

Optimization algorithms for large-scale real-world instances of the frequency assignment problem

Francisco Luna · César Estébanez · Coromoto León · José M. Chaves-González · Antonio J. Nebro · Ricardo Aler · Carlos Segura · Miguel A. Vega-Rodríguez · Enrique Alba · José M. Valls · Gara Miranda · Juan A. Gómez-Pulido

Published online: 11 September 2010
© Springer-Verlag 2010

Abstract Nowadays, mobile communications are experiencing a strong growth, being more and more indispensable. One of the key issues in the design of mobile networks is the frequency assignment problem (FAP). This problem is crucial at present and will remain important in the foreseeable future. Real-world instances of FAP typically involve very large networks, which can be handled only by heuristic methods. In the present work, we are interested in optimizing frequency assignments for problems described in a mathematical formalism that incorporates actual interference information, measured directly on the field, as is done in current GSM networks. To achieve this goal, a range of metaheuristics have been designed, adapted, and rigorously compared on two actual GSM networks modeled according to the latter formalism. To generate quickly and reliably high-quality solutions, all

metaheuristics combine their global search capabilities with a local-search method specially tailored for this domain. The experiments and statistical tests show that in general, all metaheuristics are able to improve upon results published in previous studies, but two of the metaheuristics emerge as the best performers: a population-based algorithm (Scatter Search) and a trajectory based (1+1) Evolutionary Algorithm. Finally, the analysis of the frequency plans obtained offers insight about how the interference cost is reduced in the optimal plans.

Keywords Frequency assignment problem · Large-scale real-world instances · Metaheuristics · Optimal design

F. Luna · A. J. Nebro · E. Alba
Universidad de Málaga, Malaga, Spain
e-mail: flv@lcc.uma.es

A. J. Nebro
e-mail: antonio@lcc.uma.es

E. Alba
e-mail: eat@lcc.uma.es

C. Estébanez · R. Aler (✉) · J. M. Valls
Universidad Carlos III de Madrid, Madrid, Spain
e-mail: aler@inf.uc3m.es

C. Estébanez
e-mail: cesteban@inf.uc3m.es

J. M. Valls
e-mail: jvalls@inf.uc3m.es

C. León · C. Segura · G. Miranda
Universidad de La Laguna, Tenerife, Spain
e-mail: cleon@ull.es

C. Segura
e-mail: csegura@ull.es

G. Miranda
e-mail: gmiranda@ull.es

J. M. Chaves-González · M. A. Vega-Rodríguez · J. A. Gómez-Pulido
Universidad de Extremadura, Badajoz, Spain
e-mail: jm@unex.es

M. A. Vega-Rodríguez
e-mail: mavega@unex.es

J. A. Gómez-Pulido
e-mail: jangomez@unex.es

1 Introduction

The frequency assignment problem (FAP) is a well-known combinatorial optimization problem that arises as one of the main issues in the design of GSM (Global system for mobile communications) networks (Mouly and Paulet 1992). In the FAP, a small set of frequencies has to be allocated to the elementary transceivers (TRXs) installed in the base stations of the cellular network.

Solving the FAP is crucial for today's GSM operators both at the stage of the initial design and in the subsequent modifications of the network. Indeed, by mid 2006 GSM services were used by more than 1.8 billion subscribers (<http://www.wirelessintelligence.com/>) across 210 countries, representing approximately 77% of the world's cellular market. It is widely accepted that the third generation mobile telecommunication system (*Universal Mobile Telecommunication System* or UMTS) (Rapeli 1995), will coexist with the enhanced releases of the GSM standard (GPRS; Granbohm and Wiklund 1999 and EDGE; Furiuskar et al. 1999) at least in the first phases. GSM is then expected to play an important role as a dominating technology for many years. Therefore, the FAP will remain an important issue in the near future.

The FAP is a difficult optimal design task because the usable radio spectrum is very scarce and frequencies have to be reused throughout the network, and consequently, some inevitable degree of interference will occur. From a mathematical point of view, the FAP is a generalization of the graph coloring problem, and therefore is NP-hard (Hale 1980). However, from an engineering point of view, the basic FAP formulation must be extended to address real-world issues.

Initial formulations in the 1970s were very similar to the classic graph coloring problem, and assumed that adjacent frequencies do not interfere (only co-channel interferences were considered) (Aardal et al. 2007). However, real-world models must take into account all possible sources of interference, as well as regulatory concerns, and technological limitations (Eisenblätter 2001).

Nowadays, the large traffic demand and the reduced frequency spectrum make impossible to find interference-free frequency assignments. Therefore, most efforts in the current literature are intended to obtain frequency plans that minimize the overall interference of the network (i.e., maximizing the quality of service). However, most of the research deals with problems generated in a computer by sampling random variables. In the present work, we use a formulation (Luna et al. 2007a) aimed at taking full advantage of realistic and accurate interference information, directly measured on the field, obtained from real-world GSM networks.

Even though several exact algorithms have been proposed (Fischetti et al. 2000; Mannino and Sassano 2003), they are not feasible when dealing with large instances of the problem (Aardal et al. 2007), and therefore heuristic methods are mandatory. Among these heuristic methods, metaheuristics (Glover and Kochenberger 2003) have been shown to perform specially well when addressing FAP-like problems (Amaldi et al. 2006), and therefore they have been used in the present work.

According to the number of candidate solutions that any given technique manipulates at each iteration of the search, metaheuristics fall into two main categories (Blum and Roli 2003): trajectory- and population-based methods. This work, which aims at covering these two types of techniques, is a significant extension of the preliminary results of (Luna et al. 2007a, 2008), obtained within the OPLINK coordinated research project (<http://oplink.lcc.uma.es>) that involves teams of four Spanish universities. New experiments have been carried out, a new real-world instance has been tested, and the resulting frequency plans have been analyzed.

The present work is experimental in nature and has two main goals. First of all, to compute high-quality (low interfering) frequency plans for two real-world FAP instances. These two instances correspond to real-world data from GSM networks in two large cities: Denver and Seattle. Four selected metaheuristics have been carefully adapted to deal with our FAP formulation (Luna et al. 2007a). Our second goal is to carry out a rigorous testing and statistical comparison among the four metaheuristics.

To cover a wide range of algorithms, both population- and trajectory-based metaheuristics have been considered. We have also taken into account the degree of novelty involved: commonly used algorithms have been included, but we also wanted to test the effectivity of less common algorithms on the FAP. On the one hand, the literature research (Sect. 2) shows that Evolutionary Algorithms (EAs) have been frequently and successfully applied to the FAP and therefore, a steady-state Genetic Algorithm or SSGA (population-based) and a (1+1) EA (trajectory-based) have been included in the set of metaheuristics. On the other hand, Scatter Search (SS) and estimation of distributions algorithms (like PBIL; Baluja 1994) have proven very effective on combinatorial problems, but have been seldom used on the FAP. Thus, SS (population-based) and a newly developed trajectory-based version of PBIL, called Local Search with Heuristic Restarts (LSHR) have been added to the set of algorithms to be tested.

Companies in the telecommunication industry require that appropriate solutions are reached within short times (the GSM operator cannot wait for a long time to get a frequency plan). Therefore, all these algorithms have been

hybridized with a local search method specially tailored to this problem.

To ensure that all the algorithms were programmed in the most efficient way and following common guidelines, the coordinated work of the four teams has also included the supervision of the design and development of techniques by the other groups. Each group has been assigned the metaheuristic on which they were most expertise.

All the metaheuristics described in this paper will be rigorously compared. Great care has been put into this issue by using the same programming language, compiler options, and running all the algorithms under the same experimental conditions. Additionally, in order to improve the fairness of the comparison, the four algorithms tested in this paper use the same local search algorithm. The results show that all metaheuristics can handle large real-world FAP instances very efficiently and obtain very low interfering frequency plans. In particular, one of the instances used in this paper (Denver) had been tested in previous research (Luna et al. 2007a, 2008), and the results have improved significantly for that instance.

The organization of the paper is as follows. Section 2 elaborates on the work carried out on the FAP, and specifically, on the metaheuristic techniques related to the ones used in our work. In Sect. 3, we present the background of our FAP and the mathematical formulation used for its solution. In Sect. 4, the proposed metaheuristics are outlined. Finally, the experimental evaluation of the algorithms is statistically analyzed in Sect. 5, and then the relevant conclusions are drawn in Sect. 6.

2 Related work

The FAP was first introduced in the early 1970s by Metzger (1970), and since then, it has probably become one of the most studied problems in the operation research field because of its relation to both classic combinatorial optimization and real-world engineering problems. This large production has been analyzed and organized in several surveys and books (Aardal et al. 2007; Eisenblätter et al. 2002; Jaumard et al. 1999; Leese and Hurley 2002).

Depending on the spectrum size, the objectives, and the specific technological constraints, the FAP may assume very different forms. In Aardal et al. (2007), three main FAP models are described: MO-FAP (Minimum Order Frequency Assignment Problem), MS-FAP (Minimum Span Frequency Assignment Problem), and MI-FAP (Minimum Interference Frequency Assignment Problem). They have been chronologically appearing in the literature as technology, national regulators, and markets have been determining the working conditions.

MO-FAP is aimed at reducing the number of frequencies in a given cellular network (assuming that different frequencies do not interfere with each other), MS-FAP is targeted to find an assignment that minimizes the difference between the largest and the smallest frequency, i.e., the span (the regulators started licensing blocks of frequencies), and MI-FAP tries to minimize (a measure of) the overall interference of the network, because the increasing demand made the available frequency spectrum to be insufficient for obtaining interference-free frequency plans. MI-FAP is certainly the model of FAP mostly addressed in the recent literature mainly because of its direct applicability to the solution of large instances of real-world GSM frequency planning (actually, the scope of this work). Besides these three main models of FAP, other variants exist in the literature. Interested readers are referred to Aardal et al. (2007) for further details on this topic.

As stated before, all FAP models are NP-hard problems (Hale 1980). Even though several exact algorithms have been proposed (Avenali et al. 2002; Fischetti et al. 2000; Mannino and Sassano 2003), heuristic approaches are mandatory when tackling large instances of the problem (Aardal et al. 2007). Specifically, metaheuristics (Glover and Kochenberger 2003) have shown to provide the FAP problem with very accurate solutions (Amaldi et al. 2006). It is well known that the efficacy of many of these algorithms is mostly based on the usage of local search methods specially tailored to the version of the problem being addressed, i.e., they are hybrid metaheuristics (Talbi 2002; Raidl 2006). All the algorithms considered in this work fall into this class (they all are hybridized with a greedy algorithm). Because of the large number of contributions in the literature, we have been forced to restrict ourselves to review only those works that propose solution approaches similar to our algorithms, EAs, SS, and greedy local search endowed with restarting mechanisms, in this section.

Let us start analyzing the works using EAs. Since the number of contributions here is rather large, we have included a summary of their main features in Table 1. For each work, the table shows, respectively, the bibliographic reference, the publication year, the local search method (LS) used with the EA, the FAP model addressed, and the instances used together with its size (when this information is available).

The first issue we want to discuss has to do with the column LS. All the analyzed works but (Weinberg et al. 2001) (row 4, COSEARCH) incorporate a local search method within the evolutionary cycle of the EA (either together with the recombination and mutation operators or replacing any of them). By using the taxonomy and grammar proposed by Talbi (2002) to define hybrid metaheuristics, they are *LTH(EA(LS))* or Low-level Teamwork Hybrids, in which the EA is usually devoted to search

Table 1 Hybrid EAs (in the context of this work) used to solve the frequency assignment problem

References	Year	LS	FAP	Instances
Alabau et al. (2001, 2002)	2001	Greedy, TS	MI-FAP	Owner (5,700 TRXs)
Lau and Tsang (2001)	2001	GLS	MO-FAP	CALMA
Valenzuela (2001)	2001	Greedy	MS-FAP	Philadelphia
Weinberg et al. (2001)	2001	COSEARCH	MS-FAP	Owner
Mabed et al. (2002)	2002	TS	MI-FAP	Owner (639 TRXs)
Matsui et al. (2003)	2003	Greedy	MI-FAP	Owner
Raymond et al. (2003)	2003	GLS	MS-FAP	Philadelphia
Greff et al. (2004), Idoumghar and Schott (2008)	2004	MDP	MI-FAP	Owner (5,700 TRXs)
Matsui et al. (2005)	2005	Greedy	MS-FAP	Philadelphia
Salcedo-Sanz and Bousoño-Calzón (2005)	2005	Neural Network	MI-FAP	Philadelphia
Colombo (2006)	2006	Greedy	MI-FAP	Generated (1,667 TRXs)
Kim et al. (2007)	2007	CAP3	MS-FAP	Philadelphia
Luna et al. (2007b)	2007	Greedy	MI-FAP	Owner (2,612 TRXs)
Luna et al. (2007a)	2007	Greedy	MI-FAP	Owner (2,612 TRXs)
Luna et al. (2008)	2008	Greedy	MI-FAP	Owner (2,612 TRXs)

diversification and the LS mainly promotes search intensification of promising areas. As shown in Table 1, the LS methods used range from specific greedy algorithms engineered for the addressed problem to advanced search (meta)heuristics, such as CAP3, Markov Decision Processes (MDP), Guided Local Search (GLS) (Voudouris and Tsang 1999), Tabu Search (TS) (Glover 1998), and a Neural Network. The grammar defined in Talbi (2002) also allows three additional descriptors to be defined. Here, these descriptors are (*heterogeneous, global, general*) (interested readers are referred to Talbi 2002 for their definition). The *LTH(EA(LS))* scheme is specially effective for EAs solving the FAP problem since the search operators of this kind of metaheuristics are usually stochastic and they seldom generate highly accurate solutions (due to the high number of constraint violations). Without doubt, the usage of greedy algorithms (or *hill climbers*) as LS method is the most widely used strategy (included in 9 out of the 17 analyzed works) because of its ability of easily adding specific problem domain knowledge. COSEARCH (Weinberg et al. 2001) is the only metaheuristic that does not follow this approach. Indeed, it uses a parallel heterogeneous search model (Luna et al. 2005) in which a local search method and two metaheuristics (EA and TS) cooperate via an adaptive memory mechanism.

It can also be seen that the three main versions of the FAP problem have been successfully addressed (MO-FAP, MS-FAP, and MI-FAP), thus showing the suitability of hybrid EAs to solve this kind of problems. As to the instances tested (last column in Table 1), the Philadelphia benchmark (FAP Web: <http://fap.zib.de/>) is also widely adopted. However, it is composed of a very small-sized instance set using very abstract models of the real problem

(e.g., hexagonal cell shapes). Other sets of not publicly available instances have been addressed.

With respect to the SS literature, the utilization of this technique for solving the FAP problem is rather marginal. Indeed, the orthodox technique has been used only for the particularization of FAP, the graph coloring problem in Hamiez and Hao (2002). There are two works applying this SS technique, but they have to be considered as hybrid algorithms because they use typical evolutionary operators within the scatter search template (Luna et al. 2008; Chaves-González et al. 2008a). In all these two hybrid approaches, the problem version is the same as that used here.

The related works that hybridize greedy local search algorithms with restarting mechanisms to explore the search space of FAP are analyzed next. In general, local search has been widely used for solving this kind of problems (Tiourine et al. 2000) because of their ability to both quickly provide FAP instances with acceptable solutions and to easily incorporate specific problem knowledge in the search. To avoid these algorithms getting stuck in local optima, multi-start or restarting strategies (Martí 2003) have been used in the literature. The algorithms that better fit to this scheme are the GRASP metaheuristic (Greedy Randomized Adaptive Search Procedure) (Feo and Resende 1995). In the context of the FAP problem, GRASP has been used in Liu et al. (2000) and Gomes et al. (2001). Here, we are using LSRH (Local Search with Heuristic Restarts) (Luna et al. 2008) that uses a greedy algorithm based on PBIL (Baluja 1994) with a population size of 1. The solutions found by the greedy method are used to update the probability matrix of PBIL which is used in turn to restart the greedy search. PBIL has also been used to

address the FAP problem in Chaves-González et al. (2008a, b). Chaves-González et al. (2008a) showed very bad results for PBIL (using the same problem version that has been used here), while Chaves-González et al. (2008b) targeted to parallelize the technique rather than solving the problem actually.

Finally, this work is an extension of Luna et al. (2008). We have performed additional experiments and included a new real-world instance. The analysis of the results has also been substantially improved by deeply evaluating the resulting frequency plans reached by all the proposed algorithms.

3 Problem statement

This section is devoted to presenting relevant concepts to the frequency planning in GSM networks that will be used throughout this paper, as well as a detailed mathematical formulation of the problem addressed.

3.1 The GSM system

An outline of the GSM network architecture is shown in Fig. 1. As it can be seen, GSM networks are built out of many different components. The most relevant ones to frequency planning are the Base Transceiver Station (BTS) and the transceivers (TRXs). Essentially, a BTS is a set of TRXs. In GSM, one TRX is shared by up to eight users in TDMA mode. The main role of a TRX is to provide conversion between the digital traffic data on the network side and the radio signals, thus supplying communication between the mobile terminals and the GSM network. The site at which a BTS is installed is usually organized in sectors: one to three sectors are typical. Each sector defines a cell.

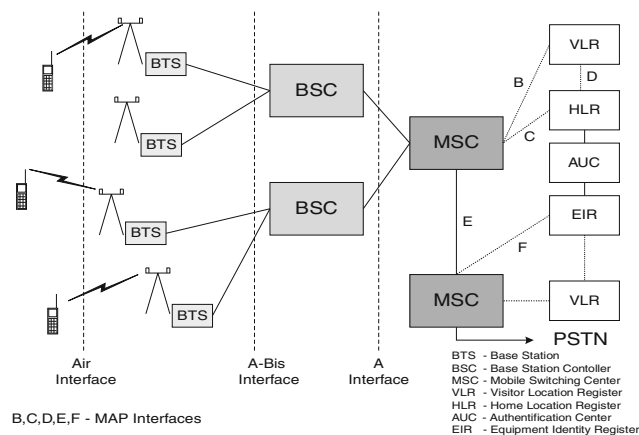


Fig. 1 Outline of the GSM network architecture

The solid lines connecting components in Fig. 1 carry both traffic information (voice or data) as well as the “in-band” signaling information. The dashed lines are signaling lines. The information exchanged over these lines is necessary for supporting user mobility, network features, operation and maintenance, authentication, encryption, and many other functions necessary for the network’s proper operation. Figure 1 shows the different network components and interfaces within a GSM network.

3.2 Frequency planning in GSM

The frequency planning is the last step in the layout of a GSM network. Prior to tackling this problem, the network designer has to address some other issues: where to install the BTSs or how to set configuration parameters of the antennae (tilt, azimuth, etc.), among others (Mishra 2004). Once the sites for the BTSs are selected and the sector layout is decided, the number of TRXs to be installed per sector has to be fixed. This number depends on the traffic demand that the corresponding sector has to support. Frequency planning lies on the assignment of a channel (a frequency) to every TRX (Eisenblätter 2001). The optimization problem arises because the usable radio spectrum is generally very scarce and, consequently, frequencies have to be reused by many TRXs in the network.

However, the multiple use of the same frequency may cause interferences that may reduce the quality of service down to unsatisfactory levels. Indeed, significant interference may occur if the same or adjacent-channels are used in neighboring, overlapping cells. The point here is that computing this level of interference is a difficult task which depends not only on the channels, but also on the radio signals and the properties of the environment. The more accurate the measure of the interference in a given GSM network, the higher the quality of the frequency plan that can be computed for this network. Several ways of quantifying this interference exist, ranging from theoretical methods to extensive measurements (Kuurne 1965). They all result in a so-called *interference matrix*, denoted by M . Each element $M(i, j)$ of M indicates the degradation of the network quality if cells i and j operate on the same frequency. This is called *co-channel interference*. Apart from co-channel interference there may exist a so-called *adjacent-channel interference*, which occurs when two TRXs operate on adjacent channels (i.e., one TRX operates on channel f and the other on channel $f+1$ or $f-1$). An accurate interference matrix is therefore an essential requirement for frequency planning because the ultimate goal of any frequency assignment algorithm will be to minimize the sum of the interferences.

In real-life situations, additional complicating factors, such as separation constraints among cells, or advanced

interference reduction techniques, such as frequency hopping or dynamic power control, etc., may be considered. The interested reader is referred to Eisenblätter (2001) for a more detailed description of frequency planning in actual GSM networks.

3.3 Mathematical formulation

Let $T = \{t_1, t_2, \dots, t_n\}$ be a set of n transceivers, and let $F_i = \{f_{i1}, \dots, f_{ik}\} \subset \mathbb{N}$ be the set of valid frequencies that can be assigned to a transceiver $t_i \in T, i = 1, \dots, n$. Note that k —the cardinality of F_i —is not necessarily the same for all the transceivers. Furthermore, let $S = \{s_1, s_2, \dots, s_m\}$ be a set of given sectors (or cells) of cardinality m . Each transceiver $t_i \in T$ is installed exactly in one of the m sectors. Henceforth, we denote the sector in which a transceiver t_i is installed by $s(t_i) \in S$. Finally, given a matrix $M = \{(\mu_{ij}, \sigma_{ij})\}_{m \times m}$, called the *interference matrix*. The two elements μ_{ij} and σ_{ij} of a matrix entry $M(i, j) = (\mu_{ij}, \sigma_{ij})$ are numerical values greater than or equal to zero. In fact, μ_{ij} represents the mean and σ_{ij} the standard deviation of a Gaussian probability distribution describing the carrier-to-interference ratio (C/I) (Walke 2002) when sectors i and j operate on a same frequency. The higher the mean value, the lower the interference and thus the better is the communication quality. Note that the interference matrix is defined at sector (cell) level, because the transceivers installed in each sector all serve the same area.

A solution to the problem is obtained by assigning to each transceiver $t_i \in T$ one of the frequencies from F_i . A solution (or frequency plan) is henceforth denoted by $p \in F_1 \times F_2 \times \dots \times F_n$, where $p(t_i) \in F_i$ is the frequency assigned to transceiver t_i . The objective is to find a solution p that minimizes the following cost function:

$$C(p) = \sum_{t \in T} \sum_{u \in T, u \neq t} C_{\text{sig}}(p, t, u). \tag{1}$$

To define the function $C_{\text{sig}}(p, t, u)$, let s_t and s_u be the sectors in which the transceivers t and u are installed, that is, $s_t = s(t)$ and $s_u = s(u)$, respectively. Moreover, let μ_{s_t, s_u} and σ_{s_t, s_u} be the two elements of the corresponding matrix entry $M(s_t, s_u)$ of the interference matrix with respect to sectors s_t and s_u . Then, $C_{\text{sig}}(p, t, u) =$

$$\begin{cases} K & \text{if } s_t = s_u, |p(t) - p(u)| < 2 \\ C_{\text{co}}(\mu_{s_t, s_u}, \sigma_{s_t, s_u}) & \text{if } s_t \neq s_u, \mu_{s_t, s_u} > 0, |p(t) - p(u)| = 0 \\ C_{\text{adj}}(\mu_{s_t, s_u}, \sigma_{s_t, s_u}) & \text{if } s_t \neq s_u, \mu_{s_t, s_u} > 0, |p(t) - p(u)| = 1 \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

K is a very large constant ($K \gg 0$) defined by the network designer so as to make it undesirable allocating the same or

adjacent frequencies to transceivers serving the same area. Furthermore, function $C_{\text{co}}(\mu, \sigma)$ is defined as follows:

$$C_{\text{co}}(\mu, \sigma) = 100 \left(1.0 - Q \left(\frac{c_{\text{SH}} - \mu}{\sigma} \right) \right) \tag{3}$$

where

$$Q(z) = \int_z^\infty \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} dx \tag{4}$$

is the tail integral of a Gaussian probability distribution function with zero mean and unit variance, and c_{SH} is a minimum quality signaling threshold. Function Q is widely used in digital communication systems because it characterizes the error probability performance of digital signals (Simon and Alouini 2005). This means that $Q(\frac{c_{\text{SH}} - \mu}{\sigma})$ is the probability of the C/I ratio being greater than c_{SH} and, therefore, $C_{\text{co}}(\mu_{s_t, s_u}, \sigma_{s_t, s_u})$ computes the probability of the C/I ratio in the serving area of sector s_t being below the quality threshold due to the interferences provoked by sector s_u . That is, if this probability is low, the C/I value in the sector s_t is not likely to be degraded by the interfering signal coming from sector s_u and thus the communication quality yielded is high. (Note that this is compliant as to defining a minimization problem.) On the contrary, a high probability—and consequently a high cost—causes the C/I mostly to be below the minimum threshold c_{SH} and thus incurring in low-quality communications.

As function Q has no closed form for the integral, it has to be evaluated numerically. For this purpose we use the complementary error function E :

$$Q(z) = \frac{1}{2} E \left(\frac{z}{\sqrt{2}} \right) \tag{5}$$

In Press et al. (1992), a numerical method is presented that allows the value of E to be computed with a fractional error smaller than 1.2×10^{-7} .

Analogously, function $C_{\text{adj}}(\mu, \sigma)$ is defined as

$$\begin{aligned} C_{\text{adj}}(\mu, \sigma) &= 100 \left(1.0 - Q \left(\frac{c_{\text{SH}} - c_{\text{ACR}} - \mu}{\sigma} \right) \right) \\ &= 100 \left(1.0 - \frac{1}{2} E \left(\frac{c_{\text{SH}} - c_{\text{ACR}} - \mu}{\sigma \sqrt{2}} \right) \right). \end{aligned} \tag{6}$$

The only difference between functions C_{co} and C_{adj} is the additional constant $c_{\text{ACR}} > 0$ (*adjacent channel rejection*) in the definition of function C_{adj} . This hardware-specific constant measures the receiver’s ability to receive the wanted signal in the presence of an unwanted signal at an adjacent channel. Note that the effect of constant c_{ACR} is that $C_{\text{adj}}(\mu, \sigma) < C_{\text{co}}(\mu, \sigma)$. This makes sense, since using adjacent frequencies (channels) does not provoke such a strong interference as using the same frequencies.

The mathematical model is aimed at incorporating interference information directly imported from real-world GSM frequency planning as currently conducted in the industry (and not generated in a computer by sampling random variables). Indeed, the computations carried out to obtain the cost values are motivated by real-world GSM networks since they are related to the computation of the Bit Error Rate (BER) performance of Gaussian Minimum Shift Keying (GMSK), the modulation scheme used for GSM (Simon and Alouini 2005).

4 Proposed metaheuristics

The aim of this section is to present the algorithms used in this work for solving the proposed FAP. Both population- and trajectory-based metaheuristics have been evaluated. All of them use the same local search algorithm, which will be described in this section as well. The Ant Colony Optimization algorithm (ACO) (Dorigo and Stützle 2004) will also be briefly described, to compare the four new metaheuristics with previous work in the literature (Luna et al. 2007a).

4.1 Local Search (LS)

The application of local search methods allows admissible solutions to be achieved in relatively short times. This is a typical requirement within commercial tools, the context in which the FAP resides. To perform a fair comparison of the implemented approaches, the local search strategy (see Algorithm 1) has been defined as a common element for all the proposed metaheuristics. The local search strategy has been specifically designed to deal with our version of FAP.

The operation of the designed local search is based on optimizing the assignment of frequencies to TRXs in a given sector, without changing the remaining network assignments. In Mannino et al. (2007), a local search method which also tries to substitute the frequencies assigned to a set of transceivers, leaving intact the remaining network is proposed. However, in such a case, the set of considered transceivers are not those inside a sector, but the ones satisfying a set of specific properties. In Luna et al. (2007a) and Bjorklund et al. (2005), the local search methods are simpler. In both cases, the new neighbors are generated by changing the assignment to only one TRX.

Candidate solutions (or plans) are encoded as arrays of integer values p , where $p(t)$ is the frequency assigned to TRX t . The neighbors of a candidate solution are obtained by replacing the frequencies in the TRXs of each sector.

The reassignment of frequencies within a sector is performed in the following way: first, the available frequencies for the sector are sorted by their involved cost. Then, two possibilities are considered, either assign the frequency with lowest associated cost to a TRX that is allowed to use that frequency, or assign its two adjacent frequencies to two different TRXs (if they are allowed to use these frequencies). For each of the newly generated partial solutions the same process is repeated until all TRXs in the sector have been assigned a frequency. The complete solution with lowest associated cost is considered as the new neighbor, while the other ones are discarded. Since the new plan is constituted by just replacing a small set of frequency assignments, a complete evaluation of the individual is not necessary. Thus, only interferences produced by the replaced TRXs need to be recalculated.

The order in which neighbors are analyzed is randomly determined (line 7 of Algorithm 1), but trying to avoid the generation of neighbors that do not improve the current solution. For such purpose a set called *currentSectors* containing the sectors that might improve the current solution is maintained. Initially, all sectors are introduced in *currentSector* (lines 2,4). For the generation of a new neighbor, a sector *sec* is randomly extracted from *currentSector* (line 7) and its frequencies reassigned (line 8). The local search moves to the first newly generated neighbor that improves the current solution (lines 9–10), adding all the sectors that interfere or are interfered by *sec* to the set of the next sectors (*nextSectors*) to consider (lines 11–12). When *currentSectors* set gets empty (line 6), sectors in *nextSectors* are transferred to the current set (line 4) and *nextSectors* set is cleared (line 5). The local search stops when none of the neighbors improves the current solution (line 3).

In the cases, where the network satisfies a set of properties, the neighbor generation process ensures the achievement of the optimal frequency assignment inside the analyzed sector, considering the remaining network fixed. Such properties are (1) all TRXs in a given sector are allowed to use the same frequency ranges, (2) it is possible to make assignments which do not use the same frequency or adjacent frequencies in any two TRXs serving the same area, and (3) the best assignment does not use the same frequency or adjacent frequencies in any two TRXs within the same sector. A sketch of the proof is presented here. Let $Cost(f)$ be the cost associated to the assignment of the frequency f to any of the TRXs in the considered sector. f_1 being the frequency with minimum associated cost, the best assignment must use f_1 , or must simultaneously use f_1-1 and f_1+1 . In fact, considering an assignment in which f_1+1 is used, but f_1-1 is not used, we can substitute

the assignment of f_{l+1} by f_l , thus obtaining an assignment with lower cost.

In the case of using f_{l-1} , but not f_{l+1} , the same property holds. In the cases, where f_{l-1} and f_{l+1} are not used, since f_l is the best possible assignment it must be used. Finally, the simultaneous assignment of both f_{l+1} and f_{l-1} , could lead to a better assignment than the ones using f_l and other frequency f_2 . For this reason, in order to ensure that the best assignment is achieved, individuals which use f_l , and simultaneously f_{l-1} and f_{l+1} should be analyzed. The way in which neighbors are generated ensure that both possibilities are explored, so the best assignment is achieved under such conditions.

Algorithm 1 Pseudocode for Local Search

```

1: Input: current solution  $S$ 
2:  $nextSectors \leftarrow \{1, \dots, numberOfSectors\}$ 
3: while ( $nextSectors \neq \emptyset$ ) do
4:    $currentSectors \leftarrow nextSectors$ 
5:    $nextSectors \leftarrow \emptyset$ 
6:   while ( $currentSectors \neq \emptyset$ ) do
7:      $sec \leftarrow$  extract a random sector from  $currentSectors$ 
8:      $neighbour \leftarrow$  reassign frequencies of  $S$  in sector  $sec$ 
9:     if ( $neighbour$  improves  $S$ ) then
10:       $S \leftarrow neighbour$ 
11:       $nextSectors +=$  sectors interfered by  $sec$ 
12:       $nextSectors +=$  sectors that interfere  $sec$ 
13:     end if
14:   end while
15: end while
16: return  $S$ 

```

4.2 Ant Colony Optimization (ACO)

Since we want to compare with previously published results on the problem using the same instances, we have included the algorithm which reported the best results for these instances so far, the ACO presented in Luna et al. (2007a). To summarize this approach, it works as any other ACO algorithm: at each iteration candidate solutions are constructed in a probabilistic way. The probabilistic solution construction is based on a so-called pheromone model (denoted by \mathcal{T}), which is a set of numerical values that encode the algorithms' search experience. After the construction phase, some of the generated solutions are used to update the pheromone values in a way that aims at biasing the future solution construction towards good solutions found during the search process. The particular approach implemented is known as \mathcal{MMAS} (Max–Min Ant System) algorithm in the so-called hyper-cube framework (HCF); see Blum and Dorigo (2004).

4.3 Steady-State Genetic Algorithm (SSGA)

This algorithm is a standard steady state GA (SSGA). The tentative solutions managed by SSGA are encoded as arrays of integer values, p , where $p(t_i) \in F_i$ is the frequency assigned to transceiver t_i . That is, the solutions manipulated are tentative frequency plans of the given FAP problem instance.

Algorithm 2 Pseudocode for SSGA

```

1: population  $\leftarrow \emptyset$ 
2: initialize(population)
3: while not time-limit do
4:   parents  $\leftarrow$  binaryTournament(population)
5:   offspring  $\leftarrow$  UX(parents,  $p_c$ )
6:   offspring  $\leftarrow$  randomMutation(offspring,  $p_m$ )
7:   offspring  $\leftarrow$  localSearch(offspring, localSearchSteps)
8:   population  $\leftarrow$  insert(population, offspring)
9: end while

```

An outline of the algorithm is shown in Algorithm 2. The algorithm starts by creating a population of random individuals, so that all the TRXs of each individual are randomly assigned with one of their valid frequencies. As to the genetic operators, SSGA uses binary tournament as selection scheme (line 4). This operator works by randomly choosing two individuals from the population and the one having the best (lowest) fitness (Eq. 1) is selected. The algorithm then applies uniform crossover (UX) in which every allele of the offspring (i.e., the frequency of each TRX) is chosen randomly from one of the two parents (line 5). The mutation operator used is the random mutation in which the frequencies of a set of randomly chosen TRXs of the solution are reassigned with a random valid frequency. Next, the offspring undergoes a local search phase (line 7) which finally replaces the worst individual in the population, if the newly generated one is better (lower FAP cost).

4.4 Scatter Search (SS)

Scatter Search (SS) (Martí et al. 2006; Laguna et al. 2002) works with a quite small set of solutions called *RefSet* (solutions are encoded by using the same representation as SSGA, that is, arrays of integer values). This set is composed of the most representative solutions from the population. The individuals in the initial population are randomly generated. *RefSet* is divided into quality solutions (the best frequency plans for the FAP problem) and diverse solutions (the most different ones). The number of individuals for each subset has been specially configured to solve the FAP problem.

Algorithm 3 Pseudocode for Scatter Search

```

1: initialize(population)
2: population ← localSearch(population)
3: RefSet ← generateFrom(population)
4: while not time-limit do
5:   SubSet ← subSetGenerator(RefSet)
6:   SubSet ← combinationMethod(SubSet)
7:   RefSet ← localSearch(SubSet)
8:   RefSet ← generateFrom(RefSet, population)
9: end while

```

A brief description of the algorithm can be seen in Algorithm 3. As we can observe, we have followed the typical scheme of SS. The algorithm starts with the random generation of the population through the assignment of a valid frequency to each single TRX in each solution (line 1). Then, an Improvement Method fixed to the FAP problem is applied to each population individual to try to improve it (line 2). This method is the local search that we have explained previously (Sect. 4.1). After generating *RefSet* (line 3) we use a Subset Generation Method (line 5) to create all possible subsets from the *RefSet*. The next step is to apply the Solution Combination Method (line 6) to the solutions in each subset. The solutions are combined in a pair-wise way. The pairs of solutions are chosen at random from the current *RefSet*. After that, we combine the solutions by using a uniform crossover at TRX level. Finally, the local search will be applied again (line 7) to try to improve a frequency planning obtained as a result of the combination method. Frequency planning is replaced in the *RefSet* so that the best solutions to the FAP problem are maintained. In our case, we use a subset of quality solutions with only one individual, and a subset of diverse solutions formed by rest of the individuals in the *RefSet*. Therefore, when all combinations have been evaluated (we have to generate a new *RefSet*), the best solution is saved in the *RefSet* and a new random population is generated to select the (*RefSetSize*-1) most diverse solutions from that one (line 8). The distance used to measure the diversity between two frequency plans is the total amount of different frequencies assigned to every sector in both plans. With this new *RefSet* the algorithm restarts a new iteration until the time limit for the experiment is expired (line 4).

4.5 (1+1) Evolutionary Algorithm (EA)

The proposed approach is a modified version of a (1+1)-EA which uses a mutation operator specifically designed to face the FAP problem.

The main modification lies in allowing the growth of the population to improve the behavior when dealing with

strong local optima. Individuals, which are encoded as in SSGA (see Sect. 4.3), are randomly generated as well.

Algorithm 4 Pseudocode for the (1+1) Evolutionary Algorithm

```

1: initialize(population)
2: population ← localSearch(population)
3: while not time-limit do
4:   offspring ← mutation(population)
5:   offspring ← localSearch(offspring)
6:   for  $i = 0$  to populationSize do
7:     if population( $i$ ) is blocked for softBloq generations then
8:       population( $i$ ) ← offspring( $i$ )
9:     else
10:      population( $i$ ) ← best(population( $i$ ), offspring( $i$ ))
11:     end if
12:   end for
13:   if population is blocked for hardBloq generations then
14:     if |population| < maxPopSize then
15:       increase population size attaching a new individual
16:     end if
17:   end if
18: end while

```

An outline of the approach is shown in Algorithm 4. The EA approach has been combined with the local search method to improve the algorithm convergence. After every creation of an individual, the local search is applied (lines 2, 5). A mutation operator is applied over each individual at each generation (line 4) to produce a new offspring. The (1,1) selection operator is deterministic and selects the individual with best fitness value (line 10). To improve the behavior of the approach when dealing with local optima, two improvements were considered. First, if after *SoftBloq* generations the fitness of the current individual has not been improved, the offspring is selected independent of its fitness value (lines 6-12). Moreover, if after *hardBloq* generations the fitness value of none of the individuals has been improved, an extra new individual is introduced in the population (lines 13-17). To avoid an excessive growth of the population, the maximum size of the population is limited to *maxPopSize* (line 14). During the following generations, each individual included in the population is evolved applying the aforementioned rules.

The variation of individuals in the population is based on a single mutation operator. First, the operator randomly selects one sector (*secMut*). Then, TRXs in *secMut* are randomly reassigned, whereas the ones in the interferer or victim sectors of *secMut* are randomly reassigned with a probability *pmut*. The previous process is repeated *mutSelected* times, but selecting as *secMut* a random sector among the previously interferer or victim sectors.

4.6 Local Search with Heuristic Restarts (LSHR)

This algorithm extends the local search algorithm described in Sect. 4.1, by adding periodically guided perturbations to avoid local minima. In short, it periodically restarts local search by means of a probability distribution F learned during the search process. F is a matrix with dimension $M \text{ sectors} \times N \text{ frequencies}$ and it is updated in a similar fashion as PBIL (Baluja 1994).

Algorithm 5 Pseudocode for Local Search with Heuristic Restarts

```

1: initialize(probability matrix  $F$ )
2:  $S^* \leftarrow S \leftarrow \text{generateFrom}(F)$ 
3: while not time-limit do
4:    $S \leftarrow \text{localSearch}(S)$ 
5:    $S^* \leftarrow \text{best}(S, S^*)$ 
6:    $F \leftarrow \text{update}(F, S^*)$ 
7:    $S \leftarrow \text{generateFrom}(F)$ 
8: end while

```

The algorithm is summarized in Algorithm 5. Matrix F is a probability distribution representing a memory of the search process so far. F_{ij} is the probability that frequency j is assigned to sector i . According to Algorithm 6, the probability distribution F is initialized in line 1 with uniform probabilities ($F_{ij} = 1/(M \times N)$). An initial solution S is generated from F by means of a roulette-wheel based procedure (line 2). This procedure is described later. The best-so-far solution S^* is also initialized. Then, a loop is entered where local search is applied on current solution S (line 4). S is considered for replacing S^* in line 5. Finally, F is updated with the improved solution S^* (line 6). To update F , F_{ij} is reinforced if the frequency j is assigned to sector i in solution S . More specifically, $F'_{ij} = F_{ij} + f_r * P_{ij}$, where $P_{ij} = 1$ if frequency j is assigned to sector i in the best solution, and $P_{ij} = 0$ otherwise. The loop is repeated until time is elapsed.

The GenerateFrom procedure in line 7 represents the heuristic restart step required so that the local search jumps out of local minima. LSHR returns the best plan S^* found during the search. This procedure is detailed next; to generate a solution, we use an auxiliary heuristics matrix, H , which is initialized as a copy of the probability matrix F . By means of roulette-wheel, a cell H_{ij} is chosen. This means that frequency j is assigned to sector i in the solution that is being generated. This partial solution involves the following modifications in matrix H ; to discard new assignments of frequency j to sector i , cell H_{ij} is set to 0. Adjacent cells H_{ij-1} and H_{ij+1} are set to 0 too, to avoid adjacent channel frequencies in the same sector. If the number of assigned frequencies in sector i equals its number of TRXs, the whole row H_i corresponding to that

sector is set to 0. Once matrix H has been updated, a new cell H_{ij} is chosen by roulette-wheel (meaning that frequency j is assigned to sector i), followed by the corresponding updating of H . This process is repeated until every sector is assigned as many frequencies as TRXs.

5 Experimental evaluation

This section describes the two problem instances used in this paper, and presents the empirical results of the four algorithms tested.

5.1 Problem instances

Here, we want to provide the reader with details on the two FAP instances which are being tackled. They correspond to two US cities: Seattle and Denver. Whereas the former has 970 TRXs and 15 different frequencies to be assigned, the latter has 2,612 TRXs and 18 frequencies. In addition to size and number of frequencies to be assigned, Denver also has a larger TRX density: every TRX has 107.84 neighbor TRXs on average, whereas this amount is only 59.65 for Seattle. This makes Denver the most difficult of the two instances.

The constants used in the mathematical formulation (Luna et al. 2007a) were set to $K = 100,000$, $c_{SH} = 6 \text{ dB}$, and $c_{ACR} = 18 \text{ dB}$, respectively. Figures 2 and 3 display the network topology, every triangle representing a sectorized antenna in which several TRXs operate. These GSM networks are currently operating so finding their optimal plannings is of great practical interest.

We also want to remark that the data source to build the interference matrix based on the C/I probability distribution uses thousands of mobile measurement reports (MMRs)

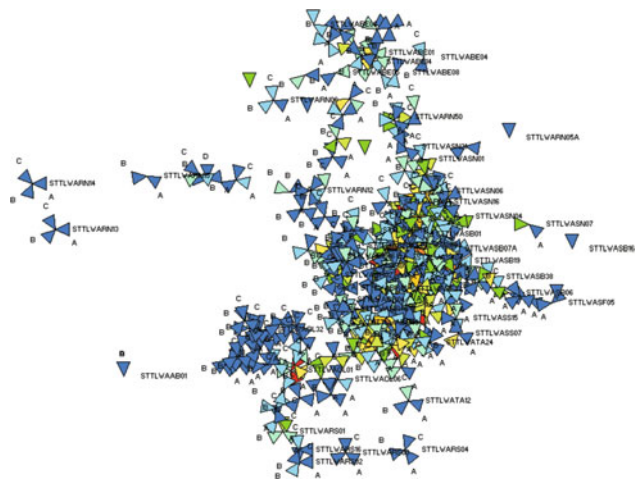


Fig. 2 Topology of the Seattle instance

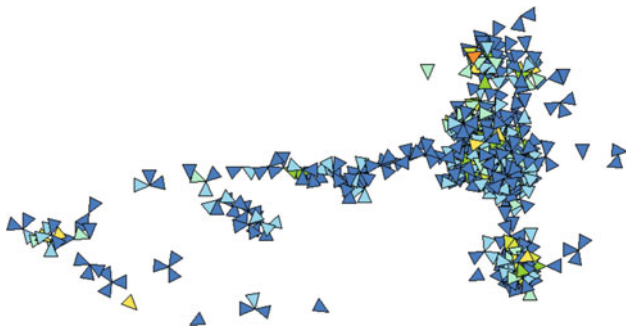


Fig. 3 Topology of the Denver instance

(Kuurne 1965) rather than propagation prediction models. MMRs are a more accurate data source, as they capture the cell location pattern in the network and do not rely on predictions. These properties make our GSM problem more realistic than standard available benchmarks (FAP Web: <http://fap.zib.de/>). Indeed, the most similar available instances are the COST 259 benchmark, but the basic traffic load is drawn at random according to an empirically observed distribution, and signals are predicted with several propagation models. The Philadelphia, CELAR and GRAPH instances (FAP Web: <http://fap.zib.de/>) are even simpler.

5.2 Parametrization

This section summarizes the parameter settings for each of the algorithms. Preliminary experiments were carried out to find the best set of parameter values for each algorithm. These parameter settings are:

- **SSGA:** Population size = 10, uniform crossover with $p_c = 1.0$, random mutation with $p_m = 0.2$, selection with binary tournament, replacement = worst individual
- **SS:** Population size = 40, RefSet size = 9, solution combination method = uniform crossover
- **EA:** softBloq = 50, hardBloq = 300, maxPopSize = 5, pmut = 0.9, mutSelected = 7
- **LSHR:** Learning rate $f_r = 0.001$
- **ACO:** 5 ants, the heuristic information used avoids using similar frequencies (Option 2 in Luna et al. 2007a), r_{det} is started with $r_{\text{det}} = 0.7$ and then gradually decreased until $r_{\text{det}} = 0.3$ in each restart phase.

5.3 Empirical results

Experiments for all the algorithms have been carried out under exactly the same conditions: an Intel Xeon 3 GHz processor and 2 GB RAM has been used for this purpose. On the software side, we have used gcc version 3.2.3 on a Red Hat Linux 2.4.21-4. Previous research (Luna et al.

2007a) reported results on applying ACO on the Denver instance, but a different machine was used. Although ACO is not one of the metaheuristics developed for this paper, we have carried out experiments with this metaheuristic to compare with these previous results (Luna et al. 2007a).

Since we are dealing with stochastic algorithms, we have carried out 30 independent runs for each metaheuristic. To provide the results with statistical confidence and detect differences between the algorithms within short- and long-time ranges, we have considered four different time limits (120, 600, 1,800, and 3,600 s) and performed statistical comparisons on them. These limits have been chosen by bearing in mind the practical interest of telecommunication companies, for which the time requirements are usually very hard. Average cost values, \bar{x} , and standard deviations, σ_n , of the 30 executions for every algorithm are summarized in Tables 2 and 3, for the Seattle and Denver instances, respectively (values in italics show the best (lowest) value).

Let us start by analyzing the resulting frequency plans computed by the considered algorithms. To properly put in context the quality of these plans, Table 4 displays the average, \bar{x} , and the standard deviation, σ_n , of the cost of randomly generated solutions, on the one hand, and randomly generated solutions that undergo the local search method used by the metaheuristic algorithms, on the other hand. The goal is to highlight two different facts. First, the high accuracy of this local search algorithm, which is able to reduce the cost of random frequency plans several orders of magnitude (specially in the Denver instance, the larger one). And, second, we want to show that the metaheuristics can profit from the hybridization with this local search by reaching frequency plans provoking weaker interference than those obtained by the standalone local search. Indeed, if we consider the results included in Tables 2 and 3, it can be easily seen that the FAP costs of the plans are lower for all the algorithms, time limits, and instances undertaken.

Taking into account these two tables (Tables 2, 3), it is remarkable that all the proposed algorithms are able to keep improving the solution quality in the two problem instances. These numerical values also show that the behavior of the different algorithms depends on the real-world instance addressed, even though the problem class is the same. However, several interesting conclusions can be drawn. For the Seattle instance, all trajectory-based metaheuristics give lower interfering plans than population-based ones for all time ranges, (1+1)-EA being the best performer. For the Denver instance, the situation seems to be reversed, although it is not so clear cut. In this case, the best plans are given by a population-based metaheuristic (SS), while one of the trajectory-based algorithms (LSHR) performs comparatively worse than the rest. As mentioned in Sect. 5.1, Denver is more difficult than Seattle, and

Table 2 Empirical results of the metaheuristics for four different time limits on the Seattle instance

Seattle	120 s	600 s	1,800 s	3,600 s
ACO	1,889.64 ± 118.97	1,578.65 ± 148.06	1,380.33 ± 128.33	1,330.49 ± 116.79
SSGA	1,894.67 ± 79.83	1,757.60 ± 87.49	1,676.12 ± 63.02	1,628.05 ± 51.17
SS	2,115.56 ± 191.43	1,606.85 ± 183.21	1,348.74 ± 145.95	1,238.68 ± 141.55
(1+1) EA	1,417.28 ± 141.94	1,142.65 ± 108.47	989.30 ± 73.07	917.43 ± 51.92
LSHR	1,677.73 ± 208.45	1,341.40 ± 123.48	1,194.13 ± 105.85	1,102.13 ± 115.49

The mean and the standard deviation are presented

Table 3 Empirical results of the metaheuristics for four different time limits on the Denver instance

Denver	120 s	600 s	1,800 s	3,600 s
ACO	93,439.46 ± 1,341.54	92,325.42 ± 1,111.56	90,649.93 ± 740.02	89,875.66 ± 699.49
SSGA	89,540.41 ± 1,008.14	87,850.79 ± 583.45	86,908.94 ± 386.37	86,870.40 ± 320.02
SS	89,401.36 ± 1,091.63	87,233.42 ± 874.73	86,122.66 ± 666.58	85,525.17 ± 494.54
(1+1) EA	89,798.47 ± 1,305.70	87,859.68 ± 1,038.68	86,835.99 ± 1,016.04	86,363.98 ± 790.68
LSHR	92,946.57 ± 1,519.37	89,680.33 ± 902.81	88,646.53 ± 526.92	88,367.90 ± 406.18

The mean and the standard deviation are presented

Table 4 Average and standard deviation of 30 random solutions and 30 executions of local search for the Denver and Seattle networks

	Random $\bar{x} \pm \sigma_n$	Local search $\bar{x} \pm \sigma_n$
Seattle	55,204.11 ± 1,815.99	3,692.66 ± 405.19
Denver	115,577,436.48 ± 3,902,668.53	105,155.60 ± 2,077.20

therefore it requires a more thorough search to find good-quality solutions. So, it is understandable that SS diversity maintenance techniques perform very well on the Denver instance. For the same reason, a pure trajectory-based technique as LSHR underperforms on Denver.

We also want to note that previous research has already used the Denver instance in Luna et al. (2007a), where an ACO algorithm was used. The same ACO metaheuristic has been tested on the new computer environment and the new instance. In the Denver instance, all metaheuristics perform better than ACO, and in the Seattle one, all of them, but one (SSGA) also give lower interference plans, so we have been able to improve upon previous results in the literature.

Overall, a trajectory-based algorithm ((1+1)-EA) and a population-based one (SS) point out a good behavior in both instances. Not only they offer the lowest interference plans for one of the instances, but also obtain reasonable results in the other. To graphically display all our claims, Figs. 4 and 5 show the evolution of the best (lowest) costs of the frequency plans every 120 s, from 2 to 60 min.

These conclusions will now be qualified by statistical tests. The following statistical analysis has been performed for every time limit (Demšar 2006; Sheskin 2003). First a Kolmogorov–Smirnov test is performed to check whether the values of the results follow a normal (Gaussian) distribution or not. Since this test is positive in all the samples, we have used means and standard deviations in the result tables. Then, a Levene test checks for the homogeneity of the variances. If samples have equal variance (positive Levene test), an ANOVA test is done; otherwise we perform a Welch test. We consider here a confidence level of 95% (i.e., significance level of 5% or p -value under 0.05), which means that the differences are unlikely to have occurred by chance with a probability of 95%. To further analyze the results statistically, we have then included a post hoc testing phase in Tables 5 and 6 which allows for a multiple comparison of samples. We have used the `multcompare` function provided by Matlab®. It chooses the most appropriate type of critical value to be used in the multiple comparison, which ranges from the more conservative *HSD* or *Tukey–Kramer* method to less conservative *Scheffe's S* procedure (Hochberg and Tamhane 1987). The same confidence level has been kept for this testing phase ($\alpha = 0.05$).

The way to interpret the pairwise comparisons of Tables 5 (Seattle) and 6 (Denver) is as follows. For every pair of algorithms, the tables display those time ranges, where differences are not significant. The “–” symbol means that the differences all are statistically significant. First, it must be noticed that for both problem instances,

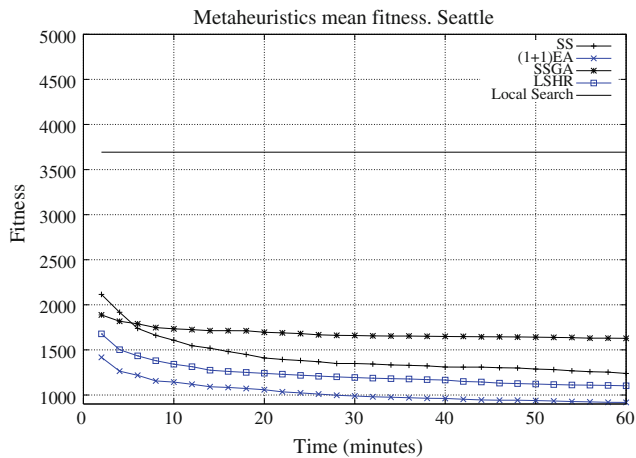


Fig. 4 Average interference of the four metaheuristics every 120 s in the Seattle network

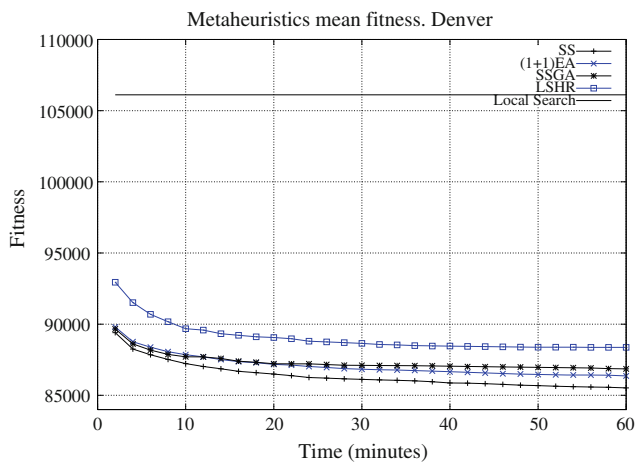


Fig. 5 Average interference of the four metaheuristics every 120 s in the Denver network

differences for the 3,600 s time range are always significant. Second, it was noticed before that in the Seattle instance, trajectory-based algorithms performed better than population-based ones. Table 5 shows that in fact, all differences between population- and trajectory-based metaheuristics are significant. Moreover, differences among the population-based algorithms (ACO, SS, and SSGA) are in some cases not significant. With respect to the trajectory-based ones, (1+1)-EA outperforms significantly LSHR for all time ranges, so this is clearly the best algorithm in this domain (Fig. 4). In the Denver instance (see Table 6), SS is the best algorithm for the 3,600-s range, while it behaves similarly to other algorithms for short time spans (120 and 600 s). It seems that the diversity maintenance techniques of SS allow this metaheuristic to avoid stagnation for the longer time periods (1,800 and 3,600 s, as it can be seen in Fig. 5). It can also be seen that a population- (SSGA) and a

Table 5 Post hoc tests of the results for the Seattle instance

SSGA	120			
SS	600, 1,800	–		
(1+1)-EA	–	–	–	
LSHR	–	–	–	–
	ACO	SSGA	SS	(1+1)-EA

Time limits for which the pairwise comparison is not significant

Table 6 Post hoc tests of the results for the Denver instance

SSGA	–			
SS	–	120, 600		
(1+1)-EA	–	120, 600, 1,800	120, 600	
LSHR	120	–	–	–
	ACO	SSGA	SS	(1+1)-EA

Time limits for which the pairwise comparison is not significant

trajectory-based ((1+1)-EA) have a similar behavior for most time ranges.

Summarizing all the above, we can conclude that (1+1)-EA and SS are particularly accurate algorithms for this domain. (1+1)-EA is a very effective algorithm for both problem instances. It is significantly the best for Seattle and the runner up in Denver, and in this latter case, differences with the best performer (SS) are not statistically significant for time ranges smaller than 600 s. SS is the statistically significant winner for time ranges over 600 s in the Denver domain, and the third performer in Seattle, not far away for the two best algorithms. The remaining algorithms behave well in one of the domains (LSHR in Seattle and SSGA in Denver), but badly in the other (LSHR in Denver and SSGA in Seattle).

To further analyze the results obtained, we have considered the resulting frequency plans reached by the two best overall metaheuristics for the Seattle and Denver instances, (1+1)-EA and SS, respectively. In these plans, we have counted not only the total number of collisions, i.e., the number of co-channel and adjacent-channel interferences, but also the amount of collisions whose C_{sig} (as given by Eq. 2) is equal to or greater than T , being $T = 10, 20, \dots, 100$. (Note that 100 is the higher value that Eq. 2 can take if no collisions appear between TRXs within the same sector.) These results for Seattle and Denver are averaged over the 30 frequency plans and for the four time limits considered. They all are, respectively, included in Tables 7 and 8. The goal here is to show how the two best algorithms work and to give some hints on the complexity of the instances tested.

In the case of the Seattle instance with (1+1)-EA, it can be seen that the improved frequency plans reached by the algorithm are based on decreasing the total number of collisions (row “Collisions” in Table 7) all over the

network. Even though this is an expected behavior, the reduction in the number of collisions is rather small (from 6,379.53 to 6,368.87) if we consider the total amount of planning cost achieved (from 1,417.28 in 120 s to 917.43 in 3,600 s). The interesting fact is that the resulting improvements of the (1+1)-EA are based on reducing the collisions that most contribute to the planning cost, e.g., the frequency plans reached by (1+1)-EA in 120 s have, on average, 10.93 collisions with $C_{\text{sig}} \geq 30$, but only 4.70 in 3,600 s. No collisions with $C_{\text{sig}} \geq 60$ appear in any of the computed plans.

If we consider now the collision information of the SS algorithm when solving the Denver instance (Table 8), the scenario is actually different. Indeed, as it can be seen in the first (number of collisions) and last rows (cost of the frequency plans), the higher the quality of the plan (lower interference), the greater is the number of collisions. Although this may be thought as contradictory, the distribution of the signaling costs helps to better understand this behavior. This distribution points out that the frequency plans reached in the longer runs have more co-channel and adjacent-channel interferences, but all of them provoking the smaller signaling cost, i.e., $C_{\text{sig}} \leq 10$. The interferences with higher signaling costs tend to be reduced. The rest of algorithms follow the same pattern, but the results are not displayed here in order to minimize the number of tables. As a main conclusion, we want to remark here the importance of having real-world interference information as we do in this work, since better frequency plans do not necessarily mean a smaller number of collisions on real GSM networks.

Table 7 Collisions and distribution of the signaling cost C_{sig} of (1+1)-EA for the Seattle instance

(1+1)-EA Collisions	Time limit (s)			
	120	600	1,800	3,600
	6,379.53	6,375.40	6,372.57	6,368.87
$C_{\text{sig}} \geq 10$	30.27	22.27	17.00	14.43
$C_{\text{sig}} \geq 20$	19.03	13.10	9.77	8.83
$C_{\text{sig}} \geq 30$	10.93	7.47	5.17	4.70
$C_{\text{sig}} \geq 40$	5.40	3.57	3.07	2.63
$C_{\text{sig}} \geq 50$	0.27	0.07	0.07	0.07
$C_{\text{sig}} \geq 60$	0.00	0.00	0.00	0.00
$C_{\text{sig}} \geq 70$	0.00	0.00	0.00	0.00
$C_{\text{sig}} \geq 80$	0.00	0.00	0.00	0.00
$C_{\text{sig}} \geq 90$	0.00	0.00	0.00	0.00
$C_{\text{sig}} \geq 100$	0.00	0.00	0.00	0.00
Total cost (from Table 2)	1,417.28	1,142.65	989.30	917.43

Table 8 Collisions and distribution of the signaling cost C_{sig} of SS for the Denver instance

SS Collisions	Time limit (s)			
	120	600	1,800	3,600
	38,779.23	38,787.33	38,860.77	38,911.93
$C_{\text{sig}} \geq 10$	2,447.93	2,374.40	2,344.30	2,325.63
$C_{\text{sig}} \geq 20$	1,995.47	1,938.63	1,906.37	1,890.60
$C_{\text{sig}} \geq 30$	1,090.00	1,074.80	1,056.97	1,057.50
$C_{\text{sig}} \geq 40$	848.90	846.13	836.07	836.37
$C_{\text{sig}} \geq 50$	46.90	44.10	41.77	41.30
$C_{\text{sig}} \geq 60$	21.73	18.17	17.20	16.17
$C_{\text{sig}} \geq 70$	19.37	15.57	14.80	13.90
$C_{\text{sig}} \geq 80$	14.20	11.13	10.80	10.47
$C_{\text{sig}} \geq 90$	14.20	11.13	10.80	10.47
$C_{\text{sig}} \geq 100$	14.20	11.13	10.80	10.47
Total cost (from Table 3)	89,401.36	87,233.42	86,122.66	85,525.17

6 Conclusions and future work

In this paper, a real-world FAP for GSM mobile networks has been addressed. A mathematical formalism that incorporates actual interference information is used (Luna et al. 2007a). Four research groups have collaborated in the design and adaptation of different metaheuristics, both trajectory- and population-based, for our version of FAP. The advice provided in previous research has been acknowledged (Talbi 2002; Luna et al. 2007), and all of the algorithms have been hybridized with an efficient local search method, specially adapted to our version of FAP. It has to be remarked that all the metaheuristics use the same local search method, to improve the fairness of the comparison. The local search was able to reduce the cost of random frequency plans several orders of magnitude. The hybrid metaheuristics based on this local search improved the frequency plans even more. All of the metaheuristics have been carefully evaluated and compared on two large real-world instances of FAP: Denver and Seattle.

The current work is an extension of Luna et al. (2008), where four metaheuristics were applied to the Denver FAP instance. In this paper, these metaheuristics have been improved and tested on two real-world instances. Two of the metaheuristics emerged as the best performers overall, for both real-world instances. These best performers are SS and (1+1)-EA, SS due to its diversity maintenance techniques that allow this metaheuristic to avoid stagnation for the longer time periods in large real-world instances (as Denver) and (1+1)-EA due to its ability to perform as a completely trajectory-based algorithm when no stagnation is detected, and increase the population size in order to

avoid strong local optima when necessary. Denver is the hardest instance, both in terms of size and in TRX density, and therefore it is very important to maintain more diversity. This explains that SS is the most adequate algorithm for this instance. In the case of Seattle, it was easier to avoid local optima, so maintaining a very diverse population is not so important, and (1+1)-EA was able to beat SS.

Another important conclusion of this work is that better frequency plans do not necessarily mean a smaller number of collisions on real GSM networks, because each collision has an associated signaling cost, and the collisions are more or less important depending on their respective signaling costs. On real GSM networks the most important objective is to maximize the quality of service, that is, to reduce the really significant interferences. We take into account this aspect, being an important distinction between this work and many other FAP researches that only deals with benchmarking-like problems.

So the most important contributions of the coordination of our four research groups are the use of a FAP formulation that uses real-world interference data; the design and adaptation of four different metaheuristics to this FAP, all of them using the same local search algorithm; the detailed statistical comparison of all our results in two real-world large problem instances; and the generation of a new best result, better than the ones published in the literature (Luna et al. 2007a, 2008).

The formulation of FAP as a multiobjective optimization problem will be investigated as future work.

Acknowledgments This work has been partially funded by the Spanish Ministry of Science and Innovation and FEDER under contract TIN2008-06491-C04-04 (the MSTAR project). J.M. Chaves-González is supported by the research grant PRE06003 from Junta de Extremadura (Spain). The work of Gara Miranda has been developed under grant FPU-AP2004-2290. Francisco Luna acknowledges support from the grant BES-2006-13075 funded by the Spanish government. The work of Carlos Segura was funded by grant FPU-AP2008-03213.

References

- Aardal KI, van Hoesel SPM, Koster AMCA, Mannino C, Sassano A (2007) Models and solution techniques for frequency assignment problems. *Ann Oper Res* 153(1):79–129
- Alabau M, Idoumghar L, Schott R (2001) New hybrid genetic algorithms for the frequency assignment problem. In: Proceedings of the 13th international conference on tools with artificial intelligence, pp 136–142
- Alabau M, Idoumghar L, Schott R (2002) New hybrid genetic algorithms for the frequency assignment problem. *IEEE Trans Broad* 48(1):27–34
- Amaldi E, Capone A, Malucelli F, Mannino C (2006) Optimization problems and models for planning cellular networks. In: *Handbook of optimization in telecommunications*. Springer, pp 917–939
- Avenali A, Mannino C, Sassano A (2002) Minimizing the span of d-walks to compute optimum frequency assignments. *Math Prog (A)* 91:357–374
- Baluja S (1994) 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, Pittsburgh, PA, USA
- Bjorklund P, Varbrand P, Yuan D (2005) Optimized planning of frequency hopping in cellular networks. *Comput Oper Res* 32(1):169–186
- Blum C, Roli A (2003) Metaheuristics in combinatorial optimization: overview and conceptual comparison. *ACM Comput Surv* 35(3):268–308
- Blum C, Dorigo M (2004) The hyper-cube framework for ant colony optimization. *IEEE Trans Syst Man Cybern Part B* 34(2):1161–1172
- Chaves-González JM, Vega-Rodríguez MA, Domínguez-González D, Gómez-Pulido JA, Sánchez-Pérez JM (2008a) SS vs PBIL to solve a real-world frequency assignment problem in GSM networks. In: *Applications of evolutionary Computing (Evo-COMNET'08)*. LNCS 4974, pp 21–30
- Chaves-González JM, Domínguez-González D, Vega-Rodríguez MA, Gómez-Pulido JA, Sánchez-Pérez JM (2008b) Parallelizing pbil for solving a real-world frequency assignment problem in gsm networks. In: *16th Euromicro conference on parallel, distributed and network-based processing (PDP 2008)*, pp 391–398
- Colombo G (2006) A genetic algorithm for frequency assignment with problem decomposition. *Int J Mobile Netw Des Innov* 1(2):102–112
- Demšar J (2006) Statistical comparison of classifiers over multiple data sets. *J Mach Learn Res* 7:1–30
- Dorigo M, Stützle T (2004) *Ant colony optimization*. MIT press, Cambridge
- Eisenblätter A (2001) Frequency assignment in GSM networks: models, heuristics, and lower bounds. PhD thesis, Technische Universität Berlin
- Eisenblätter A, Grötschel M, Koster AMCA (2002) Frequency planning and ramifications of coloring. *Discuss Math Graph Theory* 22(1):51–88
- Feo T, Resende M (1995) Greedy randomized adaptive search procedures. *J Glob Optim* 6:109–133
- Fischetti M, Lepsch C, Minerva G, Romanin-Jacur G, Toto E (2000) Frequency assignment in mobile radio systems using branch-and-cut techniques. *Eur J Oper Res* 123(2):241–255
- Furuskar A, Naslund J, Olofsson H (1999) EDGE—enhanced data rates for GSM and TDMA/136 evolution. *Ericsson Review* (1)
- Glover F (1998) A template for scatter search and path relinking. In: *AE '97: selected papers from the third European conference on artificial evolution*, London, UK, Springer, pp 3–54
- Glover FW, Kochenberger GA (2003) *Handbook of metaheuristics*. Kluwer
- Gomes FC, Pardalos P, Oliveira CS, Resende MGC (2001) Reactive GRASP with path relinking for channel assignment in mobile phone networks. In: *Proceedings of the 5th international workshop on discrete algorithms and methods for mobile computing and communications (DIALM'01)*, pp 60–67
- Granbohm H, Wiklund J (1999) GPRS—general packet radio service. *Ericsson Review* (1)
- Greff JY, Idoumghar L, Schott R (2004) Application of markov decision processes to the frequency assignment problem. *Appl Artif Intell* 18(8):761–773
- Hale WK (1980) Frequency assignment: theory and applications. *Proc IEEE* 68(12):1497–1514
- Hamiez HP, Hao JK (2002) Scatter search for graph coloring. In: *Artificial evolution*. LNCS 2310, pp 195–213

- Hochberg Y, Tamhane AC (1987) Multiple comparison procedures. Wiley
- Idoumghar L, Schott R (2006) A new hybrid GA-MDP algorithm for the frequency assignment problem. In: Proceedings of the 18th IEEE international conference on tools with artificial intelligence (ICTAI'06), pp 18–25
- Jaumard B, Marcotte O, Meyer C (1999) Mathematical models and exact methods for channel assignment in cellular networks. In: Telecommunications network planning. Kluwer, pp 239–256
- Kuurne AMJ (2002) On GSM mobile measurement based interference matrix generation. In: IEEE 55th vehicular technology conference, VTC Spring 2002, pp 1965–1969
- Kim SS, Smith AE, Lee JH (2007) A memetic algorithm for channel assignment in wireless FDMA systems. *Comput Oper Res* 34:1842–1856
- Laguna M, Hossell KP, Marti R (2002) Scatter search: methodology and implementation in C. Kluwer Academic Publishers, Norwell
- Lau TL, Tsang EPK (2001) Guided genetic algorithm and its application to radio link frequency assignment problems. *Constraints* 6(4):373–398
- Leese R, Hurley S (2002) Methods and algorithms for radio channel assignment. In: Oxford lecture series in mathematics and its applications. Oxford University Press
- Liu X, Pardalos PM, Rajasekaran S, Resende MGC (2000) Dimacs series on discrete mathematics and theoretical computer science 52:195–201
- Luna F, Alba E, Nebro AJ (2005) Parallel heterogeneous metaheuristics. In: Alba E (ed) Parallel metaheuristics. Wiley, pp 395–422
- Luna F, Blum C, Alba E, Nebro AJ (2007a) ACO vs EAs for solving a real-world frequency assignment problem in GSM networks. In: Genetic and evolutionary computation conference (GECCO 2007), pp 94–101
- Luna F, Alba E, Nebro AJ, Pedraza S (2007b) Evolutionary algorithms for real-world instances of the automatic frequency planning problem in GSM networks. In: Seventh European conference on evolutionary computation in combinatorial optimization (EVO-COP 2007). Volume 4446 of LNCS, pp 108–120
- Luna F, Estébanez C, León C, Chaves-González JM, Alba E, Aler R, Segura C, Vega-Rodríguez MA, Nebro AJ, Valls JM, Miranda G, Gómez-Pulido JA (2008) Metaheuristics for solving a real-world frequency assignment problem in gsm networks. In: Conference on genetic and evolutionary computation (GECCO 2008), pp 1579–1586
- Mabed H, Caminada A, Hao JK, Renaud D (2002) A dynamic traffic model for frequency assignment. In: Parallel problem solving from nature (PPSN VII). LNCS 2439, pp 779–788
- Mannino C, Sassano A (2003) An enumerative algorithm for the frequency assignment problem. *Discret Appl Math* 129:155–169
- Mannino C, Oriolo G, Ricci F, Chandran S (2007) The stable set problem and the thinness of a graph. *Oper Res Lett* 35(1):1–9
- Martí R (2003) Handbook of metaheuristics. pp 355–368
- Martí R, Laguna M, Glover F (2006) Principles of scatter search. *Eur J Oper Res* 169(2):359–372
- Matsui S, Watanabe I, Tokoro KI (2003) An efficient hybrid genetic algorithm for a fixed channel assignment problem with limited bandwidth. In: Genetic and evolutionary computation conference (GECCO 2003). LNCS 2724, pp 2240–2251
- Matsui S, Watanabe I, Tokoro KI (2005) Application of the parameter-free genetic algorithm to the fixed channel assignment problem. *Syst Comput Jpn* 36(4):71–81
- Metzger BH (1970) Spectrum management technique. In: 38th National ORSA Meeting
- Mishra AR (2004) Radio network planning and optimisation. In: Fundamentals of cellular network planning and optimisation: 2G/2.5G/3G... Evolution to 4G. Wiley, pp 21–54
- Mouly M, Paulet MB (1992) The GSM system for mobile communications. Mouly et Paulet, Palaiseau
- Press WH, Flannery BP, Teukolsky SA, Vetterling WT (1992) Numerical recipes in C: the art of scientific computing. Cambridge University Press
- Raidl G (2006) A unified view on hybrid metaheuristics. In: Hybrid metaheuristics (HM 2006). LNCS 4030, pp 1–12
- Rapeli J (1995) UMTS: targets, system concept, and standardization in a global framework. *IEEE Pers Commun* 2(1):30–37
- Raymond A, Lyandres V, Santiago RC (2003) On a guided genetic algorithm for frequency assignment in non-homogeneous cellular networks. In: 2003 IEEE international symposium on electromagnetic compatibility (EMC '03), pp 660–663
- Salcedo-Sanz S, Bousoño-Calzón C (2005) A portable and scalable algorithm for a class of constrained combinatorial optimization problems. *Comput Oper Res* 32:2671–2687
- Simon MK, Alouini MS (2005) Digital communication over fading channels: a unified approach to performance analysis. Wiley
- Sheskin DJ (2003) Handbook of parametric and nonparametric statistical procedures. CRC Press
- Talbi E (2002) A taxonomy of hybrid metaheuristics. *J Heuristics* 8(5):541–564
- Tiourine SR, Hurkens CAJ, Lenstra JK (2000) Local search algorithms for the radio link frequency assignment problem. *Telecommun Syst* 13(2–4):293–314
- Valenzuela C (2001) A study of permutation operators for minimum span frequency assignment using an order based representation. *J Heuristics* 7:5–21
- Voudouris C, Tsang E (1999) Guided local search. *Eur J Oper Res* 113(2):449–499
- Walke BH (2002) Mobile radio networks: Networking, protocols and traffic performance. Wiley
- Weinberg B, Bachelet V, Talbi EG (2001) A co-evolutionist metaheuristic for the assignment of the frequencies in cellular networks. In: EvoWorkshop 2001. LNCS 2037, pp 140–149