A Parallel Evolutionary Algorithm applied to the Minimum Interference Frequency Assignment Problem

Gerardo Mora　　Cristian Perfumo　　Lucas Rojas

Facultad de Ingeniería
Universidad Nacional de la Patagonia San Juan Bosco
Argentina.

Sergio Nesmachnow

Centro de Cálculo, Instituto de Computación,
Facultad de Ingeniería, Universidad de la República
Uruguay.

Abstract

This article presents the application of a Parallel Evolutionary Algorithm to solve the Minimum Interference Frequency Assignment Problem (MI-FAP). This is a capital problem in the mobile telecommunication field, which proposes to find an assignation of a set of frequencies to minimize the communication interference. MI-FAP is a NP-Complete optimization problem; so traditional exact algorithms are useless for solving real-life problem instances in reasonable execution times. This work proposes to use a metaheuristic approach to find good quality solutions for real-life MI-FAP instances never faced before using Evolutionary Algorithms. Evaluation experiments performed on those real-life instances report promising numerical results for both serial and parallel models of the algorithm proposed. In addition, the parallel version shows high levels of computational efficiency, demonstrating a superlinear speedup behavior for the instances studied.

Keywords: Parallel Evolutionary Algorithm, Minimum Interference, Frequency Assignment Problem.

Destination: Workshop on Agents and Intelligent Systems.

1. Introduction

Nowadays, wireless (also referred as *mobile*) communication is having a huge impact on developed countries' way of life. Wireless communications use radio waves to transmit voice and data between devices, using the portion of the electromagnetic spectrum in which waves can be generated by alternating current fed to an antenna (the *radio spectrum* or *radio frequencies*).

Network resource management has emerged as a capital problem in wireless telecommunication, due the fast development of wireless network infrastructures in the last fifteen years. Frequencies in the radio spectrum are a high cost and at the same time a scarcely available resource, so reutilization is a usual technique. When several connections share or use near frequencies, the communications suffer signal *interference*, a phenomenon that downgrades the transmission quality. The Frequency Assignment Problem (FAP) proposes to find an optimum assignation of a set of available frequencies to a large number of transmitter devices in order to satisfy certain constraints while optimizing some measure related with the communication quality (usually the goal consists in avoiding or at least minimizing the interference).

In its generic formulation, the FAP is a NP-Complete optimization problem [9]. Since the size of the existing wireless communication networks is continuously enlarging, the underlying instances of FAP frequently pose a challenge to classic assignment algorithms. In consequence, the research community has been searching for new methods that are able to replace and improve over to the traditional exact ones, whose low efficiency often makes them useless for solving real-life problems of large size in reasonable times. In this sense, heuristic algorithms have been applied to frequency assignment problems. Although they could sometimes fail in computing a true optimum for the problem, they are able to find appropriate quasi-optimal solutions in reasonable times. Among a whole new set of heuristics and modern optimization techniques, Evolutionary Algorithms [13] have emerged as flexible and robust methods for solving the underlying complex optimization problems found in telecommunications as well as in many other areas of application.

This article presents the application of a parallel evolutionary algorithm to the FAP, for solving real-life wireless network instances never faced before applying evolutionary techniques. The parallel evolutionary approach has been designed in the aim of solving the Frequency Assignment Problem with numerical accuracy and high computational efficiency.

The manuscript is structured as follows. Next section describes Evolutionary Algorithms. Section 3 presents the FAP, its mathematical formulation, and popular variants. Section 4 contains an overview of previous works related to EAs applied to solve the FAP. Section 5 describes the features of the algorithm used and their implementation details. The discussion on the experiments and results are summarized on Section 6, while conclusions and future work are formulated in Section 7.

2. Evolutionary Algorithms

Evolutionary Algorithms (EAs) are stochastic search methods that have been successfully applied in many real applications of high complexity.

An EA is an iterative technique that applies stochastic operators on a pool of individuals (the *population*) in order to improve their *fitness*, a measure related to the objective function. Every individual in the population is the encoded version of a tentative solution. Initially, this population is randomly generated. An evaluation function associates a fitness value to every individual indicating its suitability to the problem.

Iteratively, the applications of operations like recombinations of parts of two individuals (crossovers), or random changes (mutations) in their contents are guided by a selection-of-the-best technique to tentative solutions of higher quality. A particularly popular type of EA is the Genetic Algorithm (GA), in which all the mentioned operators are included.

Figure 1 presents a generic schema for an Evolutionary Algorithm.

```
Initialize(Population(0))
generation = 0
while (not StopCriteria) do
  Evaluate(Population(generation))
  Fathers = Selection(Population(generation))
  Offsprings = Reproduction Operators(Fathers)
  NewPop = Replace(Offsprings,Population(generation))
  generation ++
  Population(generation)=NewPop
return Best Solution Found
```

Figure 1: Schema for an Evolutionary Algorithm.

2.1. Genetic Algorithm

Goldberg [8] introduced the classical formulation of a GA. Based on the generic schema in Figure 1, the GA `Reproduction Operators` include *recombination* and *mutation*, applied to the population in each generation. GA techniques are widely spread due its versatility for solving combinatorial optimization problems [13].

2.2. CHC Algorithm

The CHC acronym stands for "*Cross generational elitist selection, Heterogeneous recombination, and Cataclysmic mutation*" [7]. CHC is a specialization of a traditional GA that incorporates a high conservative selection strategy, perpetuating the *k* better individuals over generations. CHC does not use mutation, and a special crossover operator (Uniform Crossover – HUX) is introduced: it randomly swaps exactly half of the bits that differ between the two parent strings. A mating restriction policy avoids recombining "too similar" individuals: only those parents that differ from each other by some number of bits are allowed to reproduce.

CHC does not use traditional mutation, but introduces new diversity by a re-initialization procedure using the best individual found so far as a template for creating a new population after convergence is detected (i.e., when no offspring can be inserted after a number of generations). The initial threshold for allowing mating is often set to 1/4 of the chromosome length. If no offspring is inserted into the new population during the mating procedure, this threshold is reduced by 1.

Figure 2 presents a pseudo-code for the CHC algorithm.

```
Initialize(Population(0))
generation = 0
distance = ChromosomeLength/4
while (not StopCriteria) do
  Evaluate(Population(generation))
  Fathers = Selection(Population(generation))
  Offsprings = HUX(Fathers)
  Evaluate(Offsprings)
  NewPop = Replace(Offsprings,Population(generation))
  if (NewPop == Population(generation)) then
    distance --
  generation ++
  Population(generation) = NewPop
  if (distance == 0) then
    Reinitialize(Population(generation))
    distance = ChromosomeLength/4
return Best Solution Found
```

Figure 2: Schema for the CHC algorithm.

2.3. Parallel Evolutionary Algorithms

In the last decade, parallel implementations became popular with the goal of making EAs more efficient. By splitting the population into several processing elements, parallel evolutionary algorithms (PEAs) allow to reach high quality results in reasonable execution times even for hard-to-solve optimization problems [3]. In this work we use a parallel CHC implementation categorized within the "subpopulation with migration" model according to the classification from Nowostaski and Poli [14]. The original population is divided in several subpopulations (called *demes*) separated geographically from each other. Each deme runs a serial EA, so individuals are able to interact only with other individuals in the deme. An additional *migration* operator is defined: occasionally some selected individuals are exchanged among demes, introducing a new source of diversity in the EA.

Figure 3 shows the generic schema for a subpopulation with migration PEA. Two conditions control the migration procedure: *SendMigrants* determines when the exchange of individuals takes place, and *ReceiveMigrants* establishes whether a foreign set of individuals has to be received or not. Even though in an *asynchronous* PEA these two conditions are separated in time, they coincide in a synchronous model, when the send and receive operations are executed synchronically, one just after the other. *Migrants* denote the set of individuals to exchange with some other deme, selected according to a given policy. We have explicitly distinguished between *Selection for reproduction* and *Selection for migration*; because they usually follow different policies. The *SendMigration* and *ReceiveMigration* operators carry out the exchange of individuals among demes according to a connectivity graph defined over them, most usually a unidirectional ring.

```
Initialize(Population(0))
generation = 0
Evaluate(Population(0))
while (not StopCriteria) do
  Fathers = Selection for Reproduction(Population(generation))
  Offsprings = Reproduction(Fathers)
  NewPop = Replace(Offsprings,Population(generation))
  generation ++
  Population(generation)=NewPop
  if (SendMigrants)
    Migrants = Selection for Migration(Population(generation))
    SendMigration(Migrants)
  if (ReceiveMigrants)
    Inmigrants = ReceiveMigration()
    P(generation) = Insert(Immigrants, P(generation))
return Best Solution Found
```

Figure 3: Schema for a subpopulation with migration Parallel Evolutionary Algorithm.

3. Frequency Assignment Problems

Wireless communications use the radio frequency spectrum as the medium for the information to pass trough. Since the governments charge the communication companies for using the spectrum, one of the main tasks of networks designer is to split the range of frequencies in channels to perform one communication per each (the terms *channel* and *frequency* are often used as synonyms).

The main advantage of wireless over wired networks is the possibility to tolerate the mobility of both transmitter end receiver, a feature that significantly increase the interference probability. Interference measures the ratio between signal and noise in a communication. *Low-level interferences* allow the receiver to distinguish the whole message in a clear way. On the other hand, if the interference level is big enough to unable the receiver to understand the message, turning the communication impossible, it is called *unacceptable interference*.

To be in presence of interference, the next situations between two transmitters have to happen simultaneously [9]:

1. Physical proximity and enough power to cover the same area.
2. Usage of the same (co-channel interference) or nearly (adjacent-channel interference) frequencies.

The concept of *restriction* is introduced to avoid unacceptable interference values. A restriction can involve either the separation between channels or channels that cannot be used. This last class of restrictions is because of technical or governmental limitations.

Summarizing, FAP can be defined as the problem to assign frequencies from a limited spectrum to a set of wireless transmitters maximizing the channel reuse and avoiding unacceptable interferences. Therefore, an optimal FAP solution implies balancing frequency reuse and communication quality.

3.1. FAP Taxonomy

Koster [12] proposed a categorization for the different flavours of FAP. Considering the objective to optimize as classification criteria, he identified four main categories:

- Minimum Order Frequency Assignment Problem (MO-FAP): the goal is to minimize the number of used channels.
- Minimum Span Frequency Assignment Problem (MS-FAP), the goal is to minimize the difference between the highest and the lowest used frequencies.
- Minimum Blocking Frequency Assignment Problem (MB-FAP), the goal is to minimize the blocking probability. This objective is reached by using partial assignations and dynamic evaluation of the rest of frequencies to assign, depending on the possibility of blocking in the network communications.
- Minimum Interference Frequency Assignment Problem (MI-FAP), the goal is to minimize the total sum of interferences.

This paper focuses on the last class of problems. A MI-FAP formal description is presented in the following subsection.

3.2. MI-FAP description and mathematical formulation

As described in the previous subsection, MI-FAP is aimed to assign frequencies from the spectrum to the transmitters within the network, maximizing frequency reuse, avoiding unacceptable interference and minimizing the sum of all low-level interferences involved in the scenario.

Every FAP problem can be modeled by a graph $G=(V,E)$, known as either *interference graph* or *restriction graph*, where:

- V is the set of graph vertices corresponding to nodes of wireless scenario.
- E is the set of graph edges. For two nodes $v,w \in V$, $(v,w) \in E$ if and only if the nodes v and w signals can suffer interference in at least in one pair of frequencies.
- $\forall v \in V; D_v \subseteq D$ is the available frequencies set for node v.
- c_v is the number of required frequencies by $v \in V$.
- p_{vwfg} quantifies the interference between $f \in D_v$ and $g \in D_w$ frequencies.
- p_{vvfg} quantifies the interference between f and g frequencies, both assigned to node v.

Specifically, the mathematical formulation for the MI-FAP version is presented in Figure 4.

Instance:

1. An undirected graph $G=(V,E)$; $\{v,v\}\in E;\forall v\in V$.
2. Sets $T_{vw}\subset \mathrm{Z},\{v,w\}\in \mathrm{E},0\in \mathrm{T}_{vw}$, of the relevant distances between $f(v)$ and $g(v)$ frequencies.
3. A demands set $c_v \in \mathrm{Z}^+ \ \forall\ \mathrm{v}\in \mathrm{E}$.
4. An available frequencies subset $D_v \subseteq Z^+,\forall v\in V$. $D_v = \{f/f$ is an available frequency for $v\}$.
5. Set $D=\bigcup_{v\in V} D_v$ conformed by all available frequencies for some node within the wireless scenario.
6. A set of penalization values $p_{vwfg}\in Z^+ \ \forall\ \{v,w\}\in E$, $f\in D_v, g\in D_w$.
7. A positive integer K that specifies the maximum acceptable value for the total of all penalizations within the scenario.

Question: Does it exist a frequencies assignation $f:V\rightarrow 2^D$ that satisfies the following conditions?

1. $|f(v)|=c_v$
2. $f(v)\subseteq D_v$
3. $\left(\sum_{\{v,w\}\in E}\ \sum_{\substack{\bar{f}\in f(v),\bar{g}\in g(w)\\ (v\neq w)\vee(\bar{f}\neq\bar{g})}} p_{vw\bar{f}\bar{g}}\,\delta\left(\left|\bar{f}-\bar{g}\right|\in T_{vw}\right)\right)\leq K$ ($\delta(A)=1$ if A is true or 0 otherwise).

Figure 4: MI-FAP formal description.

3.3. COST259 scenarios

Mobile telephony network design implies solving the frequency assignment problem. To perform this task, a representation of the real environment (named *scenario*) is needed. Since some communication companies consider this information confidential, it is typically difficult getting real descriptions and lot of papers have been written performing tests on random-generated instances.

Fortunately, some organizations have looked beyond these commercial limitations recently. By making available real information, they have allowed algorithms designers to test their solving techniques over an increasing number of real-life scenarios. In this work, COST 259 scenarios, published by the communications branch of European Cooperation in the field of Scientific and Technical research, have been used. Scenarios are described in Section 6.2 and their details are presented in Table 1.

4. Related work

This section briefly reviews several references reporting some work on applying EAs to frequency assignment problems.

Crompton et al. [4] were pioneers in applying parallel genetic algorithms to the FAP. In their early work, the authors introduced R2, one of the three most popular mechanisms for encoding FAP solutions, and an alternative representation grouping together those sites using the same frequencies. Computational results obtained using a PEA with migration scheme to solve an unreal (but realistic) scenario with 272 transmitters showed that the improved order representation is able to reach better assignations in terms of fewer constraints violations.

Dorne and Hao [5] faced the FAP with a special EA that had no crossover, but just mutation. The authors tested the algorithm with a set of 18 real problems in France. To improve the quality of results they hybridized the algorithm by introducing some local search features. The experimental phase showed very encouraging results, particularly because the solutions found are better than those obtained with simulated annealing and constraint programming.

Karaoglu and Manderick [11] presented a FAP solving approach, grouping transmitters in clusters. The proposal uses a cluster discovery algorithm based on the concept of grouping those closer transmitters in order to assign a different frequency to each one within the cluster. This assignation reduces the interference calculation to inter-cluster transmitters since intra-cluster interference is, by definition, zero. Karaoglu and Manderick got an algorithm roughly 33 times faster than a traditional GA applied to this problem. They attributed this significant speed-up to the reduction of both the search space and the time complexity of fitness calculation.

The proposal from Aardal et al. [1] resumed different models and solutions to all FAP flavors. The authors included a Genetic Algorithm section, presenting different approaches to the problem using this technique. GA-based FAP solving proposals are classified depending on the kind of FAP faced (MO-FAP, MS-FAP, MB-FAP, MI-FAP). In addition, the authors included a description of the more popular GA representations for FAP solutions (named R1, R2 and R3).

The work of Weinberg et al. [16] presented COSEARCH, a co-evolutionist optimization method, which combines different and complementary metaheuristics, such as tabu search and genetic algorithms. All metaheuristic algorithms run in parallel and cooperate with the others, using an adaptive memory procedure. Weinberg et al. introduced a new encoding proposal and presented two new crossover mechanisms according to their original representation of solutions. The authors tested the COSEARCH metaheuristic on several benchmark problems provided by France Telecom.

Finally, it is worth to note that there are no publications on evolutionary algorithms solving the Frequency Assignment Problem for the COST259 scenarios [6].

5. Implementation

5.1. Problem encoding

Each chromosome encodes a frequency assignment for a mobile telephony network placed in some fixed area (i.e. a city). Aardal R1 encoding [1] was adopted due its conceptual simplicity and the possibility of applying standard evolutionary operators.

R1 is the most intuitive encoding scheme for frequency assignments. It is based on using a n dimension integer vector, where n is the number of transmitters in the scenario. The element in the position i represents the frequency assigned to the transmitter i. Figure 5 presents an example of R1 encoding, where the frequency number 16 is assigned to the first transmitter, the frequency number 5 to the second transmitter, and so on.

16	5	9	13	15	7	11	16	7	12	6	10

Figure 5: Solution encoding example (R1).

5.2. Fitness function

MI-FAP is a minimizing optimization problem, so the fitness function F to evaluate an individual x was designed as a quotient, whose expression is presented in Equation 1. The dividend evaluates the sum of total interference in the network plus the sum of each violated restriction, plus 1 (to avoid division by zero). A restriction violation can be originated either from not respecting a minimum span between two frequencies or from using a channel that is not available (a blocked channel).

$$F(x) = \frac{1}{1 + \sum_{v \in V; a \notin D_v} \delta(a \in f(v)) + \sum_{v \in V; a \in D_v} \sum_{w \in V; b \in D_w} p_{vwab} \delta(|a-b| \in T_{vw}) + \sum_{v \in V; a \in D_v} \sum_{w \in V; b \in D_w} p_{vwab} \delta(a \in f(v) \wedge b \in g(w))}$$

Equation 1: MI–FAP fitness function.

The fitness function has to be designed in such a way that a solution with high level of interference and low number of restriction violated has better fitness value than another with low interference and high violations. That is because in MI-FAP context, only one violated restriction could invalidate the whole assignment, because it disallows the communication between two points in the network. Therefore, it is desirable to reach a final solution without violated restrictions. To satisfy this goal, the fitness function in Equation 1 proposes using a lineal relation among the terms within the divisor. The interference level between two transmitters is quantified in a real number between 0 and 1, and each violated restriction is computed as 1. Other fitness proposals have been studied in non-formalized experiments, yielding to similar results than the function in Equation 1.

5.3. The MALLBA library

The MALLBA project [2] is an effort to develop a library of optimization algorithms able to deal with parallelism, in a user-friendly and, at the same time, efficient way both for LAN and WAN environments. Optimization algorithms are implemented as *software skeletons* on the library. Skeletons are generic templates that the user has to instantiate with the features of the specific problem to solve. They incorporate all the knowledge related to the resolution method, its interactions with the problem, and the parallel execution. Skeletons are implemented by a set of *required* and *provided* C++ classes that represent an abstraction of the entities participating in the resolution method:

- Provided classes implement skeletons internal features, in a problem-independent way. The most important provided classes in MALLBA are `Solver` (the algorithm) and `SetUpParams` (parameters setup).
- Required classes specify information related to the problem. Each skeleton includes the `Problem` and `Solution` required classes, which encapsulate the problem-dependent entities needed by the resolution method. Depending on the skeleton, other classes may be required.

The infrastructure used in the MALLBA project is made of communication networks and clusters of computers located in Málaga, La Laguna and Barcelona, in Spain. A chain of Fast Ethernet and ATM circuits interconnects these nodes. The MALLBA library is publicly available at University of Málaga location **`http://neo.lcc.uma.es/mallba/easy-mallba`**.

The CHC algorithm used in this work has been codified using the MALLBA library.

6. Experimental results

This section describes the experiments performed using the CHC algorithm to solve the MI-FAP. Tests were mainly aimed to compare the quality of reached solutions and the computational efficiency (evaluating the time spent to obtain them) for both sequential and parallel version of the CHC algorithm.

6.1. Execution platform

All experiments were performed on a cluster of three AMD Athlon 3000 64 bits. Each cluster node has the following features: 2 GHz clock frequency, 1 GB RAM and Open SuSE Linux 10 operating system. Connectivity among nodes was done through a Fast Ethernet LAN at 100 Mbps.

6.2. Test instances

Evaluation experiments were performed over three COST259 instances, whose most important features are described in Table 1. The *Tiny* test instance was used only for the correctness evaluation of the algorithm implementation (since it does not pose a challenge as combinatorial optimization problem due its reduced size), while *K* and *Swiscomm* arise from real-life, medium-size wireless networks on dense urban environments provided by telecommunication companies [6].

Scenario	***Cells***	***Transmitters***	***Spectrum width***
Tiny	7	12	13
K	264	267	50
Swisscom	148	309	68

Table 1: Scenarios features.

6.3. CHC configuration

Configuration experiments were performed to find the more suitable parameter combination able to reach good quality solutions without increasing drastically the execution times. Table 2 shows the parameter set that allowed reaching the best solutions in the configuration experiments. The same configuration was used for both serial and parallel CHC algorithm, but the parallel version splits the population among three demes.

Parameter	***Values***	
	Sequential GA	***Parallel GA***
Population size	150	50
Crossover probability	0.7	0.7
Reinitialization probability	0.2	0.2
Stopping criterion	30000 generations	30000 generations

Table 2: CHC parameters.

6.4. Evaluation experiments

Evaluation experiments involved performing 10 independent executions of sequential and parallel CHC per scenario. Table 3 summarizes the results achieved for each test instance studied.

	Sequential CHC		***Parallel CHC***	
Scenario	***Average Fitness***	***Standard Deviation***	***Average Fitness***	***Standard Deviation***
Tiny	0,98039	0	0,97944	0,00300
K	0,30690	0,02306	0,29852	0,01447
Swisscom	0,00066	0,00031	0,00065	0,00011

Table 3: Comparative results for serial and parallel CHC.

Analyzing the quality of reached solutions showed in Table 3, it can be seen that the parallel version is not able to improve the results achieved by the serial algorithm. Both CHC versions reach the same results quality, since the difference between sequential and parallel versions is not significant statistically, because it is lower than the standard deviation on fitness values.

The evolutionary approach was unable to achieve the solution quality obtained with other metaheuristic techniques (e.g. Simulated Annealing [10] and Tabu Search [15]). However, considering that the CHC algorithm does not include problem dependant information and uses simple evolutionary operators, the accomplished results could be considered as promising, and eloquently locates the evolutionary techniques as those useful to solve the MI-FAP.

Table 4 presents the computational efficiency analysis for both CHC versions. Experiments were aimed to evaluate the execution times, and then calculate the *speedup* and *efficiency* measures. The *speedup* (S_M) relates the mean execution time demanded when using a single processor (T_1) with the mean execution time when using M processors (T_M), as Equation 2.1 states. Equation 2.2 defines the *efficiency* (E_M), which normalize the speedup value, considering the computational resources used.

$$S_M = \frac{T_1}{T_M} \quad (2.1) \qquad\qquad E_M = \frac{S_M}{M} \quad (2.2)$$

Equation 2: Speedup and efficiency definitions.

Focusing on the execution times, the computational efficiency for parallel CHC algorithm was significantly better than its sequential counterpart. As a matter of fact, in the most relevant scenarios (those that model real-life wireless networks) CHC showed a superlinear speedup behavior and, consequently, computational efficiency was greater than one. (Results on *Tiny* scenario are not representative to qualify the algorithm in real networks, since it is a very small instance created in order to teach the researchers about the file format and algorithm correctness).

Scenario	***Sequential CHC***		***Parallel CHC***		***Speedup***	***Computational Efficiency***
	Average Time Spent (hours)	***Standard Deviation***	***Average Time Spent (hours)***	***Standard Deviation***		
Tiny	0,136	0,012	0,048	0,003	2,8	0,93
K	16,530	0,174	4,649	0,030	3,55	1,18
Swisscom	26,348	0,330	8,410	0,047	3,13	1,04

Table 4: Computational efficiency comparison.

Table 4 eloquently demonstrates that splitting the population significantly reduces the execution times. Solving the MI-FAP for *K* and *Swiscomm* scenarios demand large execution times for the serial CHC algorithm (it demands more than 24 hours for *Swisscom* instance). However, employing a multi deme parallel version can notably diminish this amount of time.

Figure 6 presents a graphical analysis that tracks the fitness evolution for a representative single run of serial CHC for solving the *K* scenario. The fitness values show that CHC is able of achieve well-suited individuals in a relatively low number of generations, and it also demonstrates the CHC capability of avoiding premature convergence (i.e. fitness values show a increasing derivative). A similar behavior was detected for CHC parallel version as well as for experiments performed to solve the *Swisscom* scenario.

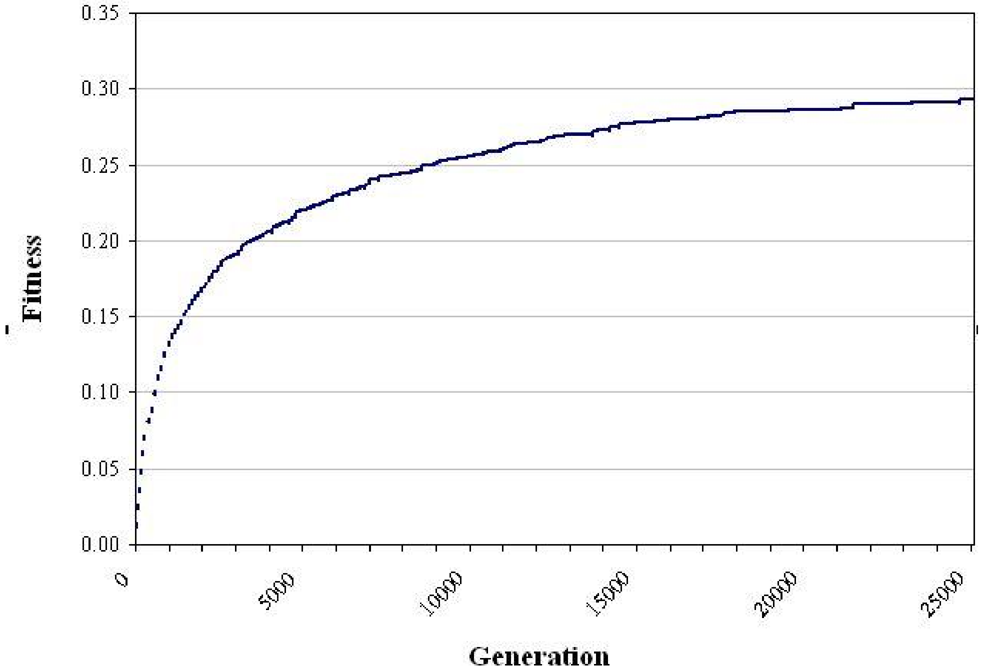

Figure 6: Fitness evolution sample (CHC serial version, K scenario).

7. Conclusions and future work

This article presents the resolution of MI-FAP using serial and parallel versions of CHC evolutionary algorithm. The work is focused on solving a set of scenarios never faced before using evolutionary techniques.

Even though using simple evolutionary operators the CHC algorithm is not able to outperform the solution quality obtained with other techniques, the parallel evolutionary approach showed itself as competitive to reach good results in reasonable execution times.

From the results and efficiency comparison, it is possible to conclude that the parallel CHC version allows reaching similar values of solutions quality than the sequential version, but considerably reducing the execution times. The computational efficiency is a promising feature, because superlinear speedup behaviour was detected.

Using R1 encoding and traditional non-specific evolutionary operators, CHC was unable to achieve results obtained by Simulated Annealing and Tabu Search methods [10] [15]. This situation lead to many possible approaches to be studied in future work. The main proposal involves trying to improve the quality of solutions using *ad hoc* evolutionary operators considering problem dependant information. Related to this topic, it is also important experimenting with other encodings such as traditional R2 and R3 or even original proposals. Other options include using different fitness function trying to reach a balance between exploring in early generations and exploiting in advanced ones. Due the differences existing among test scenarios, it is worth to try designing a dedicated fitness function for each group of scenarios with similar features.

Regarding the computational efficiency, other models of parallel evolutionary algorithms should be studied. A comparative study analyzing scalability, speedup and efficiency will be worth to reach even better results after implementing some of the problem-dependant modifications mentioned above. Since the computational resources available limited the experiments, the scalability of parallel versions should be further investigated, to determine whether it would be useful for solving complex scenarios using the power of large clusters of computers.

Finally, considering the total interference and the separation and blocked channel restrictions as three different independent objectives, it is possible to face the problem following a multi-objective evolutionary algorithm approach. We are working on some of these topics right now.

References

[1] Aardal, K., Van Hoesel, S., Koster, A., Mannino, C., Sassano, A. *Models and Solution Techniques for Frequency Assignment Problems.* Konrad-Zuse-Zentrum für Informationstechnik, Berlin, 2001.

[2] Alba E., Cotta, C. *Optimización en entornos geográficamente distribuidos. Proyecto MALLBA.* Actas del Primer Congreso Español de Algoritmos Evolutivos y Bioinspirados, pp. 38–45, 2002 (text in Spanish).

[3] Alba E., Tomassini M. *Parallelism and Genetic Algorithms.* IEEE Transactions on Evolutionary Computation **6** (5), pp. 443–462, 2002.

[4] Crompton, W., Hurley, S. Stephens, N. *A parallel genetic algorithm for frequency assignment problems*, Proceedings of the IMACS/IEEE Conference on Signal Processing, Robotics and Neural Networks, Lille, France, pp. 81–84, 1994.

[5] Dorne, R., Hao, J. *An evolutionary approach for frequency assignment in cellular radio networks*, IEEE International Conference on Evolutionary Computing, Perth, Australia, pp. 539–544, 1995.

[6] Eisenblätter, A., Koster, A. COST 259. *Wireless Flexible Personalized Communications.* Available online at `http://fap.zib.de/problems/COST259/`. Consulted in June 2006.

[7] Eshelman, L. *The CHC adaptive search algorithm: How to have safe search when engaging in nontraditional genetic recombination.* Foundations of Genetic Algorithms, pp. 265–283, Morgan Kaufmann, 1991.

[8] Goldberg, D. *Genetics Algorithms in search, optimization and machine learning.* Addison-Wesley, 1998.

[9] Hale, W. *Frequency assignment: theory and applications.* Proceedings of the IEEE **68** (12), pp.1497–1514, 1980.

[10] Jordán de Urries, L., Diaz Guerra, M., Berberana, I. *Frequency Planning using Simulated Annealing*. Technical Report COST 259, Bergen Kongress Senter, Bergen, April 2000.

[11] Karaoglu, N., Manderick, B. *FAPSTER – A Genetic Algorithm for Frequency Assignment Problem.* Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'2005), Washington, D.C., USA, 2005.

[12] Koster, A. *Frequency Assignment, Models and Algorithms.* Ph.D. Thesis, Universiteit Maastricht, Maastricht, The Netherlands, 1999.

[13] L. Davis. *Handbook of Genetic Algorithms*. Van Nostrand Reinhold, New York, 1991.

[14] Nowostawski, M., Poli, R. *Parallel Genetic Algorithm Taxonomy.* In Proceedings of the Third International Conference on Knowledge-based Intelligent Information Engineering Systems (KES'99), pp. 88–92, IEEE Computer Society, 1999.

[15] Smith, D., Moon, J., Montemanni, R. *An Improved Tabu Search Algorithm for the Fixed-Spectrum Frequency Assignment Problem*. IEEE Transactions on vehicular technology, **52** (3), pp. 891–901, 2003.

[16] Weinberg, B., Bachelet, V., Talbi, E. *A Co-evolutionist Meta-heuristic for the Assignment of the Frequencies in Cellular Networks*. Proceedings of the EvoWorkshops on Applications of Evolutionary Computing, pp. 140–149, 2001.