Population-Based Incremental Learning to Solve the FAP Problem

Jose M. Chaves-González, Miguel A. Vega-Rodríguez, David Domínguez-González, Juan A. Gómez-Pulido, Juan M. Sánchez-Pérez
University of Extremadura. Dept. Technologies of Computers and Communications
Escuela Politécnica, Campus Universitario s/n, 10071, Cáceres, Spain
{jm, mavega}@unex.es, david-dg@hotmail.com, {jangomez, sanperez}@unex.es

Abstract

Frequency assignment problem (FAP) is a very important issue in the field of telecommunications (especially in GSM –Global System for Mobile– Networks). In this work, we present the Population-Based Incremental Learning (PBIL) algorithm to solve a particular branch of the FAP problem (MS-FAP). MS-FAP (Minimum Span Frequency Assignment Problem) tries to minimize the range of frequencies which is necessary in a certain area to cover the communications which take place there. In this paper it is presented the problem and it is explained the methodology which solve it. We have performed tests with a complete set of experiments using seven well-known variations of PBIL and 7 types of MS-FAP problems. At the end, the results are presented and we compare them to conclude which variation of PBIL provides the best solution to the MS-FAP problem.

1. Introduction

Many of the problems which are found in the Telecommunications area can be formulated as optimization tasks. This is the case of the frequency assignment problem (FAP), which is an important task for current GSM (Global System for Mobile) operators. GSM is certainly the most successful mobile communication system. Indeed, by mid 2006 GSM services are in use by more than 1.8 billion subscribers [1] across 210 countries, representing approximately 77% of the world's cellular market.

As we will explain in the following section, FAP is a very complex problem (in fact, it is an NP problem), so its resolution using evolutionary algorithms is very appropriate. Many different methods have been proposed in the literature [2] and, among them, metaheuristic algorithms [3] have proved to be particularly effective. In this work we propose the use of PBIL (Population-Based Incremental Learning) algorithm to solve the FAP problem. In the FAP problem the main goal lies in assigning one or more frequencies to a set of antennas. The number of available frequencies is limited, and normally much inferior to the total number of frequencies which are required. Therefore, the problem solved in this work performs the most appropriate assignment of these available frequencies, knowing that they need to be repeated (there are more demanded frequencies than available frequencies). Among the different variations in which is divided the FAP problem, we have focused on the problem which is given by the Philadelphia instances. The Philadelphia instances set [4] is one of the most representative set of problems in the literature about FAP. In this work we have focused on the minimum span FAP (MS-FAP variant, used in Philadelphia problems). With this approach, we will evaluate the different configurations and versions of PBIL algorithm to solve this problem, as well as the influence of each configuration parameter in the goodness of the results.

The rest of the paper is structured as follows: In section 2 we present and describe the FAP problem. Section 3 provides a brief explanation of the PBIL algorithm. After that, in section 4, the Philadelphia test problems are described. The experiments and the results obtained are explained in section 5. Finally, conclusions are described in the last section.

2. The frequency assignment problem

The FAP problem can be described in an simple way as the assignment of a limited number of available frequencies to a set of base stations (BS, antennas or cells). The problem is that on the one hand, the number of demanded frequencies for each base station is far beyond the available number of frequencies, and on

978-0-7695-3369-8/08 $25.00 © 2008 IEEE
DOI 10.1109/ADVCOMP.2008.10

Authorized licensed use limited to: University of Johannesburg. Downloaded on March 02,2010 at 05:14:41 EST from IEEE Xplore. Restrictions apply.

the other hand, this assignment depends on certain constraints which may cause interferences. For example, two base stations which are close enough and use near frequencies will interfere between them, and this situation is undesirable, because depending on the interference produced, the communication will be bad (with poor quality) or even impossible, which is completely unacceptable.

There are several types of FAP problem depending on the frequency assignment which is done: the minimum order frequency assignment problem (called MO-FAP), the minimum span frequency assignment problem (called MS-FAP), the minimum blocking frequency assignment problem (MB-FAP) and the minimum interference frequency assignment problem. The MO-FAP lies in the assignment of the frequencies in such a way that no interference occurs and in the minimization of the number of different used frequencies. The objective in the MS-FAP problem is to assign frequencies in such a way that no interference occurs, and the difference between the maximum and the minimum used frequency, the span, is minimized. But sometimes interferences are unavoidable. In this case, the problem must be solved to make them minimum. Interference-free assignment could be reached by two possible approaches: trying to find an assignment that serves as many antennas as possible, or allowing an assignment with some interferences but minimizing them. The first approach is known as minimum blocking frequency assignment problem (MB-FAP), and the other is the minimum interference frequency assignment problem (MI-FAP).

2.1. Problem formulation

The formulation of the problem is the following: Given a number of antennas and a set of frequencies, each antenna must perform a concrete number of communications at the same time. We have to assign to each antenna the number of frequencies necessary to carry out all the communications that the antenna has to do. However, we have to avoid the interferences, because they can reduce the quality of service (QoS) down to unsatisfactory levels. Therefore, we have to consider three constraints [5] which represent the electromagnetic compatibility constraints (EMC) and which need to be satisfied in the FAP:

- Co-site constraints: frequencies assigned to the same base station have to respect a minimum distance.
- Co-channel constraints: equal frequencies can be assigned only to base stations which are a minimum distance from each other.
- Adjacent-channel constraints: adjacent frequencies cannot be assigned to adjacent base stations simultaneously. They need to respect a specified distance.

If these restrictions are not satisfied then interferences will take place, making not possible the use of certain frequencies. The EMC to represent a network of n cells is described by an $n_x n$ symmetric matrix named interference matrix C. Each non-diagonal element C_{ij} in C represents the minimum separation distance in the frequency domain between the frequencies assigned to the cell i and the cell j. The C_{ij}=0 indicates that cells i and j can use the same frequency. Each diagonal C_{ii} in C represents the minimum separation distance between any two frequencies assigned to cell i. This last is the co-site constraint. For the FAP problem it is also necessary to know the demands for each BS. This is represented by an n-element vector, which is called: demand vector D. Each element d_i in D represents the number of frequencies to be assigned to the BS i. The f_{ik} represents the k^{th} frequency assigned to BS i and the EMC is represented in equation (1).

$$\left| f_{ik} - f_{jl} \right| \geq C_{ij}, \, where \, i, j = 1..n; \\ k = 1..d_i; l = 1..d_j; i \neq j \; or \; k \neq l \qquad (1)$$

The FAP tries to find an optimum assignment free of conflicts. But it is necessary another parameter in addition to the interference matrix and the demand vector in the variant (MS-FAP) we work with for this paper. This parameter is called: lower bound *(lb)* [5] and it represents the minimum value of the maximum f_{ik} for all i and k, so that no interference is caused. The *lb* parameter indicates that the frequencies can take values between 1 and *lb* without violation of any constraint. Therefore, from the parameter *lb* it can be calculated the span: *lb - 1*.

3. Population-based incremental learning

PBIL [6, 7] is a method that combines the genetic algorithms with the competitive learning (typical in artificial neural networks) for function optimization. Due to the limitation of pages of this paper, we encourage the reader who wants a complete explanation about PBIL to consult references [6, 7].

4. The Philadelphia test problems

The Philadelphia instances [4] have been quite a lot studied and they count with a great amount of references in the literature. The Philadelphia instances are characterized by 21 hexagons denoting the cells

Authorized licensed use limited to: University of Johannesburg. Downloaded on March 02,2010 at 05:14:41 EST from IEEE Xplore. Restrictions apply.

(antennas) of a cellular phone network around Philadelphia. Each antenna requires a high number of frequencies due to the number of communications that it has to serve, but as we explained in section 2.1, there are some restrictions that the planning solution has to satisfy if we do not want any interference (e.g. adjacent antennas cannot have assigned the same frequencies). Philadelphia problem set is a MS-FAP, so the objective in these problems is, in a no-interference solution, to minimize the span (difference between the maximum and the minimum used frequency) which is used to cover the space where the communications are necessary. Moreover, the number of frequencies that each antenna needs to cover its area is not constant, because each cell needs a specific number of frequencies to give support to the communications that are produced in its zone. Other important parameter is the minimum separation distance, in terms of number of frequencies, between the cells to avoid the interferences. Due to the space requirements we are not going to explain more the theoretical concepts of the problem. To obtain a complete description about the Philadelphia problem set, please, consult the reference [4]. However, it is important to point that the way in which the Philadelphia problem set is solved is divided into two steps: Firstly, we have to find a solution with no interferences in any antenna and then, we have to find different solutions that make minimum the distance between the highest and the lowest frequencies which are used in the planning (MS-FAP problem). We try to generate the best solution possible using a mathematical algorithm to generate solutions free of interferences. Then, we use an evolutionary algorithm, PBIL, to generate more solutions which improve the results obtained with the first solution (using fewer frequencies for the planning).

4.1. First solution generation

The initial solution to solve the problem is obtained using an algorithm divided into 3 stages. We illustrate the generation of the first solution with an easy example in fig. 1. In the first stage only one frequency is assigned to each antenna and then it is checked (using the interference matrix) so that no interferences exist among the different antennas (figure 1.a). This assures that one frequency is assigned for each antenna with no interferences, but in the Philadelphia instances it is not possible to find not even a single antenna with only one frequency, so in the second stage, we duplicate the structure and we obtain two frequencies assigned for each antenna. However, this duplication will probably create interferences, that have to be erased making adjustments in the frequency assignation (with the insertion of empty columns where necessary -see fig. 1.b-) until there are not interferences in the frequency planning. This replication of the structure is repeated until the maximum number of frequencies that needs one antenna is covered.

In the first stage only one frequency is assigned to each antenna and then it is checked (using the interference matrix) so that no interferences exist among the different antennas (figure 1.a). This assures that one frequency is assigned for each antenna with no interferences, but in the Philadelphia instances, there are not antennas with one only frequency, so in the second stage, we duplicate the structure and we obtain two frequencies assigned for each antenna. However, this duplication will probably create interferences, that have to be erased making adjustments in the frequency assignation (with the insertion of empty columns where necessary -see fig. 1.b-) until there are not interferences in the frequency planning. This replication of the structure is repeated until the maximum number of frequencies that needs one antenna is covered. Finally, the frequencies which are not necessary in each antenna are erased (all antennas have the same number of frequencies at the beginning of the stage 3, and this is not usual) and the planning is improved packing the frequencies assignation where it is possible (deleting empty columns, as it can be seen in fig. 1.c). At the end, we have created a chromosome which represents a valid planning, and it is used in order to generate the initial probability vector [6] for the PBIL algorithm.

4.2. PBIL improvement solution

Once we have an initial probability vector created using the algorithm described in the last section, we follow the PBIL algorithm, generating the *S* samples (individuals) in the population. Each individual is evaluated, calculating the difference between the highest and the lowest values of the frequencies (span) which are used in that chromosome. The objective of PBIL consists in improving these solutions to obtain individuals that get a minimum span. For doing this, PBIL learns from the matrix which represents the chromosome how to use the same frequencies (or very close frequencies) in antennas which do not cause interferences among them. At the end of the algorithm, PBIL obtains a frequency planning which obtains the minimum span frequency assignment, which is the objective in the MS-FAP problems.

Authorized licensed use limited to: University of Johannesburg. Downloaded on March 02,2010 at 05:14:41 EST from IEEE Xplore. Restrictions apply.

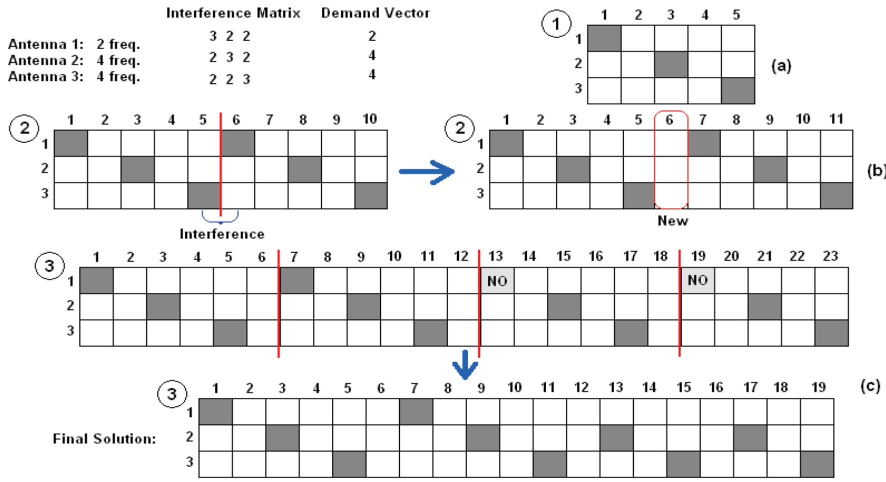

Figure 1. Simple example for the first solution generation to solve the Philadelphia problem

5. Experiments and results

In this section we explain the experiments we have done to verify the effectiveness of the usage of the PBIL algorithm to solve the FAP problem. We have used the Philadelphia problem instances [4, 5] (well-known instances for the FAP problem). They include a number of hexagonal cells (habitually 21, [4]) that represents the cellular phone network in Philadelphia city. In these instances (table 1), the number of antennas (cells) varies from 4 to 21, and the number of frequencies finally used varies between 11 and 533. Figure 2 shows the different interference matrices (C_i) and the demand vectors (D_i) used.

D_1	D_2	D_3
1	8	5
1	25	5
1	8	5
3	8	8
	8	12
	15	25
	18	30
	52	25
	77	30
	28	40
	13	40
	15	45
	31	20
	15	30
	36	25
	57	15
	28	15
	8	30
	10	20
	13	20
	8	25

C_1

5	4	0	0
4	5	0	1
0	0	5	2
0	1	2	5

C_2

5	1	1	0	0	1	1	1	1	0	0	0	0	1	1	1	0	0	0	0	0
1	5	1	1	0	0	1	1	1	1	0	0	0	0	1	1	1	0	0	0	0
1	1	5	1	1	0	0	1	1	1	1	0	0	0	0	1	1	1	0	0	0
0	1	1	5	1	0	0	0	1	1	1	1	0	0	0	0	1	1	0	0	0
0	0	1	1	5	0	0	0	0	1	1	1	0	0	0	0	0	1	0	0	0
1	0	0	0	0	5	1	1	0	0	0	0	1	1	1	0	0	0	0	0	0
1	1	0	0	0	1	5	1	1	0	0	0	1	1	1	1	0	0	1	0	0
1	1	1	0	0	1	1	5	1	1	0	0	0	1	1	1	1	0	1	1	0
1	1	1	1	0	0	1	1	5	1	1	0	0	0	1	1	1	1	1	1	1
0	1	1	1	1	0	0	1	1	5	1	1	0	0	0	1	1	1	0	1	1
0	0	1	1	1	0	0	0	1	1	5	1	0	0	0	0	1	1	0	0	1
0	0	0	1	1	0	0	0	0	1	1	5	0	0	0	0	0	1	0	0	0
0	0	0	0	0	1	1	0	0	0	0	0	5	1	1	0	0	0	0	0	0
1	0	0	0	0	1	1	1	0	0	0	0	1	5	1	1	0	0	1	0	0
1	1	0	0	0	1	1	1	1	0	0	0	1	1	5	1	1	0	1	1	0
1	1	1	0	0	0	1	1	1	1	0	0	0	1	1	5	1	1	1	1	1
0	1	1	1	0	0	0	1	1	1	1	0	0	0	1	1	5	1	1	1	1
0	0	1	1	1	0	0	0	1	1	1	1	0	0	0	1	1	5	0	1	1
0	0	0	0	0	0	1	1	1	0	0	0	0	1	1	1	1	0	5	1	1
0	0	0	0	0	0	0	1	1	1	0	0	0	0	1	1	1	1	1	5	1
0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	1	1	1	1	1	5

C_3

7	1	1	0	0	1	1	1	1	0	0	0	0	1	1	1	0	0	0	0	0
1	7	1	1	0	0	1	1	1	1	0	0	0	0	1	1	1	0	0	0	0
1	1	7	1	1	0	0	1	1	1	1	0	0	0	0	1	1	1	0	0	0
0	1	1	7	1	0	0	0	1	1	1	1	0	0	0	0	1	1	0	0	0
0	0	1	1	7	0	0	0	0	1	1	1	0	0	0	0	0	1	0	0	0
1	0	0	0	0	7	1	1	0	0	0	0	1	1	1	0	0	0	0	0	0
1	1	0	0	0	1	7	1	1	0	0	0	1	1	1	1	0	0	1	0	0
1	1	1	0	0	1	1	7	1	1	0	0	0	1	1	1	1	0	1	1	0
1	1	1	1	0	0	1	1	7	1	1	0	0	0	1	1	1	1	1	1	1
0	1	1	1	1	0	0	1	1	7	1	1	0	0	0	1	1	1	0	1	1
0	0	1	1	1	0	0	0	1	1	7	1	0	0	0	0	1	1	0	0	1
0	0	0	1	1	0	0	0	0	1	1	7	0	0	0	0	0	1	0	0	0
0	0	0	0	0	1	1	0	0	0	0	0	7	1	1	0	0	0	0	0	0
1	0	0	0	0	1	1	1	0	0	0	0	1	7	1	1	0	0	1	0	0
1	1	0	0	0	1	1	1	1	0	0	0	1	1	7	1	1	0	1	1	0
1	1	1	0	0	0	1	1	1	1	0	0	0	1	1	7	1	1	1	1	1
0	1	1	1	0	0	0	1	1	1	1	0	0	0	1	1	7	1	1	1	1
0	0	1	1	1	0	0	0	1	1	1	1	0	0	0	1	1	7	0	1	1
0	0	0	0	0	0	1	1	1	0	0	0	0	1	1	1	1	0	7	1	1
0	0	0	0	0	0	0	1	1	1	0	0	0	0	1	1	1	1	1	7	1
0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	1	1	1	1	1	7

C_4

5	2	1	1	0	1	2	2	1	1	0	0	1	1	1	1	1	0	1	1	0
2	5	2	1	1	1	1	2	2	1	1	0	0	1	1	1	1	1	1	1	1
1	2	5	2	1	0	1	1	2	2	1	1	0	0	1	1	1	1	1	1	1
1	1	2	5	2	0	0	1	1	2	2	1	0	0	0	1	1	1	0	1	1
0	1	1	2	5	0	0	0	1	1	2	2	0	0	0	0	1	1	0	0	1
1	1	0	0	0	5	2	1	1	0	0	0	2	2	1	1	0	0	1	0	0
2	1	1	0	0	2	5	2	1	1	0	0	1	2	2	1	1	0	1	1	0
2	2	1	1	0	1	2	5	2	1	1	0	1	1	2	2	1	1	1	1	1
1	2	2	1	1	1	1	2	5	2	1	1	0	1	1	2	2	1	1	1	1
1	1	2	2	1	0	1	1	2	5	2	1	0	0	1	1	2	2	1	1	1
0	1	1	2	2	0	0	1	1	2	5	2	0	0	0	1	1	2	0	1	1
0	0	1	1	2	0	0	0	1	1	2	5	0	0	0	0	1	1	0	0	1
1	0	0	0	0	2	1	1	0	0	0	0	5	2	1	1	0	0	1	0	0
1	1	0	0	0	2	2	1	1	0	0	0	2	5	2	1	1	0	1	1	0
1	1	1	0	0	1	2	2	1	1	0	0	1	2	5	2	1	1	2	1	1
1	1	1	1	0	1	1	2	2	1	1	0	1	1	2	5	2	1	2	2	1
1	1	1	1	1	0	1	1	2	2	1	1	0	1	1	2	5	2	1	2	2
0	1	1	1	1	0	0	1	1	2	2	1	0	0	1	1	2	5	1	1	2
1	1	1	0	0	1	1	1	1	1	0	0	1	1	2	2	1	1	5	2	1
1	1	1	1	0	0	1	1	1	1	1	0	0	1	1	2	2	1	2	5	2
0	1	1	1	1	0	0	1	1	1	1	1	0	0	1	1	2	2	1	2	5

Figure 2. Interference matrices (C_i) and demand vectors (D_i) for the tested instances

As we have said in the previous sections, we have focused our experiments on the variant MS-FAP, which objective is to find the minimum difference between the greatest and the smallest frequencies used by all the antennas. That is, the smallest span to obtain a null cost (there are not any interferences). The values presented in the column *lb* from table 1 correspond to the best lower bound value (minimum span +1) reached so far by other researchers in this area using very different metaheuristics, as we can see in [4]. These values have been used to validate the results obtained with PBIL.

Table 1. Philadelphia instances used in the experiments

Instance	BS number	Lower Bound *(lb)*	Interference matrix (*C*)	Demand vector (*D*)
Prob. 1	4	11	C_1	D_1
Prob. 2	21	381	C_2	D_2
Prob. 3	21	533	C_3	D_2
Prob. 4	21	533	C_4	D_2
Prob. 5	21	221	C_2	D_3
Prob. 6	21	309	C_3	D_3
Prob. 7	21	309	C_4	D_3

Moreover, it is important to point out that we have performed all the experiments with the standard configuration of PBIL [6] (100 individuals, 0.10 of learning rate, 0.02 of mutation probability, 0.05 of mutation amount, 0.075 of negative learning rate and M=2), but using 7 different variations of the algorithm, which are the following: *PBIL-Standard*; *PBIL-NegativeLR* (where the probability vector is moved towards the best individual -using the learning rate- and also away from the worst individual -using the negative learning rate- in each generation; *PBIL-Different* (the probability vector is only moved, in each

Authorized licensed use limited to: University of Johannesburg. Downloaded on March 02,2010 at 05:14:41 EST from IEEE Xplore. Restrictions apply.

generation, towards the bits in the best individual which are different than those in the worst individual); *PBIL-M-Equitable* (the probability vector is moved equally in the direction of each of the M selected individuals -M best samples- in each generation); *PBIL-M-Relative* (the probability vector is moved according to the relative evaluations -fitness functions- of the M best individuals in each generation); *PBIL-M-Consensus* (the probability vector is moved only in the positions in which there is a consensus in all of the M best individuals in each generation); and *PBIL-Complement* (where the probability vector is moved towards the complement of the lowest evaluation individual -the worst sample- in each generation).

The results obtained in the resolution of all the problems (table 1) are shown in figures 3, 4 and 5 and they are summarized in table 2.

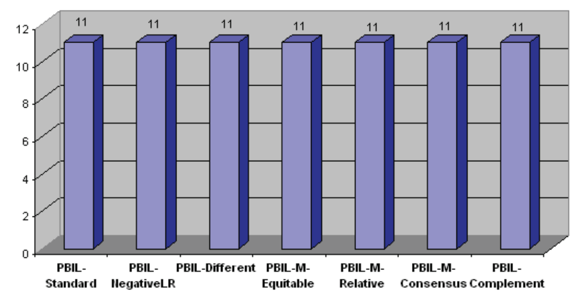

Figure 3. Frequency assignment span for the problem 1

As we can see in figure 3, all the different versions of PBIL studied obtain the optimum solution when the problem 1 is solved. With the first 11 frequencies the frequency planning can be done perfectly with no interferences. Problem 1 is clearly the simplest problem in table 1, for this reason, all the variants of PBIL obtain very good solutions.

Figure 4.a summarizes the results obtained in the problem 2. In this figure we can observe that the PBIL-Standard and the different variants of PBIL-M are the ones which obtain the best results. Using these variants, only 383 frequencies are used. This number is very near to the theoretical lower bound (381 -optimal result-). The results obtained in the resolution of problem 3 are shown in figure 4.b. In this case, all versions offer the same optimum results (533 frequencies to perform the frequency planning with no interferences).

For the problem 4 (fig. 4.c), it is displayed that the best versions are: the standard one (541), and all the M solutions (541); and the worst: NegativeLR, Different and Complement solutions. Again, the best versions are very close to the corresponding lower bound (533).

Figure 5.a summarizes the results obtained with problem 5. In this case, the solutions standard, M-equitative and M-relative (222) give us almost the optimum result (remember that the lower bound is 221, please, see table 1).

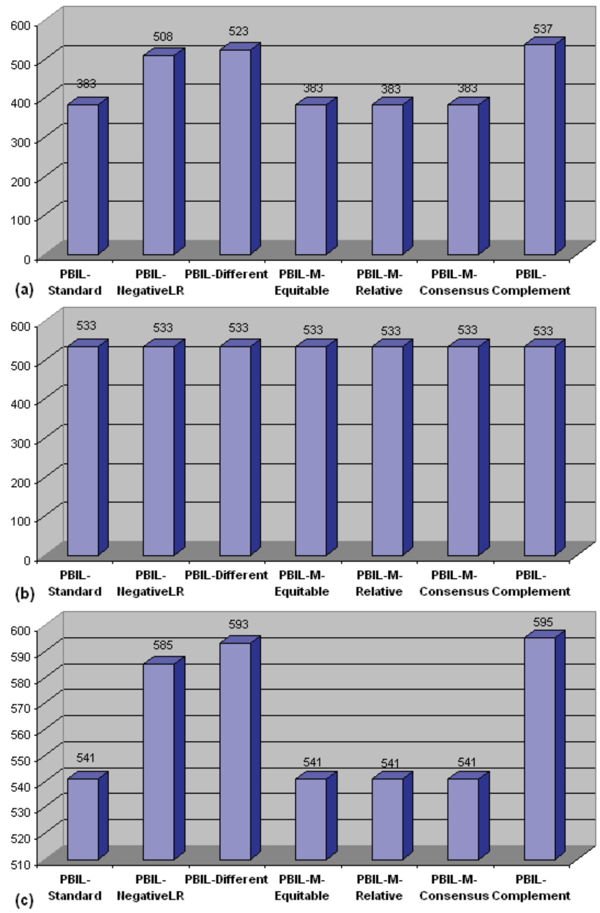

Figure 4. Frequency assignment span for the problems 2 (a), 3 (b) and 4 (c)

The results obtained in the resolution of problem 6 are shown in figure 5.b. We can see that the optimum results to solve this problem are obtained by the standard and the M-solutions (309).

Finally, to solve the problem 7 (figure 5.c) the best solutions of FAP are the standard and the M solutions, with values between 346 and 350 frequencies needed, however, in this case, these values are not very good, because the lower bound for this problem is 309.

6. Conclusions and future work

This paper describes the usage of different variations of PBIL to solve the FAP problem. This is a very important problem in current GSM networks. We

Authorized licensed use limited to: University of Johannesburg. Downloaded on March 02,2010 at 05:14:41 EST from IEEE Xplore. Restrictions apply.

have performed a set of experiments with 7 different types of problems to check which PBIL algorithm is the best alternative to solve the FAP problem.

After studying the results, summarized in table 2, we can conclude that the best versions of PBIL for solving this kind of problem are the standard and the M-solutions, because they obtain for all the experiments the best results and even the optimum values in almost all the studied cases. On the other hand, the worst versions of PBIL are NegativeLR, Different and Complement, which, although are recommended in the literature [6], obtain quite bad results with problems 2, 4, 5, 6 and 7. We have to say that these three last versions do not work only with the information about the best individual in the population, but as well with data from the worst individual. In our opinion, this is the cause of their bad results, because we think that this technique do not work very well in the advanced stages of the search, where the best and the worst individuals may be very similar.

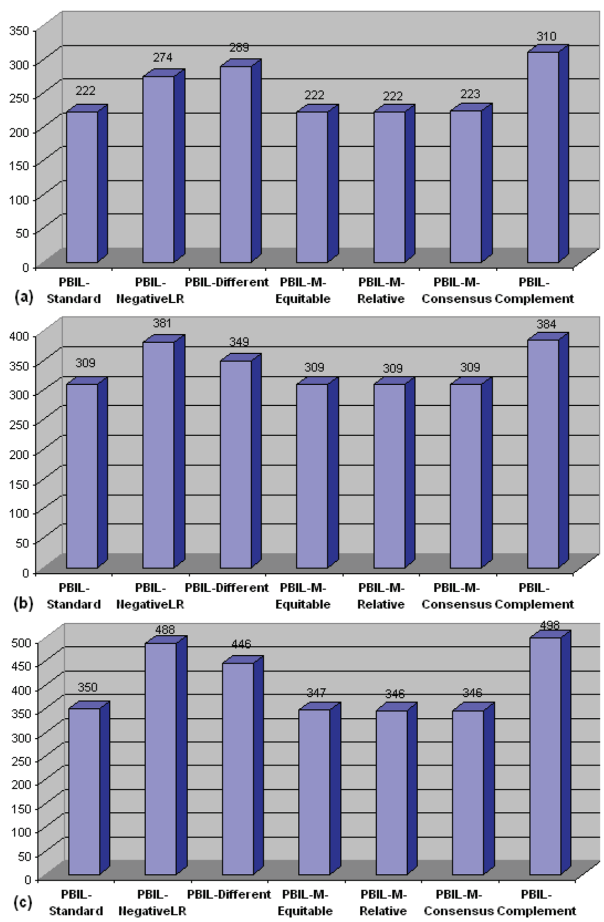

Figure 5. Frequency assignment span for the problems 5 (a), 6 (b) and 7 (c)

Anyway, it has been clearly proved that PBIL algorithm is a valid option to solve the MS-FAP problem, and this is important, because to the best of our knowledge, this is the first time that PBIL is employed for this task.

Future work includes the resolution of other types of FAP problems using PBIL algorithm and the study of other evolutionary algorithms for the same purpose in order to perform comparative studies.

Table 2. Results summary

	P1	P2	P3	P4	P5	P6	P7
Lower bound	***11***	***381***	***533***	***533***	***221***	***309***	***309***
PBIL-Standard	**11**	383	**533**	541	222	**309**	350
PBIL-NegativeLR	**11**	508	**533**	585	274	381	488
PBIL-Different	**11**	523	**533**	593	289	349	446
PBIL-M-Equitable	**11**	383	**533**	541	222	**309**	347
PBIL-M-Relative	**11**	383	**533**	541	222	**309**	346
PBIL-M-Consensus	**11**	383	**533**	541	223	**309**	346
PBIL-Complement	**11**	537	**533**	595	310	384	498

Acknowledgements

This work has been partially funded by the Spanish Ministry of Education and Science and FEDER under contract TIN2005-08818-C04-03 (OPLINK project). José M. Chaves-González is supported by the research grant PRE06003 from Junta de Extremadura (Spain).

7. References

[1] Wireless Intelligence. http://www.wirelessintelligence.com/, 2006.

[2] Aardal, K.I.; et al. Models and Solution Techniques for Frequency Assignment Problems. 4OR, 1(4): 261-317, 2003.

[3] Blum, C. and Roli, A. Metaheuristics in Combinatorial Optimization: Overview and Conceptual Comparison. ACM Computing Surveys 35, pp: 268–308, 2003.

[4] FAP Web (Philadelphia instances). http://fap.zib.de/problems/Philadelphia, 2007.

[5] Shirazi, S.A.G., Amindavar, H. Fixed Channel Assignment Using New Dynamic Programming Approach in Cellular Radio Networks. Computers and Electrical Engineering, vol. 31, nº 4-5, pp. 303-333, June-July 2005.

[6] Baluja, S. Population-based Incremental Learning: A Method for Integrating Genetic Search based Function Optimization and Competitive Learning. Technical Report CMU-CS-94-163, Carnegie Mellon University, June 1994.

[7] Baluja, S. and Caruana, R. Removing the Genetics from the Standard Genetic Algorithm. Twelfth International Conference on Machine Learning, San Mateo, CA, USA, pp. 38-46, May 1995.

Authorized licensed use limited to: University of Johannesburg. Downloaded on March 02,2010 at 05:14:41 EST from IEEE Xplore. Restrictions apply.