and (2). If a feasible solution is found for an M greater than the lower bound, M is decreased by one and a new maximum τ is searched for, until either the search is not succesful or the lowest possible M is reached. In this process, the algorithm to maximize τ and the estimation of a lower bound for M are the essential ingredients.

A. BSP Partitioning

The problem of maximizing τ, which we will refer to as *P1*, can be formulated as

$$\max \sum_{t=1}^{M}\sum_{j=1}^{N} s_{tj} \tag{3}$$

under

$$\sum_{t=1}^{M} s_{tj} \geq 1 \qquad (j=1,2,\ldots,N) \tag{4}$$

$$\sum_{t=1}^{M}\sum_{i=1}^{N}\sum_{j=1}^{N} f_{ij}s_{ti}s_{tj} = 0. \tag{5}$$

P1 is a hard problem for M close to its lower bound. Essentially, the difficulty arises due to constraint (4), which asks for *at least one* time slot for each radio station in the frame. To deal with this constraint, an intermediate optimization problem *P2* is defined to find a feasible frame with *one and only one* transmission per radio station. If a feasible solution for *P2* can be found, it will be improved by adding noninterferring transmissions to obtain an optimal solution to *P1*. The formulation for *P2* is as follows:

$$\text{find} \quad S=[s_{tj}] \tag{6}$$

such that

$$\sum_{t=1}^{M} s_{tj} = 1 \qquad (j=1,2,\ldots,N) \tag{7}$$

$$\sum_{t=1}^{M}\sum_{t=1}^{N}\sum_{j=1}^{N} f_{ij}s_{ti}s_{tj} = 0. \tag{8}$$

Note that objectives of *P1* and *P2* are quite different: for *P1*, the maximization of the number of simultaneous transmissions is proposed, but for *P2* the objective is to search a feasible frame, which is forced to have the same number of transmitting stations by constraint (7). These disparate objectives accomodate to different optimization strategies (GA and HNN, respectively), as detailed in Section III.

B. Some Comments About Lower Bounds for M

Given an interference pattern $G=(V,E)$ for a PRN, Wang and Ansari [3] establish a lower bound for M, that we will call here WA-LB

$$M \geq \triangle(G)+1 \tag{9}$$

where $\triangle(G)$ is the maximum degree[2] in G.

[2]The degree of a vertex is the number of incident arcs to this vertex.

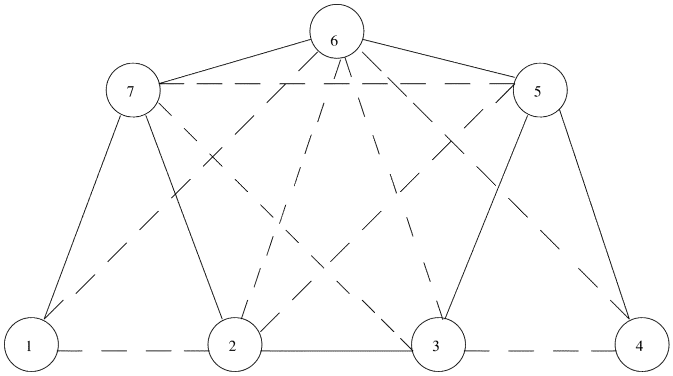

Fig. 1. Simple example of a graph for which WA-LB and GT-LB differ.

This bound is easy to calculate and gives a hint for initializing of the search process for the BSP. However, the bound is not tight, so that it might not halt the process at the right value of M, leading to unsuccessful and costly searches. A tighter bound coming from GT [9] can be easily applied, which will be useful in Section IV to show that our results for the selected problems give an optimum M. now, we introduce this bound, hereafter GT-LB, by means of a simple example.

Consider graph $G = (V, E)$ of Fig. 1, with seven nodes and edges in solid lines. This graph may be considered as the interference pattern of a PRN, where the nodes V represent stations and the arcs (solid lines) E represent collisions between one-hop apart stations. If we enlarge E with the edges in dashed lines in Fig. 1, we come up with a new graph, $G' = (V, E')$, with the same nodes or stations and arcs representing collisions between one hop and two hops apart stations. GT-LB is formulated over G' as[3]

$$M \geq \omega(G') \tag{10}$$

where $\omega(G')$ is the maximal cardinality of a clique[4] in G'. Applying (9) and (10), WA-LB for G is four while GT-LB for G is five, since $\{2, 3, 5, 6, 7\} \in G'$ is a clique.

III. Two-Stage Algorithm

A. HNN for P2

The algorithm proposed for *P2* is a discrete-time binary HNN [10]. The structure of this HNN can be described as a graph where the set of nodes are the neurons and the set of edges defines the connections between them. To solve *P2*, this graph is mapped to the frame structure S defined in Section II. Neuron s_{ti} represents a possible assignment of a time slot t to station i: if this station is allowed to transmit at time slot t, $s_{ti} = 1$, otherwise, $s_{ti} = 0$. The connection between neurons s_{ti} and $s_{t'j}$ is represented by setting $w_{ti,t'j} = 1$; if there is no connection, $w_{ti,t'j} = 0$. In order to avoid interference, if stations i and j are one or two hops apart, $w_{ti,tj}$ is one for every t. To constrain one and only one transmission per frame, $w_{ti,t'i}$ is one for every $t \neq t'$. The rest of $w_{ti,t'j}$, incluiding terms of the form $w_{ti,ti}$, are set to zero. Note that the symmetry assumed for the interference pattern in Section II makes the connections of the HNN also symmetric: $w_{ti,t'j} = w_{t'j,ti}$.

After a random initialization of the neuron values, this HNN operates in serial mode, which means that only one neuron is updated at a time, while the rest are left unchanged. Denoting by $s_{ti}(h)$ the state of neuron s_{ti} at time h, the update rule is described by

$$s_{ti}(h+1) = \operatorname{isgn}\left[\sum_{t'=1}^{M}\sum_{j=1}^{N} w_{ti,t'j}s_{t'j}(h)\right] \tag{11}$$

for every t and i; the isgn operator is defined by

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

A cycle is defined as the set of $M \times N$ sucessive neuron updates in a given order. In a cycle, every neuron is updated once following a random order, which is fixed during the execution of the algorithm. After every cycle, the HNN is checked for convergence, which is reached if none of the neurons have changed their state in the cycle. The final state of the HNN dynamics, where the algorithm has converged, is interpreted as the frame solution for *P2*.

The solution provided for the neural algorithm described above is guaranteed to satisfy interference constraint (2), but it may not provide a time slot for every station; therefore, it is necessary to run the neural network several times until we find a feasible solution, or we decide to halt the procedure.

B. Combination of the GA and HNN for P1

A solution for *P2*, hereafter S_{P2}, provides one transmission for every station which will be preserved by the hybrid algorithm; a simple GA is used to obtain optimal throughputs and the HNN will preserve feasibility in every step of the GA. In this way, we force the two characteristics of a solution for *P1*: to have maximum thoughput and to be feasible.

The HNN for *P1* is essentially the neural algorithm described in the previous subsection with two modifications: the neurons which represent S_{P2} are fixed to one, hence, every station transmits at least once per frame; and $w_{ti,t'i}$ are switched to zero for every $t \neq t'$, so that the constraint of one and only one transmission per frame is eliminated.

A simple GA takes a population of solutions for *P1*, represented as binary strings and evolves it through sucessive generations by means of the application of three operations on the population: selection, crossover, and mutation [11].

Selection is the process by which individuals in the population are randomly sampled with probabilities proportional to their fitness values, which, in this case, are defined as their throughputs. The larger the frame throughput, the higher the probability of been selected is. An elitist strategy, consisting of copying the highest throughput frame always, is applied in order to preserve the best solution encontered, thus, far in the evolution.

[3]This is a well-known bound for the chromatic number in GT, see [9] for details.

[4]A clique in a graph $G' = (V, E')$ is a subset H of V such that any two vertices in H are adjacent.

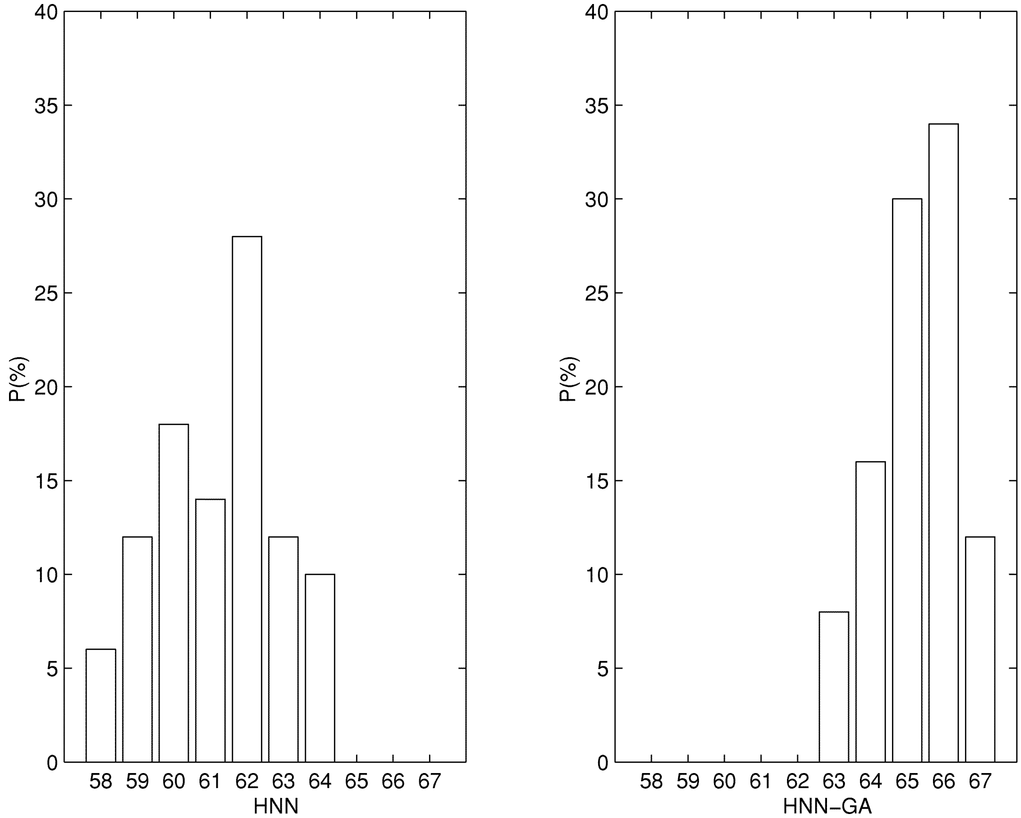

Fig. 2. Histograms of the number of time slots obtained with and without the GA in *P1* (Problem #3). P stand for the percentage of feasible solutions, whereas HNN-GA and HNN stands for the number of slots assigned with and without GA, respectively.

The selected set, of the same size of the initial population, is subjected to the crossover operation. First, the frames are coupled at random. Secondly, 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 one to zero, or *vice versa*, with a very small probability P_m.

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

The overall algorithm for the BSP may proceed as follows. First, an initial value for the frame length can be guessed from the lower bounds discussed in the previous section. For a given frame length, the HNN for *P2* is used to search a feasible frame. If such a frame is found, a shorter frame length can be tried until the lower bound for the frame length is reached or the computational resources are exhausted. Once such a feasible frame with minimal length is found, the algorithm for *P1* gives an optimal throughput frame.

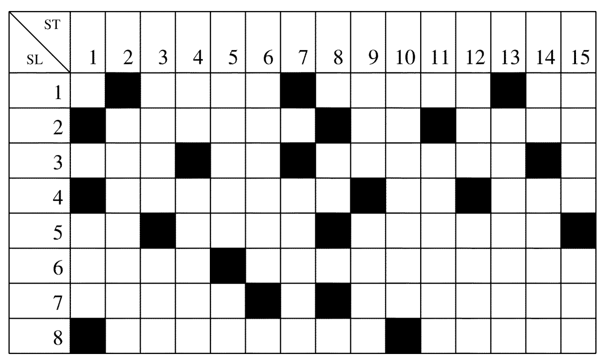

Fig. 3. Broadcast schedule obtained by the HNN-GA algorithm in Problem #1. ST and SL stand for the number of stations and the number of slots, respectively.

TABLE I
BROADCAST PROBLEMS #1, #2 AND #3 CHARACTERISTICS. N, $\min D$, $\max D$, avg$D$, WA-LB, GT-LB STAND FOR NUMBER OF STATIONS, MINIMUM DEGREE, MAXIMUM DEGREE AND AVERAGE DEGREE, WANG'S LOWER BOUND AND OUR LOWER BOUND FOR M, RESPECTIVELY

Problem	N	minD	maxD	AvgD	$WA-LB$	$GT-LB$
#1	15	2	7	3.86	8	8
#2	30	2	8	4.63	9	10
#3	40	1	7	3.3	8	8

IV. EXPERIMENTAL RESULTS

In order to test the performance of our algorithm, three scheduling problems have been chosen for which results by alternative algorithmic approaches are available [3]. Among these approaches, the mean field annealing (MFA) [3] provides the best results; we will use it as a comparison reference for the new algorithm. These BSPs are referred as Problems #1, #2, and #3, and their main characteristics along with lower bounds WA-LB

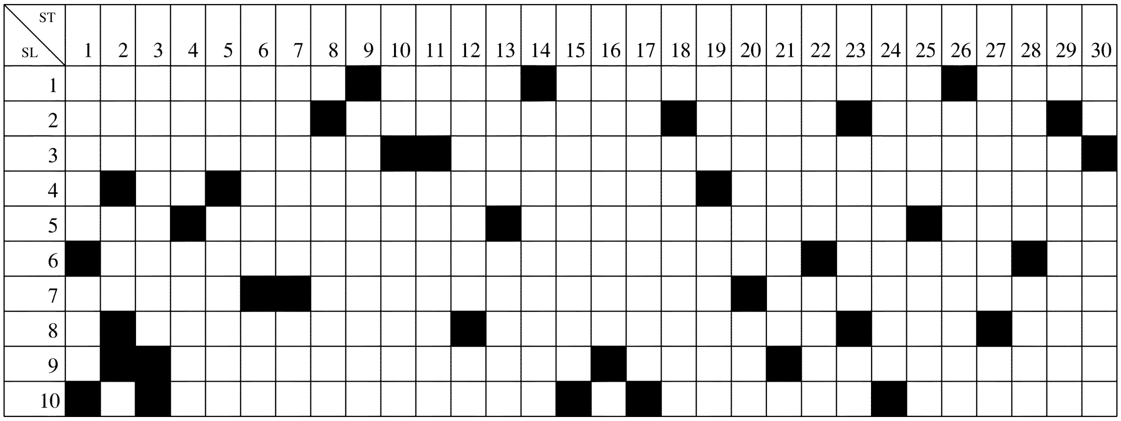

Fig. 4. Broadcast schedule obtained by the HNN-GA algorithm in Problem #2. ST and SL stand for the number of stations and the number of slots, respectively.

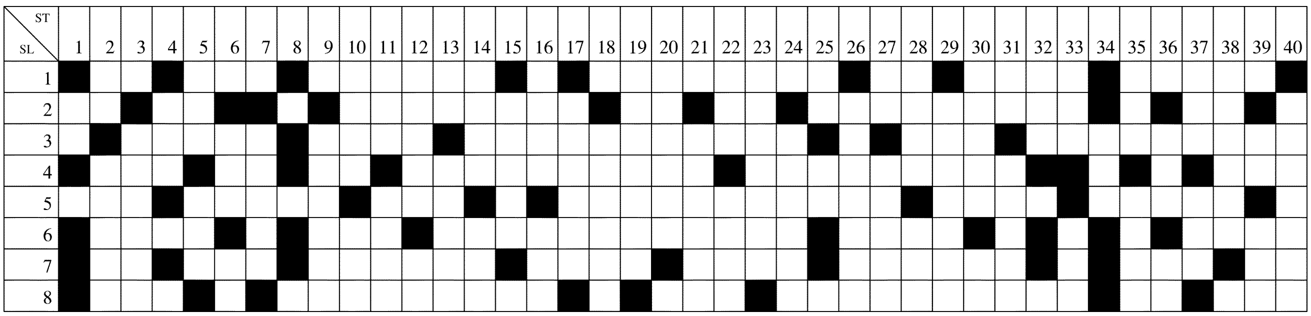

Fig. 5. Broadcast schedule obtained by the HNN-GA algorithm in Problem #3. ST and SL stand for the number of stations and the number of slots, respectively.

TABLE II
THROUGHPUTS BY THE MFA AND THE HNN-GA ALGORITHMS

Net Size	M	τ from MFA	τ from HNN-GA
15	8	18	20
30	10	-	35
30	11	-	40
30	12	39	47
40	8	-	67
40	9	71	77

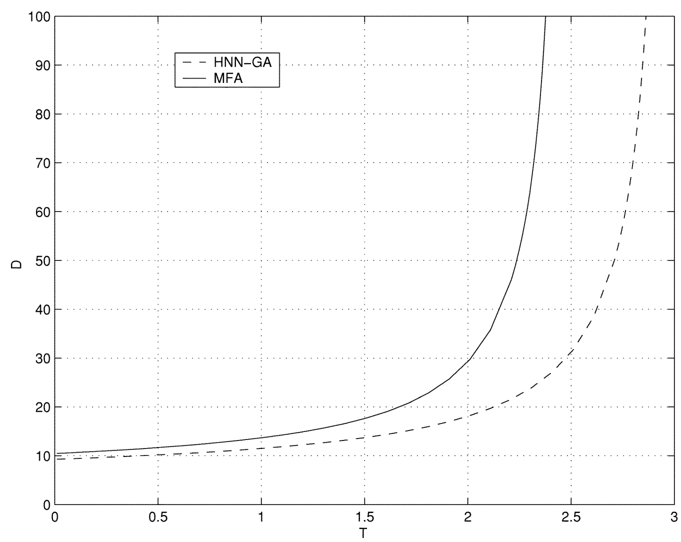

Fig. 6. Comparison of average time delays obtained by the MFA algorithm and the HNN-GA algorithm (Problem #2). D stands for the average time delay whereas T stands for the total arrival rate (packets/slot).

and GT-LB (see Section II-B), are summarized in Table I. The GT-LBs are derived from cliques {5, 6, 9, 10, 11, 13, 14, 15}, {4, 10, 12, 14, 17, 18, 19, 20, 21, 22}, and {11, 12, 18, 19, 20, 26, 27, 28} for Problems #1, #2, and #3, respectively. Note that only for Problem #2 WA-LB and GT-LB are different. Since BSP is NP-hard and the size of the problems grows from Problem #1 to Problem #3, their difficulty is also expected to grow accordingly. However, the net structure[5] for Problem #2 makes this problem the most difficult of them. This fact also clarifies some of the results reported below.

The HNN structures for the benchmark problems are obtained from their interference pattern graphs in [3], applying the mapping as explained in Section II. For the simple GA, probability of crossover P_c and probability of mutation P_m have been set to typical values: 0.6 and 0.01, respectively [11]. The GAs population sizes are 10, 50, and 25 for Problems #1, #2, and #3, respectively. Although these population sizes are quite small when compared with typical ones, the GA exhibit good performance essentially due to its synergy with the HNN algorithm [7], [8]. The performance difference in adding a GA to the HNN for ***P1*** can be observed in the histograms of Fig. 2, corresponding to Problem #3.

The optimal frames obtained by our algorithm, HNN-GA hereafter, are given in Figs. 3, 4, and 5 for Problems #1, #2, and #3, respectively. Note that their frame lengths are optimum,

[5]This structure is summarized in Table I by means of the minimum, the maximum, and the average degree in a graph, which is the most important parameter.

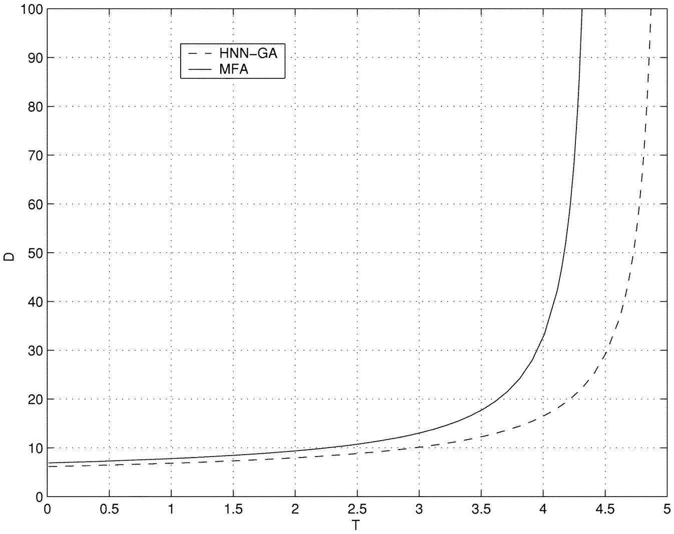

Fig. 7. Comparison of average time delays obtained by the MFA algorithm and the HNN-GA algorithm (Problem #3). D stands for the average time delay whereas T stands for the total arrival rate (packets/slot).

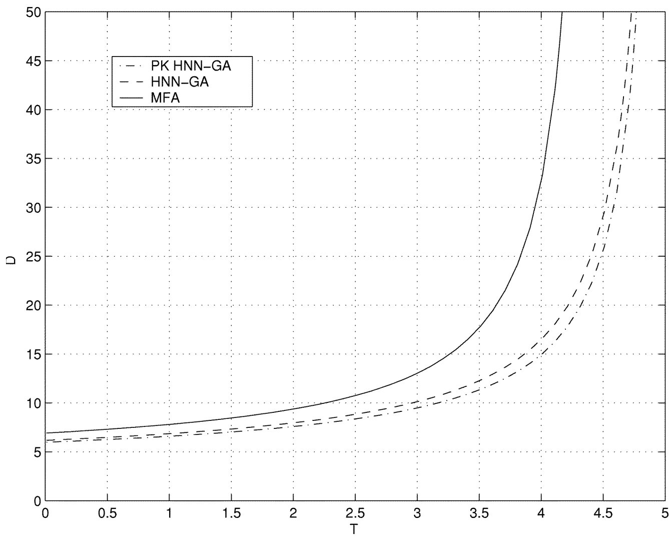

Fig. 8. Comparison of average time delays obtained by the MFA algorithm, the HNN-GA algorithm and the PK HNN-GA algorithm (Problem #3). D stands for the average time delay whereas T stands for the total arrival rate (packets/slot).

as can be verified from GT-LBs in Table I. The MFA obtains an optimum frame length only for the easiest problem and has special difficulties with Problem #2, as can be concluded from data in Table II, which shows the optimal throughputs obtained by MFA and HNN-GA for the three BSPs and different frame lengths. HNN-GA outperforms MFA in all cases and the difference between their results grows as the problem difficulty does.

The influence of these results in the PRN packet delay is quite significant, as it can be observed in Figs. 6 and 7, which show the comparison between average time delays obtained from the MFA and the HNN-GA for Problems #2 and #3, respectively. The average time delay obtained by the HNN-GA for Problem #1 equals the obtained by the MFA and we do not show them. The delay has been calculated with the Pollaczec–Khinchin formula [12], which assumes independent Poisson transmissions by the radio stations.

If the ultimate objective is a minimum network delay, a straightforward modification can be done without essentially changing the computational load. This modification consists of setting the delay (as given by the Pollaczec–Khinchin formula) as the GA fitness function instead of the throughput, we will call this approach PK HNN-GA. The differences in delay for both fitness functions are illustrated in Fig. 8.

V. Conclusion

In this paper, we propose a new algorithm for the BSP. The algorithm solves this optimization problem in two stages: the first to find a feasible solution and the second to obtain maximal transmission packing. A HNN is used for the first stage and a combination of the neural net and a GA for the second. The algorithm is tested for some examples, in which it obtains optimum frame lengths and better transmission packings than MFA.

References

[1] "Special issue on packet radio networks," *Proc. IEEE*, vol. 75, pp. 6–167, Jan. 1987.
[2] A. Ephremides and T. V. Truong, "Scheduling broadcast in multihop radio networks," *IEEE Trans. Commun.*, vol. 38, pp. 456–460, Apr. 1990.
[3] 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.
[4] N. Funabiki and Y. Takefuji, "A parallel algorithm for broadcast scheduling problems in packet radio networks," *IEEE Trans. Commun.*, vol. 41, pp. 828–831, June 1993.
[5] W. K. Hale, "Frequency assignment: Theory and applications," *Proc. IEEE*, vol. 68, pp. 1497–1514, Dec. 1980.
[6] 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.
[7] C. Bousoño-Calzón and A. R. Figueiras-Vidal, "Emerging techniques for dynamic frequency assignment: Merging genetic algorithms and neural networks," in *Proc. NATO Symp. Information Systems Technology*, Aalborg, Denmark, Oct. 1998, pp. 12.1–12.5.
[8] Y. Watanabe, N. Mizuguchi, and Y. Fujii, "Solving optimization problems by using a Hopfield neural network and genetic algorithm combination," *Syst. Comput. Japan*, vol. 29, no. 10, pp. 68–73, Jan. 1998.
[9] D. Jungnickel, *Graphs, Networks and Algorithms*. Berlin, Germany: Springer-Verlag, 1999.
[10] K. Y. Siu, V. Roychowdhury, and T. Kailath, *Discrete Neural Computation: A Theorical Foundation*. Englewood Cliffs, NJ: Prentice-Hall, 1997.
[11] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*. Reading, MA: Addison-Wesley, 1989.
[12] D. Bertsekas and R. Gallager, *Data Networks*. Englewood Cliffs, NJ: Prentice-Hall, 1987.
[13] J. Hertz, A. Krogh, and R. G. Palmer, *Introduction to the Theory of Neural Computation*. Redwood City, CA: Addison-Wesley, 1991.
[14] C. M. Barnhart, J. E. Wieselthier, and A. Ephremides, "Neural network techniques for scheduling and routing problems in multihop radio networks," in *Proc. MILCOM'91 IEEE*, New York, 1991, pp. 407–413.
[15] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. New York: W. H. Freeman, 1979.
[16] E. Arikan, "Some complexity results about packet radio networks," *IEEE Trans. Inform. Theory*, vol. 30, pp. 681–685, July 1984.
[17] B. Hajek and G. Sasaki, "Link scheduling in polynomial time," *IEEE Trans. Inform. Theory*, vol. 38, pp. 910–917, Sept. 1988.
[18] J. H. Ju and V. O. K. Li, "TDMA scheduling design of multihop packet radio networks based on latin squares," *IEEE J. Select. Areas Commun.*, vol. 17, pp. 1345–1352, Aug. 1999.
[19] E. Aarts and J. K. Lengstra, *Local Search in Combinatorial Optimization*. New York: Wiley, 1987.

[20] C. M. Barnhart, J. E. Wieselthier, and A. Ephremides, "A neural network approach to solve the link activation problem in multihop radio networks," *IEEE Trans. Commun.*, vol. 43, pp. 1277–1283, Feb./Mar./Apr. 1995.

[21] Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs*. Berlin, Germany: Springer-Verlag, 1992.

[22] 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.

[23] F. Q. Bac and V. L. Perov, "New evolutionary algorithm for NP-complete combinatorial optimization problems," *Biol. Cybern.*, vol. 69, no. 3, pp. 229–224, July 1993.

Sancho Salcedo-Sanz (S'00) was born in Madrid, Spain, in 1974. He received the B.S. degree in physics from the Universidad Complutense de Madrid, Spain, in 1998. He is currently working toward the Ph.D. degree in telecommunications in the Department of Signal Theory and Communications, Universidad Carlos III de Madrid, Spain.

He has published/copublished several papers in the field of genetic algorithms and hybrid algorithms. His current interests deal with genetic optimization, hybrid algorithms, and neural networks.

Carlos Bousoño-Calzón (M'95) received 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, Leganés-Madrid, Spain. His research interests are focused on optimization in communications, genetic algorithms, and neural networks. He has coauthored more than 20 international journals and conference papers in these areas.

Anibal R. Figueiras-Vidal (S'74–M'76–SM'84) received the Tecomm Engineer degree from Universidad Politécnica de Madrid, Spain, in 1973, and the Ph.D. degree from Universidad Politécnica de Barcelona, in 1976.

He is a Professor in the Department Signal Theory and Communications, Universidad Carlos III de Madrid, and Head of the Department of Signal Theory and Communications. His research interests are digital signal processing, digital communications, neural networks, and learning theory. In these subjects, he has coauthored more than 200 international journals and conference papers.

Dr. Figueiras is a Member of the Spain Academy of Engineering. He is an Associate Editor for the IEEE TRANSACTIONS ON NEURAL NETWORKS.