ACO vs EAs for Solving a Real-World Frequency Assignment Problem in GSM Networks[*]

Francisco Luna
Computer Science Dept.
Universidad de Malaga
Malaga, Spain
flv@lcc.uma.es

Christian Blum
ALBCOM research group
Universitat Polit. de Catalunya
Barcelona, Spain
cblum@lsi.upc.edu

Enrique Alba,
Antonio J. Nebro
Computer Science Dept.
Universidad de Malaga
Malaga, Spain
{eat,antonio}@lcc.uma.es

ABSTRACT

Frequency planning is a very important task for current GSM operators. In this work we present a new mathematical formulation of the problem in which the frequency plans are evaluated by using accurate interference information coming from a real GSM network. We have developed an ant colony optimization (ACO) algorithm to tackle this problem. After accurately tuning this algorithm, it has been compared against a (1,10) Evolutionary Algorithm (EA). The results show that the ACO clearly outperforms the EA when using different time limits as stopping condition for a rather extensive comparison.

Categories and Subject Descriptors

I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search—*Heuristic methods*; G.1.6 [**Numerical Analysis**]: Optimization—*Global optimization*

General Terms

Algorithms, Experimentation, Performance

Keywords

Ant colony optimization, frequency assignment, evolutionary algorithms

1. INTRODUCTION

Frequency planning in the *global system for mobile* (GSM) communications [22] is an important task for today's GSM operators not only at the stage of the initial deployment of the system, but also in subsequent expansions/modifications of the network which deal with unpredicted interference reports, and/or handle anticipated scenarios (e.g., expected increase of traffic demand in some areas). Indeed, by mid 2006 GSM services were used by more than 1.8 billion subscribers[1] across 210 countries, representing approximately 77% of the world's cellular market. It is widely accepted that the *Universal Mobile Telecommunication System* (UMTS) [24], the third generation mobile telecommunication system, will coexist with the enhanced releases of the GSM standard (GPRS [14] and EDGE [13]) at least in the first phases. GSM is then expected to play an important role as a dominating technology for many years. Therefore, frequency planning in these networks will be an important task, at present as well as in the future.

The success of this multi-service cellular radio system lies in efficiently using the scarcely available radio spectrum. The available frequency band is slotted into channels (or frequencies) which have to be allocated to the elementary transceivers (TRXs) installed in the base stations of the network. This problem is known as Automatic Frequency Planning (AFP), Frequency Assignment Problem (FAP), or Channel Assignment Problem (CAP). Several different problem types are subsumed under these general terms and many mathematical models have been proposed since the late sixties [1]. This work is focussed on concepts and models which are relevant for current GSM frequency planning [10]. For these reasons, we separate ourselves from existing results, since our problem is far different from those reported in the literature with similar names (which are benchmarking-like problems). We have developed a new formulation for the problem so as to take full advantage of realistic and accurate interference information from a real-world GSM network. As a generalization of the graph coloring problem, the FAP is NP-hard [15] and therefore using exact algorithms to solve real-sized instances of the problem is not practical. Metaheuristics [5] become the choice here.

We have tackled the FAP with a standard ant colony optimization (ACO) [8] approach since this technique has provided good results when applied to other formulations of the FAP (see, for example, [19]). The algorithm was evaluated on a real-world GSM network with 2612 transceivers operating in a rather large U.S. city. After tuning, we compared our algorithm against an evolutionary algorithm (EA) [18]. The ACO has shown to be very effective on the given problem instance, clearly outperforming the EA approach.
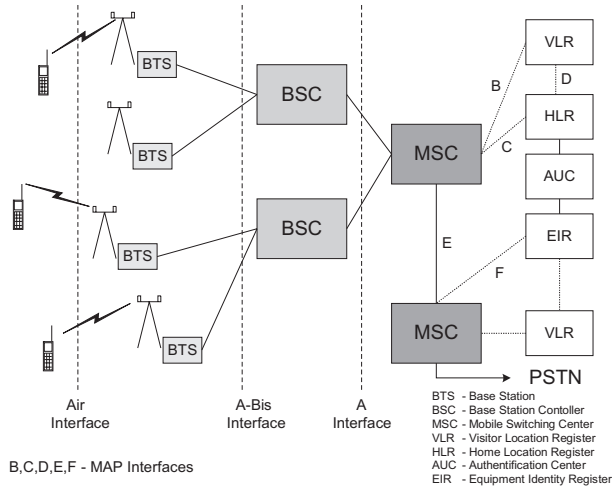
[1]http://www.wirelessintelligence.com/

Figure 1: Outline of the GSM network architecture.

The organization of the paper is as follows. In Section 2 we present the background and the mathematical formulation of our frequency assignment problem. In Section 3 we outline the ACO approach and the EA algorithm. Finally, in Section 4 we present the experimental evaluation of the algorithms, and then offer some conclusions and an outline of future work in Section 5.

2. FREQUENCY ASSIGNMENT IN GSM NETWORKS

In the following we first provide the reader with a brief description of the GSM architecture, whereby we introduce the basic terminology of the problem. Next, we give details on the frequency planning task in GSM networks. Finally, a precise mathematical formulation of the FAP model addressed in this paper is presented.

2.1 The GSM System

An outline of the GSM network architecture is shown in Figure 1. As it can be seen, GSM networks consist of many different components. The most relevant ones to frequency planning are the Base Transceiver Stations (BTSs) 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 (*Time Division Multiple Access*) mode. The main role of a TRX is to provide conversion between the digital traffic data on the network side and radio communication between the mobile terminal and the GSM network. The site at which a BTS is installed is usually organized in sectors: one to three sectors are typical. The area in which each sector operates defines a cell.

The solid lines connecting components in Figure 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.

2.2 Frequency Planning in GSM Networks

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 [20]. 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 in the assignment of a channel (a frequency) to every TRX [10]. 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 a same frequency may cause interferences that may reduce the quality of service (QoS) 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 [17]. 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 it 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 [10] for a more detailed description of frequency planning in actual GSM networks.

2.3 Mathematical Formulation

Let $T = \{t_1, t_2, \ldots, t_n\}$ be a set of n *transceivers*, and let $F_i = \{f_{i1}, \ldots, f_{ik}\} \subset \mathbb{N}$ be the set of valid *frequencies* that can be assigned to a transceiver $t_i \in T$, $i = 1, \ldots, n$. Note that k —the cardinality of F_i— is not necessarily the same for all the transceivers. Furthermore, let $S = \{s_1, s_2, \ldots, s_m\}$ be a set of given *sectors* (or cells) of cardinality m. Each transceiver $t_i \in T$ is installed in exactly 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 is 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 or equal than 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) [27] when sectors i and j operate on a same frequency. The higher the

mean value, the lower the interference and thus the better 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 $p \in F_1 \times F_2 \times \cdots \times F_n$, 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:

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

In order to define the function $C_{\mathrm{sig}}(p, t, u)$, 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 $\mu_{s_t s_u}$ and $\sigma_{s_t s_u}$ 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, $C_{\mathrm{sig}}(p, t, u) =$

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

$K >> 0$ is a very large constant defined by the network designer so as to make it undesirable allocating the same or adjacent frequencies to transceivers serving the same area (i.e., installed in the same sector). Furthermore, function $C_{\mathrm{co}}(\mu, \sigma)$ is defined as follows:

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

where

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

is the tail integral of a Gaussian probability distribution function with zero mean and unit variance, and c_{SH} is a minimum quality signalling threshold. Function Q is widely used in digital communication systems because it characterizes the error probability performance of digital signals [26]. This means that $Q \left(\frac{c_{\mathrm{SH}} - \mu}{\sigma} \right)$ is the probability of the C/I ratio being greater than c_{SH} and, therefore, $C_{\mathrm{co}}(\mu_{s_t s_u}, \sigma_{s_t s_u})$ 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:

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

In [23], a numerical method is presented that allows the value of E to be computed with a fractional error smaller than $1.2 \cdot 10^{-7}$. Analogously, function $C_{\mathrm{adj}}(\mu, \sigma)$ is defined as

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

The only difference between functions C_{co} and C_{adj} is the additional constant $c_{\mathrm{ACR}} > 0$ (*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 $C_{\mathrm{adj}}(\mu, \sigma) < C_{\mathrm{co}}(\mu, \sigma)$. This makes sense, since using adjacent frequencies (channels) does not provoke such a strong interference as using the same frequencies.

2.4 Relation to FAPs from the Literature

Many mathematical formulations have been proposed in the literature for modelling FAP-like problems [2] and, more specifically, FAPs arising in GSM networks [6, 10]. Some of the existing models even include specific GSM issues such as frequency hopping [3, 21]. However, none of these formulations considers all the concepts and advanced technologies used in real GSM frequency planning [11]. Researchers facing real-world planning problems had to rely on simulation, e.g. [7, 16, 18, 25].

The new feature of our model (as outlined before) is to be found in the definition of the interference matrix information. Our model ultimately aims at measuring the overall signalling performance of the GSM network. Up to now, interference matrices have only included one single value for each pair of interfering TRXs (typically normalized to be between 0 and 1), which characterizes the interference between TRXs. In contrast, the interference matrix used in our work includes the interference between cells, by giving the entire probability distribution of the C/I ratio. This definition, which is directly imported from real world GSM frequency planning as currently conducted in the industry (and not generated in a computer by sampling random variables), allows not only the computation of high performance frequency plans, but also the prediction of QoS. Indeed, both the definition of the interference matrix and the subsequent computations to obtain the cost values are motivated by real-world GSM networks since they are related to the computation of the BER (Bit Error Rate) performance of Gaussian Minimum Shift Keying (GMSK), the modulation scheme used for GSM [26]. To the best of our knowledge, this is the first formulation of the FAP using such accurate interference and performance measurements coming from GSM networks. We also want to note that this is the first approach at using such interference information and subsequent extensions may be considered in future works.

3. ALGORITHMS

This section is devoted to presenting the two algorithms used in this work for solving the proposed FAP: the ACO algorithm (Section 3.1) and the EA approach (Section 3.2).

3.1 The ACO Approach

In general, our approach works as any other ACO algorithm: at each iteration candidate solutions are constructed in a probabilistic way. The probabilistic solution construc-

tion 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 that we implemented is known as $\mathcal{MMAS}$ algorithm in the so-called hyper-cube framework (HCF); see [4]. A high level description of our algorithm is given in Algorithm 1. The data structures used by this algorithm, in addition to counters and to the pheromone model $\mathcal{T}$, are: (1) the *iteration-best* solution p_{ib}, that is, the best solution (frequency plan) generated in the current iteration by n_a ants; (2) the *best-so-far* solution p_{bs}, i.e., the best solution generated since the start of the algorithm; (3) the *restart-best* solution p_{rb}, that is, the best solution generated since the last restart of the algorithm; (4) the *convergence factor* cf, $0 \leq cf \leq 1$, which is a measure of how far the algorithm is from convergence; and (5) the Boolean variable bs_update, which becomes true when the algorithm reaches convergence.

Roughly, the algorithm works as follows. First, all the variables are initialized. In particular, the pheromone values are set to their initial value 0.5 by the procedure InitializePheromoneValues($\mathcal{T}$). Each algorithm iteration consists of the following steps. First, n_a solutions are generated by applying function GenerateFrequencyPlan($\mathcal{T}$). The iteration-best solution p_{ib} might be improved by the optional application of a local search method. Second, the values of the variables p_{ib}, p_{rb} and p_{bs} are updated in function Update(p_{ib},p_{rb},p_{bs}). Third, pheromone trail values are updated via the ApplyPheromoneUpdate(cf, bs_update, $\mathcal{T}$, p_{ib}, p_{rb}, p_{bs}) procedure. Fourth, a new value for the convergence factor cf is computed. Depending on this value, as well as on the value of the Boolean variable bs_update, a decision on whether to restart the algorithm or not is made. If the algorithm is restarted, then the procedure ResetPheromoneValues($\mathcal{T}$) is applied and all the pheromone values are reset to their initial value (set to 0.5). The algorithm is iterated until some opportunely defined termination conditions are satisfied. Once terminated, the algorithm returns the best-so-far solution s_{bs}. In the following we describe the main procedures of Algorithm 1 in more detail.

GenerateFrequencyPlan($\mathcal{T}$): ACO algorithms utilize a so-called pheromone model $\mathcal{T}$, which is a set of numerical values, for the probabilistic construction of solutions. Here, $\mathcal{T}$ consists of a pheromone value $\tau_{ij} > 0$ for each combination of a transceiver t_i and a valid frequency $f_{ij} \in F_i$. τ_{ij} represents the desirability of assigning frequency f_{ij} to transceiver t_i. A solution is constructed by choosing for each transceiver t_i a frequency from F_i in the order $i = 1, \ldots, n$. This is done as follows. First, at each construction step the following probabilities are generated:

$$\mathbf{p}(f_{ij}) = \frac{\tau_{ij} \cdot \eta_{ij}}{\sum_{f_{il} \in F_i} \tau_{il} \cdot \eta_{il}} \ , \ \forall \ f_{ij} \in F_i \qquad (7)$$

The η values in this equation are called the *heuristic information*. They are generally defined by using the weights of

---

Algorithm 1 Pseudo-code of the ACO approach

1: **input:** a problem instance
2: $p_{bs} \leftarrow$ NULL, $p_{rb} \leftarrow$ NULL, $cf \leftarrow 0$, $bs\_update \leftarrow$ FALSE
3: InitializePheromoneValues($\mathcal{T}$)
4: **while** termination conditions not satisfied **do**
5:    **for** $j \leftarrow 1$ to n_a **do**
6:       $p_j \leftarrow$ GenerateFrequencyPlan($\mathcal{T}$)
7:    **end for**
8:    $p_{ib} \leftarrow$ argmin($C(p_1), ..., C(p_{n_a})$)
9:    $p_{ib} \leftarrow$ LocalSearch(p_{ib},d)   /* optional */
10:   Update(p_{ib},p_{rb},p_{bs})
11:   ApplyPheromoneUpdate(cf,bs_update,$\mathcal{T}$,p_{ib},p_{rb},p_{bs})
12:   $cf \leftarrow$ ComputeConvergenceFactor($\mathcal{T}$)
13:   **if** $cf > 0.99$ **then**
14:     **if** $bs\_update =$ TRUE **then**
15:       ResetPheromoneValues($\mathcal{T}$)
16:       $p_{rb} \leftarrow$ NULL
17:       $bs\_update \leftarrow$ FALSE
18:     **else**
19:       $bs\_update \leftarrow$ TRUE
20:     **end if**
21:   **end if**
22: **end while**
23: **output:** p_{bs}

---

a greedy heuristic for the tackled problem. Here we consider 3 different options:

- **Option 1:** No heuristic information, that is, all η_{ij} values are set to 1.

- **Option 2:** Equation 2 shows that the objective function punishes the assignment of *too similar* frequencies to transceivers installed in the same sector. Therefore, we try to avoid using too similar frequencies as follows. Let T_i be the set of transceivers that are installed in sector $s(t_i)$, and let $\hat{T}_i \subset T_i$ be the set of transceivers that have already assigned a frequency. Then,

$$\eta_{ij} \leftarrow \left(\left(100 \cdot \sum_{t_l \in \hat{T}_i} \delta(p(t_l), f_{ij}) \right) + 1 \right)^{-1} , \quad (8)$$

where $\delta(x, y) = 1$ if $|x - y| \leq 1$, and $\delta(x, y) = 0$ otherwise. In this setting, we multiply the sum of the *interferences* with 100 in order to increase their importance with respect to the pheromone information. The value of 100 was decided by tuning by hand.

- **Option 3:** As the objective function is additive, we can compute the exact influence of assigning a frequency $f_{ij} \in F_i$ to a transceiver t_i with respect to the current partial solution. Let $\hat{T} = \{t_1, \ldots, t_{i-1}\}$ be the set of transceivers to which a frequency is already assigned. The increase of the objective function value caused by the setting $p(t_i) = f_{ij}$ can be computed as follows:

$$\Delta(p, p(t_i) = f_{ij}) = \sum_{t \in \hat{T}} \left(C_{\text{sig}}(p, t, t_i) + C_{\text{sig}}(p, t_i, t) \right) \qquad (9)$$

Accordingly, we define

$$\eta_{ij} = ((100 \cdot \Delta(p, p(t_i) = f_{ij})) + 1)^{-1} \ . \qquad (10)$$

Algorithm 2 Pseudo-code for the local search

1: **input:** a solution p, a number of steps d
2: *improved* $\leftarrow$ TRUE
3: $k \leftarrow 1$
4: **while** $k \leq d$ **and** *improved* = TRUE **do**
5:     *improved* $\leftarrow$ FALSE
6:     **for** $i \leftarrow 1$ to n **do**
7:         Replace frequency $p(t_i)$ with the frequency from F_i that most reduces the objective function value
8:         **if** the objective function value was reduced **then** *improved* = TRUE
9:     **end for**
10:    $k \leftarrow k+1$
11: **end while**
12: **output:** a possibly improved solution p

In order to choose one of the possible frequencies for transceiver t_i, we proceed as follows. First, a random number $r \in [0,1]$ is generated. In case $r < d_{\det}$, frequency $f_{ij} \in F_i$ is chosen such that $\mathbf{p}(f_{ij}) \geq \mathbf{p}(f_{il})$, $\forall f_{il} \in F_i$. Otherwise, a frequency for t_i is chosen probabilistically by roulette-wheel-selection with respect to the probabilities defined in Equation 7. Note that $d_{\det}$ is an important parameter of the algorithm. It is used to determine the percentage of deterministic versus probabilistic construction steps.

LocalSearch(p_{ib},d): Optionally we apply a simple local search method to the iteration-best solution p_{ib}. Parameter d indicates for how many steps this algorithm should be maximally executed (see Algorithm 2). This parameter is crucial due to the considerable computation time consumption of each local search step.

ApplyPheromoneUpdate(cf,bs_update,$\mathcal{T}$,p_{ib},p_{rb},p_{bs}): In general, three solutions are used for updating the pheromone values. These are the iteration-best solution p_{ib}, the restart-best solution p_{rb}, and the best-so-far solution p_{bs}. The influence of each solution on the pheromone update depends on the state of convergence of the algorithm as measured by the convergence factor cf. At the beginning of each restart phase (i.e., when bs_update = FALSE and cf is close to zero), the iteration-best solution p_{ib} has maximal influence. Then, as the algorithm progresses (i.e., when cf increases), the influence of the restart-best solution p_{rb} increases. Shortly before convergence (i.e., $cf \leq 0.99$, but close to 0.99), the restart-best solution p_{rb} has maximal influence, and once convergence is detected (i.e., $cf > 0.99$), the control variable bs_update is set to TRUE which has the effect that the best-so-far solution p_{bs} has maximal influence. This is done in order to intensify the search around the best-so-far solution and can have the effect to move the search to a different zone of the search space, with the consequent decrease of the convergence factor.

Each pheromone value $\tau_{ij} \in \mathcal{T}$ is updated as follows:

$$\tau_{ij} \leftarrow \tau_{ij} + \rho \cdot (\mu_{ij} - \tau_{ij}) \ , \tag{11}$$

where

$$\mu_{ij} \leftarrow \kappa_{ib} \cdot \delta(p_{ib}, f_{ij}) + \kappa_{rb} \cdot \delta(p_{rb}, f_{ij}) + \kappa_{bs} \cdot \delta(p_{bs}, f_{ij}) \ , \tag{12}$$

where κ_{ib} is the weight (that is, the influence) of solution

Table 1: Setting of κ_{ib}, κ_{rb}, κ_{bs}, and ρ depending on the convergence factor cf and the Boolean control variable bs_update.

	bs_update = FALSE				bs_update = TRUE
	$cf < 0.4$	$cf \in [0.4, 0.6)$	$cf \in [0.6, 0.8)$	$cf \geq 0.8$	
κ_{ib}	1	2/3	1/3	0	0
κ_{rb}	0	1/3	2/3	1	0
κ_{bs}	0	0	0	0	1
ρ	0.2	0.2	0.2	0.15	0.15

p_{ib}, κ_{rb} is the weight of solution p_{rb}, κ_{bs} is the weight of solution p_{bs}, and $\kappa_{ib} + \kappa_{rb} + \kappa_{bs} = 1$. Moreover, $\delta(p, f_{ij}) = 1$ if $p(t_i) = f_{ij}$, and $\delta(p, f_{ij}) = 0$ otherwise. After the pheromone update rule (Equation 11) is applied, pheromone values that exceed $\tau_{\max} = 0.999$ are set back to $\tau_{\max}$ (similarly for $\tau_{\min} = 0.001$).

Equation 12 allows to choose how to schedule the relative influence of the three solutions used for updating the pheromone values. For our application we used the update schedule as shown in Table 1.

ComputeConvergenceFactor($\mathcal{T}$): The convergence factor cf, which is a function of the current pheromone values, is computed as follows:

$$cf \leftarrow 2 \left(\left(\frac{\sum\limits_{\tau_{ij} \in \mathcal{T}} \max\{\tau_{\max} - \tau_{ij}, \tau_{ij} - \tau_{\min}\}}{|\mathcal{T}| \cdot (\tau_{\max} - \tau_{\min})} \right) - 0.5 \right)$$

In this way, $cf = 0$ when the algorithm is initialized (or reset), that is, when all pheromone values are set to 0.5. On the other side, when the algorithm has converged, then $cf = 1$. In all other cases, cf has a value in $(0, 1)$. This completes the description of the ACO algorithm.

3.2 A simple EA approach

For comparison purposes we also implemented the simple (μ, λ) EA approach presented in [18]. The choice of this algorithm is motivated by the fact that it was specifically designed to face the FAP problem in real GSM networks. This optimization technique firstly generates μ initial solutions. These initial solutions undergo the application of local search (see Algorithm 2) which allows "good" solutions to be reached in relatively short times. This is a typical requirement within commercial applications, the context in which the algorithm was developed. Next, the EA perturbs and evaluates these μ individuals at each iteration, from which λ new ones are obtained. Then, the best μ solutions taken from the newly generated λ individuals are moved to the next iteration.

The goal of this section is to describe the modifications performed in the (μ, λ) EA because in [18] the EA used a commercial application not only for evaluating the tentative frequency plans on the real GSM scenario, but also for guiding the search by means of information obtained from the network. This accurate information is referred to as the "hardest to deal with" measure in [18] and it is used in both the initialization method and the perturbation operator to rank the TRXs according to interference provoked, capacity overload, etc. Since we do not use any simulation tool in this

work, we use instead the following information. Given a frequency plan p, the component cost function CC is defined as follows:

$$CC(p, t) = \sum_{u \in T, u \neq t} C_{\mathrm{sig}}(p, t, u) \qquad (13)$$

that is, $CC(p, t)$ is the value with which TRX t contributes to the total cost of the frequency plan p.

The precise configuration of the (μ, λ) EA used here is as follows. The values for μ and λ are 1 and 10, respectively. In the perturbation operator, only tournament selection is used for selecting the TRXs which undergo the reassignment of their frequency. These settings reported the best results in [18]. Two additional modifications have been considered. On the one hand, the perturbation involves reallocating the frequency assignment to three sites of the network (only one single site was the original strategy) so three different tournaments have to be executed; and, on the other hand, a final step of local search has been used to further refine the best solution computed by the (1,10) EA.

4. EXPERIMENTAL EVALUATION

We implemented our algorithms in ANSI C++ using GCC 3.2.2 for compiling the software. Our experimental results were obtained on a PC with an AMD64X2 4400 processor and 4 Gb of memory, running Debian Linux.

4.1 Problem Instance

Here, we want to provide the reader with details on the FAP instance which is being tackled. The GSM network used has 711 sectors with 2,612 TRXs to be assigned a frequency; the constants in Equations 2, 3, and 6 were set to $K = 100,000$, $c_{\mathrm{SH}} = 6\ dB$, and $c_{\mathrm{ACR}} = 18\ dB$, respectively. Each TRX has 18 available channels (from 134 to 151). Figure 2 displays the network topology, every triangle representing a sectorized antenna in which operate several TRXs.

This GSM network is currently operating in a U.S. 400 km^2 city with more than 500,000 people, so its solution is of great practical interest. The data source to build the interference matrix based on the C/I probability distribution uses thousands of Mobile Measurement Reports (MMRs) [17] rather than propagation prediction models. MMRs are a more accurate data source, as they capture the call location pattern in the network and do not rely on predictions. These properties make our GSM problem more realistic than standard available benchmarks [12]. 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 [12] are even simpler. Moreover, the largest instance in [12] scales only up to 1886 cells. Other realistic-sized GSM instances exist in the literature, e.g. [6], but neither they are publicly available nor they use accurate interference information as our mathematical model requires.

4.2 Tuning of the ACO algorithm

In the following we present the results of tuning experiments that were performed in order to fix the parameters of the ACO algorithm. The first set of experiments
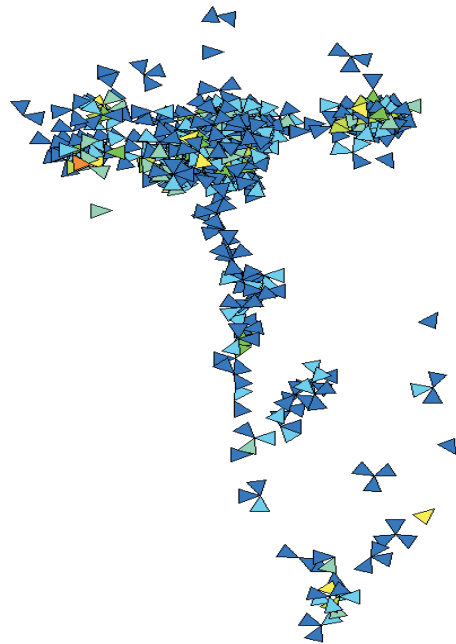

Figure 2: Topology of the GSM instance used.

concerns the three options for the heuristic information (see Section 3.1). We applied the ACO algorithm with the following fixed settings: $n_a = 5$, that is, 5 ants per iteration,[2] and $r_{\mathrm{det}} = 0.5$. The local search algorithm was applied with $d = 3$, that is, maximally 3 steps per application. We applied the ACO algorithm with all three options for the heuristic information (with and without local search) to the given problem instance. For all the runs we used a time limit of 600 seconds. The results—averaged over 10 independent runs—are shown in Table 2. For each setting (for example, Option 1 without local search) we provide the best value found in 10 runs (column **best**), the average over 10 runs (column **average**), and the standard deviation (column **std.**). The results allow us to draw the following conclusions. When no local search is used, the heuristic information plays a crucial role. In fact, the algorithm with Option 2, respectively Option 3, is around 280 times, respectively 430 times, better than the algorithm version using Option 1 (that is, no heuristic information). When comparing between using Options 2 and 3, we note a clear advantage of the algorithm version using Option 3. Once local search is used, the three algorithm versions do not differ that much. In fact, the best algorithm version is now the one using Option 2. The reason is based in the computation time needed for constructing a solution. While the computations required by Option 2 do not cause a significant increase in the computation time needed to construct a solution, the use of Option 3 is quite expensive. More in detail, the construction of a solution with Option 3 is around 12 times more expensive than the construction of a solution with Options 1 or 2. Based on these results we decided for Option 2 for all further experiments.

A second set of experiments concerned the setting of

---

[2]Note that this setting is not crucial. Hence, we just chose a reasonable value without performing tuning experiments.

Table 2: Numerical results concerning the heuristic information: Options 1, 2, and 3

	No local search			Local search		
	best	**average**	**std.**	**best**	**average**	**std.**
Option 1	67460090.28	69754591.73	1592508.81	90515.57	92021.80	983.03
Option 2	240193.31	242167.29	990.92	89703.44	91064.36	887.85
Option 3	154835.98	158307.52	2140.39	91496.06	93494.52	1511.98

the parameter r_{det}—the so-called *determinism rate*—which determines the percentage of deterministic versus probabilistic solution construction steps. We applied 10 versions of the ACO algorithm ($r_{det} \in \{0.0, 0.1, \ldots, 0.9\}$) with the fixed settings $n_a = 5$, Option 2, and the use of local search ($d = 3$) each 10 times for 600 seconds to the given problem instance. The results indicate a clear advantage of the algorithm versions that use medium determinism rates. With the intuition that at the beginning of a restart phase we need a higher determinism rate in order to reach good solutions quickly, each restart phase is started with $r_{det} = 0.7$. The determinism rate is then gradually decreased until $r_{det} = 0.3$ in each restart phase.

In a last set of tuning experiments we tested different settings for the number of allowed local search steps, that is, different settings for parameter d. We tested the 4 different settings $d \in \{1, 2, 3, 4\}$. The outcome (which is not shown for space reasons) was such that we used $d = 3$ for the final set of experiments.

4.3 Experimental results

In this section we present the results of ACO versus the results of the simple EA that was outlined in Section 3.2. Additionally, we tested a version of our ACO approach in which the pheromone update procedure is not used (line 11 of Algorithm 1). This is often done in order to prove the usefulness of the learning component of ACO. The resulting ACO version is henceforth denoted by ACO*. We applied all three algorithms 30 times to our problem instance. This was done for 3 different time limits (120, 600, and 1800 seconds) in order to detect possible differences in the behaviour of the algorithms when different execution times are considered. The results are shown in numerical form in Table 3 and in graphical form in Figure 3. Since we are dealing with stochastic algorithms and we want to provide the results with statistical confidence, the following analysis has been performed. Firstly, a Kolmogorov-Smirnov test is performed in order to check whether the values of the results follow a normal (gaussian) distribution or not. If so, an ANOVA test is done, otherwise we perform a Kruskal-Wallis test. We always consider in this work a confidence level of 95% (i.e., significance level of 5% or p-value under 0.05) in the statistical tests, which means that the differences are unlikely to have occurred by chance with a probability of 95%. Successful tests are marked with "+" symbols in the last column of Table 3; conversely, "−" means that no statistical confidence was found (p-value > 0.05).

We can observe that the ACO approach clearly outperforms the simple EA for what concerns all three time limits and with statistical confidence (see "+" symbols in the last column). Moreover, the ACO approach clearly outperforms
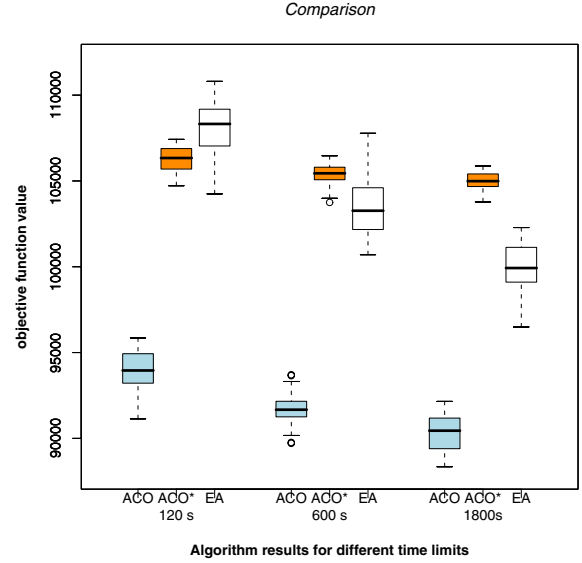

Figure 3: This graphic presents the results shown in Table 3 in the form of box-plots.

the ACO* version, which means that the learning component of ACO is useful in this case. One main conclusion can be drawn from these results: the collaborative search plays a very important role here. Indeed, in ACO* no pheromone update is performed and therefore the constructive phase does not profit from the search experience, that is, previous ant paths, whereas in the (1,10) EA no recombination is required, so no genetic material is exchanged among individuals. In fact, it has been shown in the literature that well known crossover operators for EAs such as single point crossover do not perform well on this problem [9] and this is why $\mu = 1$ was chosen in [18] for the (μ, λ) EA.

5. CONCLUSIONS AND FUTURE WORK

We presented an ACO algorithm for solving the frequency assignment problem in real GSM networks. A new mathematical formulation of the problem has been addressed so as to take full advantage of accurate interference information coming from a real-world GSM network. After carefully tuning the ACO, it has been compared against an (1,10) EA. The results have shown that the ACO algorithm clearly outperforms the EA approach using different time limits as stopping conditions. Evaluating the algorithms by using additional real-world instances and extending the mathematical model to deal with more realistic issues in GSM frequency planning (e.g., separation constraints, frequency hopping, etc.) are matter of future work.

Table 3: Numerical results of ACO, ACO*, and EA for 3 different time limits

time limit	ACO			ACO*			EA			
	best	average	std.	best	average	std.	best	average	std.	
120 s	91140.04	93978.21	1165.93	104719.72	106330.66	742.47	104247.72	108071.98	1723.40	+
600 s	89703.44	91726.41	1002.96	103752.12	105322.90	682.78	100701.28	103535.95	1939.74	+
1800 s	88345.94	90382.56	935.31	103781.86	104967.31	509.08	96490.58	99862.35	1553.10	+

6. REFERENCES

[1] K. I. Aardal, S. P. M. van Hoesen, A. M. C. A. Koster, C. Mannino, and A. Sassano. Models and solution techniques for frequency assignment problems. *4OR*, 1(4):261 – 317, 2003.

[2] K. I. Aardal, S. P. M. van Hoesen, A. M. C. A. Koster, C. Mannino, and A. Sassano. Models and solution techniques for frequency assignment problems. *Annals of Operations Research*, To appear, 2007.

[3] P. Björklund, P. Värbrand, and D. Yuan. Optimized planning of frequency hopping in cellular networks. *Computers and Operations Research*, 32(1):169 – 186, 2005.

[4] C. Blum and M. Dorigo. The hyper-cube framework for ant colony optimization. *IEEE Transactions on Systems, Man, and Cybernetics – Part B*, 34(2):1161–1172, 2004.

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

[6] R. Borndörfer, A. Eisenblätter, M. Grötschel, and A. Martin. Frequency assignment in cellular phone networks. *Annals of Operations Research*, 76:73 – 93, 1998.

[7] A. De Pasquale, N. P. Magnani, and P. Zanini. Optimizing frequency planning in the GSM system. In *IEEE 1998 Int. Conf. on Universal Personal Communications*, pages 293 – 297, 1998.

[8] M. Dorigo and T. Stützle. *Ant Colony Optimization*. MIT press, Cambridge, MA, 2004.

[9] R. Dorne and J.-K. Hao. An evolutionary approach for frequency assignment in cellular radio networks. In *Proc. of the IEEE Int. Conf. on Evolutionary Computation*, pages 539 – 544, 1995.

[10] A. Eisenblätter. *Frequency Assignment in GSM Networks: Models, Heuristics, and Lower Bounds*. PhD thesis, Technische Universität Berlin, 2001.

[11] A. Eisenblätter, M. Grötschel, and A. M. C. A. Koster. Frequency planning and ramifications of coloring. *Discussiones Mathematicae Graph Theory*, 22(1):51 – 88, 2002.

[12] FAP Web. http://fap.zib.de/.

[13] A. Furuskar, J. Naslund, and H. Olofsson. EDGE – enhanced data rates for GSM and TDMA/136 evolution. *Ericsson Review*, (1), 1999.

[14] H. Granbohm and J. Wiklund. GPRS – general packet radio service. *Ericsson Review*, (1), 1999.

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

[16] N. Jaldén. Autonomous frequency planning for GSM networks. Master's thesis, Royal Institute of Technology, Stockholm, 2004.

[17] A. M. J. Kuurne. On GSM mobile measurement based interference matrix generation. In *IEEE 55th Vehicular Technology Conference, VTC Spring 2002*, pages 1965 – 1969, 2002.

[18] F. Luna, E. Alba, A. Nebro, and S. Pedraza. Evolutionary algorithms for real-world instances of the automatic frequency planning problem in GSM networks. In *7th European Conf. on Evolutionary Computation in Combinatorial Optimisation, EVOCOP 2007*, 2007 (to appear).

[19] V. Maniezzo and A. Carbonaro. An ANTS heuristic for the frequency assignment problem. *Future Generation Computer Systems*, 16(9):927–935, 2000.

[20] A. R. Mishra. *Fundamentals of Cellular Network Planning and Optimisation: 2G/2.5G/3G... Evolution to 4G*, chapter Radio Network Planning and Optimisation, pages 21 – 54. Wiley, 2004.

[21] J. N. J. Moon, L. A. Hughes, and D. H. Smith. Assignment of frequency lists in frequency hopping networks. *IEEE Trans. on Vehicular Technology*, 54(3):1147 – 1159, 2005.

[22] M. Mouly and M. B. Paulet. *The GSM System for Mobile Communications*. Mouly et Paulet, Palaiseau, 1992.

[23] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge Unive. Press, 1992.

[24] J. Rapeli. UMTS: Targets, system concept, and standardization in a global framework. *IEEE Personal Communications*, 2(1):30 – 37, 1995.

[25] S. Ruíz, X. Colet, and J. J. Estevez. Frequency planning optimisation in real mobile networks. In *IEEE VTS 50th Vehicular Technology Conference*, pages 2082 – 2086, 1999.

[26] M. K. Simon and M-S. Alouini. *Digital Communication over Fading Channels: A Unified Approach to Performance Analysis*. Wiley, 2005.

[27] B. H. Walke. *Mobile Radio Networks: Networking, protocols and traffic performance*. Wiley, 2002.