

# COMPLEXITY OF DIFFERENT ILP MODELS OF THE FREQUENCY ASSIGNMENT PROBLEM

*Zoltán Ádám Mann\* and Anikó Szajkó†*

Budapest University of Technology and Economics  
Department of Computer Science and Information Theory

This chapter appeared in:

Zoltán Ádám Mann (editor): *Linear Programming – New Frontiers in Theory and Applications*, Nova Science Publishers, ISBN 978-1-61209-579-0, pages 305-326, 2012

## Abstract

The frequency assignment problem (FAP) arises in wireless communication networks, such as cellular phone communication systems, television broadcasting, WLANs, and military communication systems. In all these applications, the task is to assign frequencies to a set of transmitters, subject to interference constraints. The exact form of the constraints and the objective function vary according to the specific application.

Integer linear programming (ILP) is widely used to solve the different flavors of the FAP. For most FAP versions, there are more than one natural ILP formulations, e.g. using a large number of binary variables or a smaller number of integer variables.

A common experience with these solution techniques, as well as with NP-hard optimization problems in general, is a high variance in problem complexity. Some problem instances are tremendously hard to solve optimally. There are also examples of relatively big problem instances that are nevertheless quite easy to solve. In general, it is hard to predict how long it will take to solve a given problem instance.

This article presents a systematic study of how the complexity of the FAP depends on different parameters of the ILP model. We examine different types of constraints, different problem sizes and constraint densities, and varying sets of available frequencies. We conduct empirical measurements with an ILP solver to assess how problem complexity depends on these factors.

Based on the empirical data, it becomes possible to predict how time-consuming the solution of a given problem instance is, depending on the ILP model parameters. The ability to predict complexity is useful in several scenarios. First of all, it allows a sound judgement whether it is feasible to solve a given ILP formulation optimally. Moreover, it also supports sophisticated load-balancing when multiple FAPs are solved on parallel machines. Eventually, a better understanding of the origins of complexity may lead to enhanced optimization techniques.

---

\*E-mail address: zoltan.mann@gmail.com

†E-mail address: szajko.aniko@gmail.com

## 1. Introduction

Already in the 1890s, research started in developing wireless communication devices. In 1909, Marconi and Braun won the Nobel Prize for the invention of wireless telegraphy. Due to the rapid evolution of the technology, a hundred years later, one almost cannot imagine life without radio and TV transmission, satellite communication, Wi-Fi, GPS (Global Positioning System<sup>1</sup>), mobile phones, remote controls, radars, WLANs (Wireless Local Area Network) etc.

All these services operate in the radio waveband [3Hz, 300GHz]. Because of the extensive use of frequencies, the phenomenon of interference must be taken into account. This occurs if two communication channels are close to each other both geographically and in the radio band. Therefore, in each frequency planning task, frequencies should be assigned to communication channels, so that besides certain constraints the intensity of interference should be minimized. Research on frequency assignment problems (FAP) led to the insight in the 1960s, that finding an optimal solution is a quite hard mathematical problem.

The use of the radio spectrum is regulated by governments and worldwide by the International Telecommunication Union (ITU). Suppliers and operators of wireless networks are allowed to use only certain frequency bands, depending also on the geographical location. Usually, the available frequency band  $[f_{min}, f_{max}]$  is divided into channels with the same bandwidth ( $\Delta$ ). In this way, the channels (which are often called frequencies too) can be numbered from 1 to  $N$ , where  $N = \frac{f_{max} - f_{min}}{\Delta}$ . In some cases, an operator may not be allowed to use all the channels it paid for, for instance because of special regulations near country borders. We will denote the set of frequencies with  $F = \{1, \dots, N\}$ , and the set of frequencies available to a certain connection  $v$  with  $F(v)$ , where  $F(v) \subseteq F$ .

The magnitude of interference of signals depends not only on the location of the transmitters and receivers, but also on signal strength, the direction of transmission, geographical circumstances and weather conditions. In practice, more than two signals together can lead to interference too, but in most cases only the interference between pairs of communication channels is taken into consideration.

In this paper, we deal with Fixed Channel Assignment (FCA): time-invariant systems, in which the communication channels are constant over time. Extensions to time-varying models (Dynamic Channel Assignment, DCA) or mixed models (Hybrid Channel Assignment, HCA) are not considered.

For solving frequency assignment problems, several solution techniques are used, both exact algorithms and heuristics. A comprehensive survey can be found in [1] and [10]. One of the most popular solution techniques involves modeling the FAP by means of an Integer Linear Program (ILP) and using a general-purpose ILP solver to solve it. Unfortunately, all natural formulations of the FAP are NP-hard [3]. The ILP approach, just like any other known exact method for solving the FAP, takes exponentially long in the worst case. On the other hand, there are also many problem instances that are relatively easy to solve. This high variability in algorithm runtime poses a significant challenge on its practical application, because it is hard to predict if the algorithm will solve a given problem instance within a couple of seconds or will run for several days (or even longer). This phenomenon is common in the case of NP-hard problems [4, 5, 7, 9].

<sup>1</sup>The list of used abbreviations can be found at the end of the chapter.

To cope with this challenge, this chapter presents a comprehensive empirical study about the dependence of problem complexity on the problem formulation and the problem's parameters. In the first half of the chapter, we review the different FAP problem models and their possible ILP formulations. In the second half of the chapter, we present the empirical results.

## 2. Frequency assignment problems

In this section, we introduce the different flavors and models of FAPs.

### 2.1. Common application domains

#### 2.1.1. Mobile phone networks

In this application, communication takes place between a fixed antenna and a mobile phone. Each antenna covers a certain region, where it can serve the mobile devices. In TDMA (Time Division Multiple Access) systems, each frequency can be used to serve several mobile phones. In addition, with the help of TRXs (multiple transmitter/receiver unit) more frequencies can be assigned to the same antenna. In general, several antennas are installed on one physical unit (site). In GSM (Global System for Mobile Communications) networks, usually one TRX can serve 8 mobile devices using TDMA, and up to 12 TRXs can be installed on one antenna.

Depending on the available frequencies (especially near country borders), the extent of interference and the applied technology, we distinguish four types of constraints:

- *co-cell separation constraint*: the difference between frequencies assigned to the same cell has to be at least  $\gamma(v, v)$ . In most cases  $\gamma(v, v) = 3$ .
- *co-site separation constraint*: if the antennas  $v$  and  $u$  are located on the same site, then the difference between their frequencies has to be at least  $\gamma(v, u)$ . In general  $\gamma(v, u) = 2$ .
- *interference constraint*: due to other interference reasons, the difference between frequencies of antennas  $u$  and  $v$  has to be at least  $\gamma(v, u)$ .
  - If  $\gamma(v, u) = 1$  (i.e., they are not allowed to get the same frequency), then it is called a *co-channel constraint*.
  - If  $\gamma(v, u) = 2$  (i.e., they are not allowed to get even neighboring frequencies), then it is an *adjacent channel constraint*.
- *hand-over separation constraint*: at times, a mobile phone might have to switch to another server antenna. Therefore, in GSM systems, the *Broadcast Control Channels* (BCCH) are available. Their frequencies have to differ by at least two units from any other frequencies used by the concerned antennas.

### **2.1.2. Radio and television transmission**

The model is much simpler in this case. However, the prohibited frequency differences are not continual. In general, the frequency differences 1, 2, 5 and 14 are banned because of the corresponding harmonics.

### **2.1.3. Military applications**

In military applications, both parties might change their location. To all communication channels, two frequencies are assigned (one per communication direction). Their difference is constant, so the frequencies are ordered into pairs (with this constant difference), and these pairs are assigned to communication channels. The situation becomes more complicated by using horizontal and vertical polarization, hence the extent of interference is dependent not only on the geographical circumstances, but also on the polarization of signals.

### **2.1.4. Satellite communication**

The transmitter and receiver are located on the Earth, but they communicate through one or more satellites. First, all signals are conveyed to a satellite by means of uplink frequencies, and the receiver then gets the signals by means of downlink frequencies. As the difference between the downlink and uplink frequencies is fixed and considerably large, it is enough to focus on planning just one of them. Successive frequencies have to be assigned to transmitters, so that each of them might be used just once. In addition to the standard interference model (where only the interference between two signals is considered), efforts have been made to also take into account the interference caused by more than two signals as well [2].

### **2.1.5. Wireless local area networks**

Planning the frequencies of WLANs is one of the newest application domains [13]. WLANs allow mobile devices (for instance notebooks) to communicate with the help of an access point, which in turn is directly connected to a wired network or the internet. For the operation of such systems, only 13 frequencies are available, whose differences are 5 MHz. At the same time, frequencies must differ by at least 24 MHz to avoid interference. Thus, planning WLANs is often handled as a 3-frequency problem, in which the location of access points also plays a crucial role.

## **2.2. Models of frequency planning**

Based on the application domains presented above, researchers have developed different models of frequency planning [1]. First, we review the general constraints and then the differences in the models.

### 2.2.1. General constraints

Let  $V$  denote the set of transmitters (or antennas or communication channels) to which we have to assign frequencies. As mentioned, the set of frequencies is  $F = \{1, \dots, N\}$ , whilst  $F(v)$  denotes the frequencies available to a transmitter  $v \in V$ . ( $F(v) \subseteq F$ .)

- *multiplicity constraint*:  $m(v)$  frequencies have to be assigned to antenna  $v$ . In most cases,  $m(v) = 1$ , except for mobile phone networks.
- *interference matrix*: to each pair of antennas  $v, w$  and frequencies  $f \in F(v), g \in F(w)$ , a number  $p_{v,w}(f, g) \geq 0$  is defined, which is proportional to the extent of interference caused by assigning  $f$  to  $v$  and  $g$  to  $w$ . It is called:
  - *Co-channel penalty*, if  $|f - g| = 0$ .
  - *Adjacent channel penalty*, if  $|f - g| = 1$ .
- *maximum tolerable interference*: there is a given value  $p_{max}$ , and all interference above this level must be avoided. That is, if  $p_{v,w}(f, g) > p_{max}$ , then it is not allowed to assign frequency  $f$  to antenna  $v$  and frequency  $g$  to antenna  $w$  at the same time. The rationale behind such a hard constraint is that, although it is possible to minimize the overall interference in a network, it would not be acceptable if in one part of the network communication were hindered by too high local interference.
- *blocked channel*: for any reason, a frequency must not be used by a communication channel.

By setting the interference matrix and the maximum tolerable interference appropriately, an arbitrary *separation constraint* of  $s$  units (i.e., the frequencies of the transmitters  $v$  and  $u$  must differ by at least  $s$ ) can be achieved.

### 2.2.2. F-FAP

*Feasibility Frequency Assignment Problem*: given the constraints described above, the task is to decide whether there exists a solution satisfying all the conditions.

### 2.2.3. Max-FAP

*Maximum Service FAP*: if the F-FAP is not solvable, i.e. there is no solution satisfying all the conditions, we may settle for a slightly worse solution by relaxing the condition on the number of assigned frequencies and trying to maximize the number of assigned frequencies. On the other hand, even if the F-FAP is solvable, maximizing the number of assigned frequencies is still a valid objective, since the overall service quality could be improved this way.

Given the described constraints and a number  $l(v)$  to each antenna denoting the minimum number of frequencies that have to be assigned to  $v$ , we define new variables  $n(v)$  denoting the number of frequencies actually assigned to antenna  $v$ .

- If the F-FAP is solvable, then  $n(v) \geq m(v)$  should hold for all antennas, where  $m(v)$  is the previously defined multiplicity number. Providing also an upper bound to some  $n(v)$  is possible but not obligatory.
- If the F-FAP is not solvable, then we specify  $l(v) \leq n(v) \leq m(v)$ .

The aim is to maximize the overall service being made available, i.e. the total number of assigned frequencies  $\sum_{v \in V} n(v)$ . The lower bounds on each  $n(v)$  guarantee sufficient service quality in the whole system.

#### 2.2.4. MO-FAP

*Minimum order FAP:* given the general constraints as above, the aim is to minimize the number of used frequencies in the network. We include this model only for the sake of completeness, as nowadays operators pay for a frequency band and not for individual frequencies. Naturally, MO-FAP is only meaningful if the F-FAP is solvable.

#### 2.2.5. MS-FAP

*Minimum span FAP:* given the general constraints as above, the aim is to minimize the span of the used frequency band, i.e. the difference between the highest and the lowest frequency used. Obviously, F-FAP needs to be solvable in this case too. As opposed to MO-FAP, MS-FAP is a realistic model of operators' cost minimization efforts, and hence of great practical importance.

#### 2.2.6. MI-FAP

*Minimum interference FAP:* the aim is to minimize the total interference in the entire network:

$$\sum_{v,w \in V} \sum_{f \in F(v), g \in F(w)} p_{v,w}(f, g).$$

This is the most difficult sort of FAPs; its solution with an exact algorithm tends to take too long for practical applications, so that usually heuristic methods are used. As we focus on exact ILP-based algorithms in this chapter, we do not address MI-FAP in more depth.

In the next sections, we deal with the solution of the following FAP formulations: F-FAP, Max-FAP, MO-FAP and MS-FAP. F-FAP is the only decision problem, the others are optimization problems.

### 3. ILP formulations of FAP

We present two different approaches to formulating FAP problems as integer programs. The first approach makes use of a large number of binary variables, whereas the second approach uses a moderate number of integer variables.

### 3.1. Using binary variables

In all FAP versions, the following binary variables are defined for all  $v \in V$  and  $f \in F(v)$ :

$$x_{v,f} := \begin{cases} 1 & \text{if frequency } f \text{ is assigned to antenna } v, \\ 0 & \text{otherwise.} \end{cases}$$

The constraints and objective function depend on the FAP version, as shown below.

#### 3.1.1. F-FAP (Feasibility-FAP)

F-FAP is a decision problem, hence there is no objective function. The constraints are as follows:

$$\sum_{f \in F(v)} x_{v,f} = m(v) \leftarrow \forall v \in V \quad (1)$$

$$x_{v,f} + x_{w,g} \leq 1 \leftarrow \forall v, w \in V, f \in F(v), g \in F(w), p_{v,w}(f, g) > p_{max} \quad (2)$$

Here, (1) ensures that an adequate number of frequencies is assigned to all transmitters. (2) ensures the minimum necessary distance between frequencies of transmitter pairs by not allowing the assignment of frequency pairs to antenna pairs that would cause higher interference than the maximum tolerable interference.

In this formulation, the number of variables is  $|V| \cdot |F|$  (all of them are binary), and the number of constraints is at most  $|V| + |V|^2 \cdot |F|^2$ .

#### 3.1.2. Max-FAP (Maximum Service FAP)

As defined earlier, the task is to maximize the total number of assigned frequencies, where  $n(v)$ , the number of frequencies assigned to antenna  $v$ , is subject to given lower and upper bounds ( $l(v)$  and  $m(v)$ , respectively).

To create the ILP formulation, we extend the set of variables with the new integer variables  $n(v)$  for each antenna  $v$ . Thus, we obtain the following ILP formulation:

$$\text{Maximize: } \sum_{v \in V} n(v)$$

$$\sum_{f \in F(v)} x_{v,f} = n(v) \leftarrow \forall v \in V \quad (3)$$

$$l(v) \leq n(v) \leftarrow \forall v \in V \quad (4)$$

$$n(v) \leq m(v) \leftarrow \forall v \in V \quad (5)$$

$$x_{v,f} + x_{w,g} \leq 1 \leftarrow \forall v, w \in V, f \in F(v), g \in F(w), p_{v,w}(f, g) > p_{max} \quad (6)$$

Constraint (6) is the same as in the previous case. Constraints (4) and (5) guarantee that the number of frequencies assigned to a transmitter will be between the desired lower and upper bounds. It should be noted that the condition of the  $n(v)$  variables being integer will be automatically satisfied because of constraint (3). Thus, the  $n(v)$  variables need not be declared as integer.

The number of binary variables is  $|V| \cdot |F|$ , the number of other variables is  $|V|$ , and the number of constraints is at most  $3 \cdot |V| + |V|^2 \cdot |F|^2$ .

### 3.1.3. MO-FAP (Minimum order FAP)

Here, the aim is to minimize the number of used frequencies. Therefore, we augment the basic F-FAP model with one more binary variable for each frequency:

$$y_f := \begin{cases} 1 & \text{if frequency } f \text{ is used by at least one transmitter,} \\ 0 & \text{otherwise.} \end{cases} \quad (7)$$

Beside constraints (1) and (2) of the basic F-FAP model, the following extensions are necessary:

$$\text{Minimize: } \sum_{f \in F} y_f \quad (8)$$

$$y_f \geq x_{v,f} \leftarrow \forall v \in V, f \in F(v) \quad (9)$$

It should be noted that the  $y_f$  variables do not need to be declared binary or even integer. They will obtain the right value of 0 or 1 automatically:

**Proposition 1.** *In an optimal solution of the integer program (8)-(9), (1)-(2), the value of the  $y_f$  variables is – even if they are not declared as integer – as in (7).*

*Proof.* Consider an optimal solution. If  $f \in F$  is used by at least one communication channel, then  $\exists v$  with  $x_{v,f} = 1$ . Because of (9),  $y_f \geq 1$  follows. Since the solution is optimal,  $y_f = 1$  must hold.

If  $f$  is not used by any communication channel, then  $x_{v,f} = 0$  for each  $v \in V$ , thus (9) specifies  $y_f \geq 0$ . Taking the optimality of the solution into consideration,  $y_f = 0$  holds.  $\square$

In this model, the number of binary variables is  $|V| \cdot |F|$ , the number of further variables is  $|F|$ , whilst the number of constraints is at most  $|V| + |V| \cdot |F| + |V|^2 \cdot |F|^2$ .

### 3.1.4. MS-FAP (Minimum span FAP)

The difference between the highest and lowest used frequency is to be minimized. There are several ways to solve MS-FAP by means of ILP.

For instance, a series of F-FAPs can be solved with a decreasing range of frequencies. The optimal solution of the MS-FAP is the frequency band used by the last feasible F-FAP instance, before the F-FAPs turn infeasible.

It is also possible to model the MS-FAP as a single integer program. Even for this, there are several possibilities.

#### 1. General solution:

Define two new binary variables for each frequency:

$$u_f := \begin{cases} 1 & \text{if } f \in F \text{ is the highest used frequency,} \\ 0 & \text{otherwise.} \end{cases}$$

$$l_f := \begin{cases} 1 & \text{if } f \in F \text{ is the lowest used frequency,} \\ 0 & \text{otherwise.} \end{cases}$$

The objective function is:



$$\text{Minimize: } \sum_{f \in F} f \cdot u_f - \sum_{f \in F} f \cdot l_f$$

Similarly to the previous case, the basic FAP constraints (1) and (2) need to be supplemented with some others:

$$\sum_{f \in F} u_f = 1 \quad (10)$$

$$\sum_{f \in F} l_f = 1 \quad (11)$$

$$x_{v,f} + u_g \leq 1 \leftarrow \forall v \in V, f \in F(v), g \in F, f > g \quad (12)$$

$$x_{v,f} + l_g \leq 1 \leftarrow \forall v \in V, f \in F(v), g \in F, f < g \quad (13)$$

(10) and (11) guarantee that, from all the available frequencies, exactly one will be marked as highest used frequency and one as lowest used frequency. It follows that the objective function is exactly the difference between the highest and the lowest used frequency. Because of (12), no higher frequency can be assigned to any communication channel than the highest frequency used in the network, and analogously, due to (13), no lower frequency might be assigned than the lowest one used.

The number of variables in this formulation is  $|V| \cdot |F| + 2|F|$  (all of them binary), and the number of constraints is at most  $|V| + |V|^2 \cdot |F|^2 + 2|V| \cdot |F|^2 + 2$ , which is significantly more than in the previous cases.

## 2. Simplified solution:

We present here one more ILP formulation for a special case of MS-FAP. Here,  $F = \{1, 2, \dots, |F|\}$ , from which the frequencies  $\{1, 2, \dots, f_{max}\}$  are used and the objective is to minimize  $f_{max}$ . This model needs significantly fewer variables and constraints than the previous formulation. Moreover, if no globally blocked channels or locally blocked channels exist in a network, then assuming that 1 is the lowest used frequency means no restriction, as only the differences between the frequencies play a role, not their actual values. This formulation is somewhat similar to the one for MO-FAP shown earlier.

Necessary variables:

$$x_{v,f} := \begin{cases} 1 & \text{if frequency } f \text{ is assigned to transmitter } v, \\ 0 & \text{otherwise.} \end{cases}$$

$$y_f := \begin{cases} 1 & \text{if there is a frequency } g \geq f \text{ that is used by at least one transmitter,} \\ 0 & \text{otherwise.} \end{cases} \quad (14)$$

ILP representation:

$$\text{Minimize: } \sum_{f \in F} y_f \quad (15)$$

$$\sum_{f \in F(v)} x_{v,f} = m(v) \leftarrow \forall v \in V \quad (16)$$

$$x_{v,f} + x_{w,g} \leq 1 \leftarrow \forall v, w \in V, f \in F(v), g \in F(w), p_{v,w}(f, g) > p_{max} \quad (17)$$

$$y_f \geq x_{v,f} \leftarrow \forall v \in V, f \in F(v) \quad (18)$$

$$y_{f+1} \leq y_f \leftarrow \forall 1 \leq f \leq |F| - 1 \quad (19)$$

(16) and (17) are exactly as in F-FAP.

Similarly as with MO-FAP, the  $y_f$  variables need not be declared as binary or even integer, as they will automatically obtain the correct values of 0 or 1:

**Proposition 2.** *In an optimal solution of the integer program (15)-(19), the value of the  $y_f$  variables is – even if they are not declared as integer variables – as in (14).*

*Proof.* If there is a frequency  $g \geq f$  that is used by at least one transmitter, then we have  $y_g \geq 1$  because of (18). Applying (19)  $g - f$  times, we obtain  $y_f \geq 1$ . Since the solution is optimal,  $y_f = 1$  must hold.

If there is no frequency  $g \geq f$  that is used by at least one transmitter, then (18) results in  $y_g \geq 0$  for all  $g \geq f$ . Thus,  $y_f \geq 0$ , and (19) will not result in a stronger bound on  $y_f$ . Hence, in an optimal solution,  $y_f = 0$  holds.  $\square$

As a consequence,  $y_f = 1$  for all  $f \leq f_{max}$  and 0 afterwards. Hence, the objective function will be exactly  $f_{max}$ .

The number of variables in this formulation is  $|V| \cdot |F| + |F|$ , from which  $|V| \cdot |F|$  must be declared binary. The number of constraints is at most  $|V| + |V|^2 \cdot |F|^2 + |V| \cdot |F| + |F|$ .

## 3.2. Using integer variables

The ILP formulations presented so far make use of a large number of binary variables. In this section, we present ILP formulations with a smaller number of integer variables instead. The basic idea is to define one variable for each transmitter, the value of which is the frequency to be assigned to the given transmitter. The challenge in this approach is that separation constraints are given in terms of the difference between pairs of frequencies, e.g.  $|f_v - f_w| \geq s$ , which is not linear in the frequencies. Hence, we first need a way to linearize constraints involving absolute values.

### 3.2.1. Linearizing absolute values

This technique can be used in other ILP problems too, if one would like to linearize constraints that involve absolute values. Henceforward, let  $c$  be an arbitrary positive constant, and  $x$  a variable or a linear expression containing one or more variables.

1. Case  $|x| \leq c$ :

The linearization of this sort of constraint is quite simple, as it can be substituted with two linear inequalities:  $x \geq -c$  and  $x \leq c$ .

2. Case  $|x| \geq c$ :

At first sight, this kind of constraint cannot be linearized, because the set of feasible values of  $x$  is not convex. However, if  $x$  is bounded, then even this kind of constraint can be linearized by means of an auxiliary binary variable:

**Theorem 3.** *Assume that  $x$  is bounded and let  $M$  be a sufficiently large constant, so that  $M \geq |x| + c$  for each feasible value of  $x$ . Then  $x$  fulfils  $|x| \geq c$  if and only if it fulfils the following ILP program with a suitable choice of the new variable  $B$ :*

$$\begin{aligned} x + M \cdot B &\geq c \\ -x + M \cdot (1 - B) &\geq c \\ B &\in \{0, 1\} \end{aligned}$$

*Proof.* First, assume that  $|x| \geq c$  is fulfilled.

- If  $x \geq 0$ , then let  $B = 0$ . It can be easily verified that the constraints of the ILP program are fulfilled: the first because  $x \geq c$  and the second because of the choice of  $M$ .
- If  $x < 0$ , then let  $B = 1$ . Again, it can be easily verified that the constraints of the ILP program are fulfilled: the first because of the choice of  $M$  and the second because  $-x \geq c$ .

Second, assume that the constraints of the ILP program are fulfilled.

- If  $B = 0$ , then the first constraint of the ILP program guarantees that  $x \geq c$ .
- If  $B = 1$ , then the second constraint of the ILP program guarantees that  $-x \geq c$ .

In both cases, it follows that  $|x| \geq c$ . □

Next, we show ILP formulations based on this technique for some FAP variants.

### 3.2.2. F-FAP (Feasibility-FAP)

We define for each antenna as many integer variables as frequencies should be assigned to that antenna. Denote these variables by  $z_1, z_2, \dots, z_k$  ( $k \geq n$ ). Hence the minimum required distances between frequencies can be written as

$$|z_i - z_j| \geq s_{i,j} \leftarrow 1 \leq i, j \leq k,$$

where  $s_{i,j}$  are given positive constants. Furthermore, we define the constraints

$$f_{min} \leq z_i \leq f_{max} \leftarrow \forall 1 \leq i \leq k,$$

so that the assigned frequencies  $z_i$  are in the allowed range. It is also possible to require globally or locally blocked channels: e.g. the inequality

$$|z_i - f| \geq 1$$

assures that frequency  $f$  will not be assigned to  $z_i$ .

The constraints involving absolute values can be linearized as stated in Theorem 3. The condition that the expressions in absolute values must be bounded is fulfilled, since  $|z_i - z_j| \leq f_{max}$  and  $|z_i - f| \leq f_{max}$ .

### 3.2.3. MS-FAP (Minimum span FAP)

The above formulation for F-FAP can be easily extended to a formulation for MS-FAP. We define two new integer variables:  $sf$  for the lowest and  $gf$  for the highest frequency used by any transmitter in the network. In accordance with the meaning of  $sf$  and  $gf$ , the following constraints are added to the integer program:

$$gf \geq z_i \leftarrow \forall 1 \leq i \leq k$$

$$sf \leq z_i \leftarrow \forall 1 \leq i \leq k$$

The objective function is:

$$\text{Minimize: } gf - sf,$$

which is exactly the span of the used frequency band.

## 4. Empirical measurements

### 4.1. Implementation

In order to assess the practical behaviour of the different ILP models of the FAP, we implemented them in the BCAT (Budapest Complexity Analysis Toolkit) framework [12].

The aim of BCAT is to facilitate the development and testing of algorithms. It offers the possibility to set up different *problem classes*, *analyzers* of these problems, *converters* that transform the instances of a certain problem to the instances of another problem, and *algorithms* that solve the problems. Afterwards, one can give the settings of the program by means of a configuration file:

- what problem instances should be loaded or generated,
- into which problems should they be converted with the available converters,
- with what analyzers should the problems be analyzed,
- what algorithms should be applied to solve the different problem instances.

A further advantage of BCAT is that it delivers the output in structured files that can be easily processed e.g. in a spreadsheet application.

Beside the existing problem classes, analyzers, algorithms, and converters in BCAT, we implemented a FAP and an LP problem class with appropriate analyzers, converters, and algorithms. We used an external LP solver called `lp_solve`<sup>2</sup>.

#### 4.1.1. Testing process

As test cases, we used the COST259 benchmarks<sup>3</sup>. These include the data of different GSM 900 and GSM 1800 networks.

In order to use a machine-independent metrics, we measure complexity with the number of iterations of the ILP solver.

As mentioned previously, *NP*-complete problems often exhibit very large differences concerning the time to solve problem instances. We experienced this phenomenon even for problem instances that only differed slightly in a single parameter. For example, it often occurred that a problem instance was solvable in half a minute, but after increasing the maximum tolerable interference by 0.01, we did not get a result even in several hours. Due to 'unfortunate cases' (see Section 4.5. for more detail), the run of the algorithm may take several weeks. For this reason, we were forced to use timeout values, i.e. we stopped the algorithm, even if it was not finished, after the defined timeout. In these cases, what we measured as complexity is actually only a lower bound on the number of iterations necessary to solve the given problem instance.

In order to decrease the impact of such 'unfortunate cases' and 'noise', we repeated the tests multiple times. Generally, we solved each problem instance ten times, and calculated the median of the ten results. Generally, this method works better than taking the average value of the runtimes, as a longer runtime could abnormally increase the average [5].

## 4.2. Complexity of F-FAP

In the following, let  $n$  denote the number of variables and  $m$  the number of constraints in the given ILP formulation (see Section 3. for details). Furthermore, let  $k$  denote the number of available frequencies in the network.

### 4.2.1. Constant number of frequencies and communication channels

In our first set of experiments, we kept  $n$  constant and varied  $m$  by means of changing the maximum tolerable interference. Figures 1 and 2 show the results. These tests were carried out using the data of a GSM network consisting of 20 cells and 5 frequencies ( $\Rightarrow n = 100$ ). The number of constraints ( $m$ ) was changing between 100 and 1300. We used the standard ILP formulation with binary variables for F-FAP. On the right side of the diagram, the vertical axis shows the number of iterations (IterNumber) used by the algorithm, while the left vertical axis presents the solvability (Solvable). If this value equals zero, then all problem instances with the given  $m$  value were unsolvable; if this value equals one, then

<sup>2</sup>`lp_solve` is an open-source LP and ILP solver. Further information as well as downloadable source and binary files are available from <http://sourceforge.net/projects/lpsolve/>.

<sup>3</sup>Downloadable from <http://fap.zib.de/>.

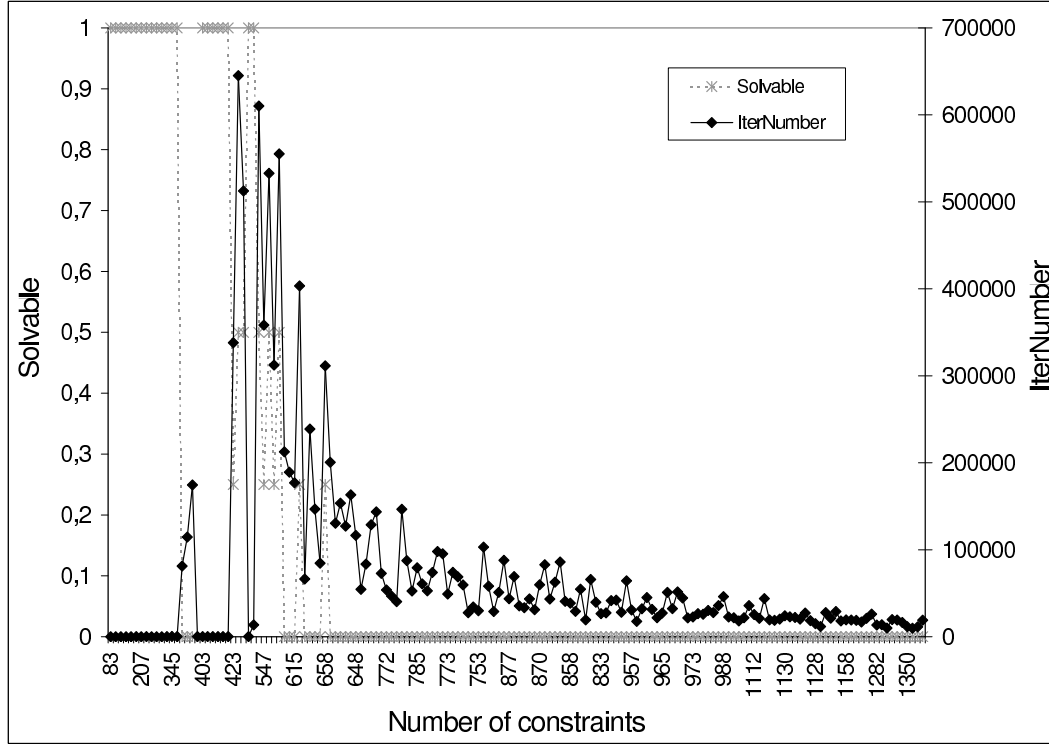


Figure 1. Complexity of F-FAP without globally or locally blocked channels, where  $n = 100$  and  $k = 5$ .

the algorithm found a solution in all cases. If this value is between zero and one, then the algorithm was stopped prematurely because of the timeout. Hence, in these cases, the given number of iterations is only a lower bound on the actual number of iterations that are necessary.

The complexity pattern of the Figure is in line with what is known as the 'phase-transition phenomenon' [4, 6, 7]. Briefly, this means that for small values of the constraints/variables ratio (underconstrained case), almost all problem instances are solvable. When the number of constraints increases, the ratio of solvable problem instances drops relatively abruptly from almost 1 to almost 0 (phase transition). After this critical regime, almost all problem instances are unsolvable (overconstrained case). In the underconstrained case, the problem is easy: even simple heuristics usually find a proper solution. In the overconstrained case, it is easy for backtracking algorithms to prove unsolvability because they quickly reach contradiction. The hardest instances lie in the critical regime [4]. This phenomenon has been described in the literature for some theoretical problems (e.g., graph coloring [11, 14]). Our findings indicate that it also applies to FAP.

The only difference between Figure 1 and Figure 2 is that in the measurements depicted in Figure 2, there are also several globally and locally blocked channels in the network. As can be seen, the complexity patterns in the two figures barely differ from each other. We can conclude that including blocked channels does not make the problem significantly harder.

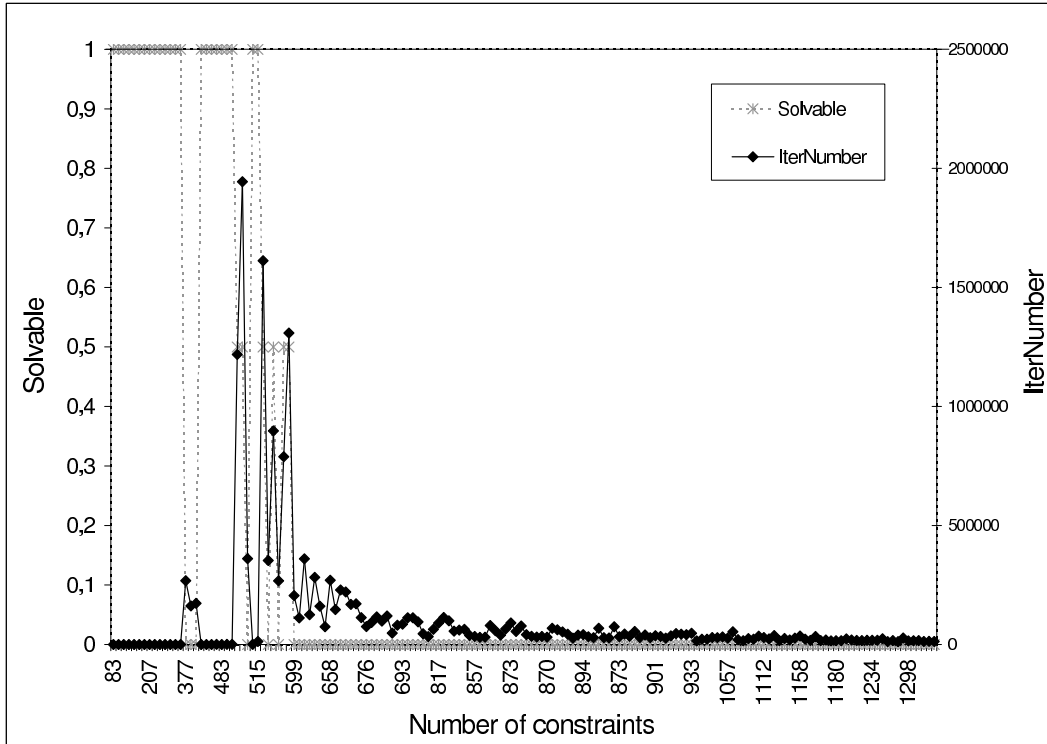


Figure 2. Complexity of F-FAP in presence of globally and locally blocked channels, where  $n = 100$  and  $k = 5$ .

#### 4.2.2. Varying number of communication channels

Figure 3 also shows the solvability (Solvable) and the complexity (IterNumber), but this time as a function of the number of cells, with a constant number of frequencies and constant maximum tolerable interference value. Note that increasing the number of cells increases the number of communication channels and thus the number of variables and constraints as well. In the Figure, we can again recognize an easy-hard-easy pattern, with significant variance within the hard regime. It is interesting to note that increasing the number of communication channels does not increase the complexity beyond some threshold. On the contrary, complexity decreases to very low values after the hard regime. The reason for this phenomenon is probably that in the unsolvable case, with increasing number of cells, the algorithm is likely to reach contradiction earlier, i.e. on a higher level of the search tree.

#### 4.2.3. Varying number of frequencies

In Figure 4, we present an analogous diagram for the case of varying number of frequencies. Here, the number of communication channels was constant. Note that increasing the number of frequencies increases both the number of variables and the number of constraints. Nevertheless, we can again witness an easy-hard-easy pattern. If the number of frequencies is high, then the problem is easy, regardless of its huge size.

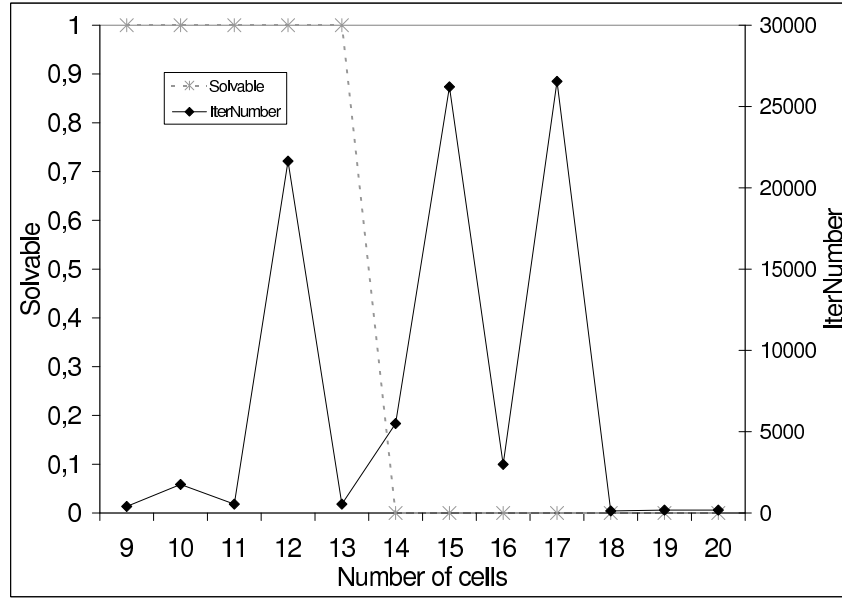


Figure 3. Complexity of FAP as a function of the number of cells.

### 4.3. Comparing different ILP formulations

For F-FAP, we introduced two essentially different ILP formulations: one of them used binary variables, while the other used integer variables. On the ground of the empirical examinations, applying the ILP formulations using binary variables, the problem instances could be solved significantly more efficiently than using integer variables and linearizing the expressions involving absolute values. The binary representation needs far more variables, but this is likely compensated by the smaller domain of the variables.

Table 1. Number of problem instances that could be solved by the two solution techniques for MS-FAP.

		Binary variables	
		<i>successful</i>	<i>unsuccessful</i>
Integer variables	<i>successful</i>	46	25
	<i>unsuccessful</i>	11	18

In the case of MS-FAP, the situation is different. Applying 10 minutes as timeout, we analyzed the solution of 100 instances with entirely different parameters. As can be seen in Table 1, there were 25 cases when the formulation with integer variables proved to be better, and only 11 cases when the formulation with binary variables won. So for MS-FAP, the formulation with integer variables seems to be more suitable.



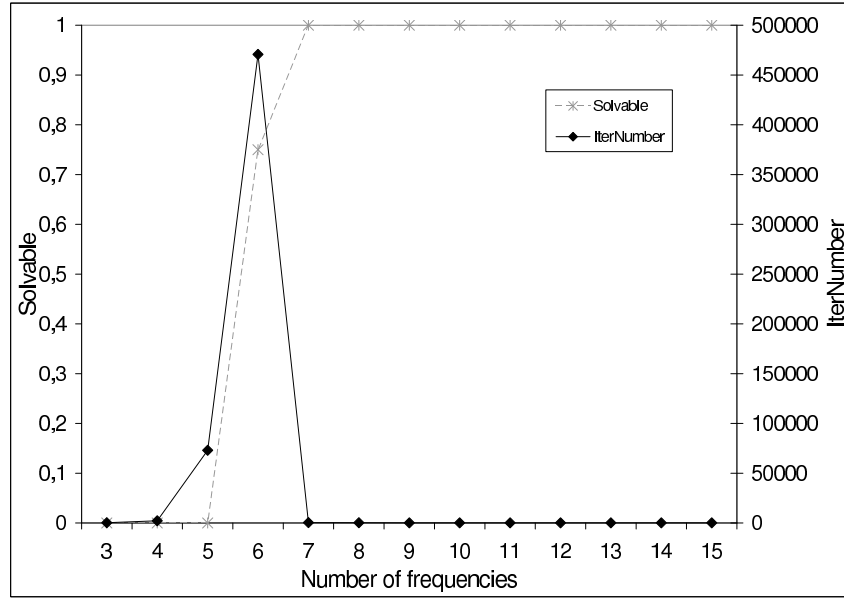


Figure 4. Complexity of FAP as a function of the number of frequencies.

#### 4.4. Sub-optimal solutions of the optimization versions of FAP

Optimization FAP problems are extremely time-consuming, that is why we considered sub-optimal solutions too. We examined when the algorithms should be stopped in order to find a nearly optimal solution in a relatively short time.

Table 2. Best solutions of Max-FAP for different timeout values

Runtime	Best solution found					
	44	43	42	44	40	43
half min.	44	43	42	44	40	43
1 min.	45	46	44	44	39	46
2 min.	44	45	43	45	38	45
5 min.	44	44	43	46	42	44
10 min.	45	44	44	43	40	44
quarter hour	44	45	43	45	40	43
half hour	43	46	43	45	39	44
1 hour	44	45	44	46	43	45
2 hour	44	46	43	45	42	44

Table 2 shows the best solutions found within given time limits for some Max-FAP problem instances. Each column shows the test results corresponding to one problem instance. For instance, the first column shows that the algorithm succeeded to assign 44 frequencies before being aborted after half a minute. Similarly, in one minute 45, in two

minutes 44, . . . , in two hours 44 frequencies could be assigned. Apparently, despite the increasing runtime, the results improved only a little bit or even did not improve. In other words: aborting the run as early as after half a minute, the best solution found so far will not be far from the optimum.

Table 3. Best solutions of MO-FAP for different timeout values

<b>Runtime</b>	<b>Best solution found</b>					
half min.	24	23	13	11	–	30
1 min.	–	21	12	12	37	28
2 min.	–	22	–	11	34	29
5 min.	22	22	11	11	30	28
10 min.	22	22	12	12	-	28
quarter hour	–	22	14	12	27	27
half hour	–	25	12	12	32	29
1 hour	29	22	13	11	32	27
2 hour	24	–	–	11	29	29

The same phenomenon also applies to MO-FAP (see Table 3). Where a dash is shown, it means that the algorithm was not able to find any solution within the given time limit.

#### 4.5. Acceleration with restarts

If an algorithm involves random choices, it might make sense to run it several times on a given problem instance. Empirical evidence on several NP-hard problems shows that there is often several orders of magnitude variance in algorithm runtime on the same problem instance. For example, suppose that the median runtime of a random algorithm on problem instances of a given size is 1 minute. Assume that it has been running on a problem instance for 5 minutes without any results yet. Intuitively, one could think that the algorithm will most probably finish very soon, so we should keep waiting. However, empirical evidence shows that – generally for NP-hard problems – it is better to stop the current run of the algorithm and restart it [8]. The rationale is that it might actually happen with surprisingly high probability that the current run of the algorithm will take several hours, days, or even longer. On the other hand, if we restart the algorithm, chances are high that the next run will be more fortunate and may finish in a minute or so.

So far, this phenomenon has been mainly investigated for theoretical problems (e.g., satisfiability, graph coloring). We examined whether it is also beneficial to apply this strategy in solving FAPs. For this purpose, we had to randomize our algorithm. Luckily, `lp_solve` offers the possibility to randomize the order in which variables are considered.

We compared two versions of the algorithm on 34 instances. In the *standard method*, we ran the algorithm once, with a timeout of 900 seconds. In the *restart method*, we started the algorithm with a timeout of only 90 seconds; if it did not finish by that time, it was restarted again with 90 seconds timeout etc. We allowed up to 10 runs, so that the total time budget for a problem instance was 900 seconds for this method as well.

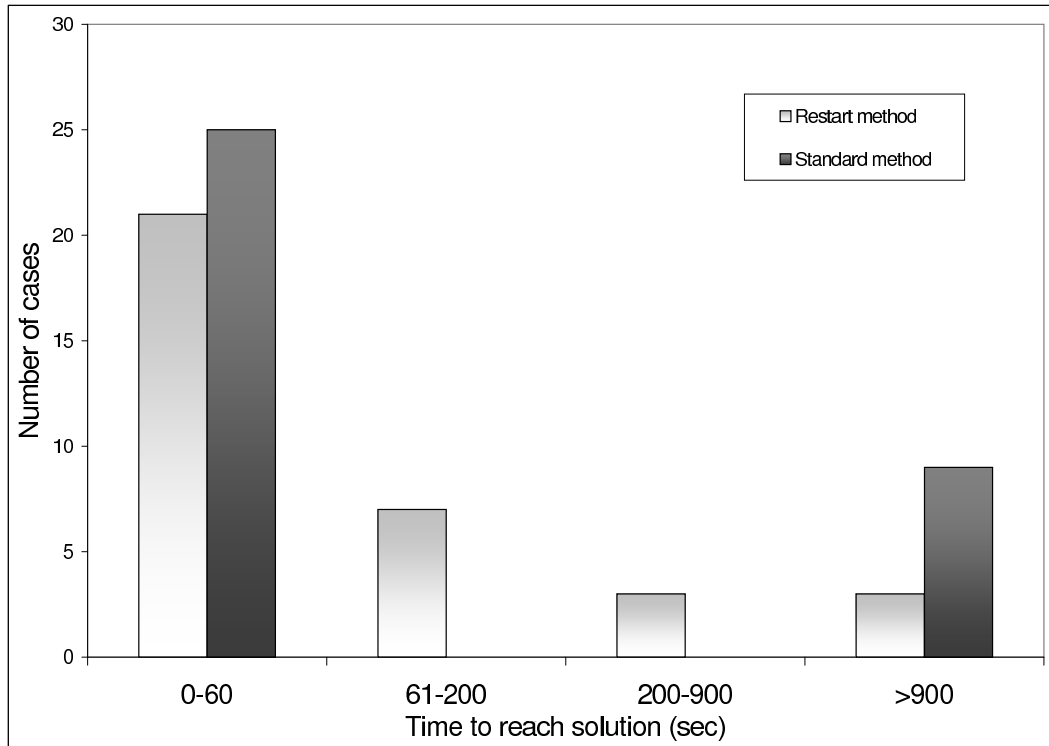


Figure 5. Histogram of runtime using standard vs. restart method.

The results are summarized in Figure 5. As can be seen, the standard method suffers from the large variability in algorithm runtime mentioned above: 25 problem instances could be solved within 60 seconds, but the remaining 9 problem instances could not be solved even in 900 seconds. With the restart method, the variance is much lower, and hence there are only 3 problem instances that cannot be solved within 900 seconds. Moreover, the average runtime of the restart method was about 50% of the average runtime of the standard method. All in all, we can conclude that it is worth using restarts for solving FAPs. Elaborating the best restart strategy (i.e. the best timeout values) is an important topic for future research.

## 5. Conclusion

In view of the rapid development of wireless networks, we considered a general model of frequency assignment problems. In this chapter, we presented different techniques to solve FAPs by means of ILP, and examined their efficiency empirically.

In line with research on other NP-hard problems, we were also faced by several orders of magnitude variance in algorithm runtime. Our results help in predicting algorithm runtime based on model parameters (number of constraints, number of cells, number of frequencies). In cases where the time needed to optimally solve a problem instance is unrealistically high, the algorithm can be stopped prematurely by specifying a timeout. Our results show that the value of the objective function hardly changes after some time, so that such a premature finish will usually result in a solution that is quite close to the optimum.

Moreover, we analyzed which ILP formulations are worth to use. In case of F-FAP, the ILP formulation using binary variables turned out to be more efficient. However, in case of MS-FAP, the ILP formulation with integer variables proved to be better. Additionally, we showed that restarting the solver at certain times can significantly improve the algorithm's behavior.

## Acknowledgements

This work was partially supported by the Hungarian National Research Fund and the National Office for Research and Technology (Grant Nr. OTKA 67651).

## List of abbreviations

BCAT	Budapest Complexity Analysis Toolkit
BCCH	Broadcast Control Channel
BIP	Binary Integer Programming
DCA	Dynamic Channel Assignment
FAP	Frequency Assignment Problem
FCA	Fixed Channel Assignment
F-FAP	Feasibility Frequency Assignment Problem
GPS	General Positioning System
GSM	General System for Mobile Communication
HCA	Hybrid Channel Assignment
ILP	Integer Linear Programming
ITU	International Telecommunication Union
LP	Linear Programming
Max-FAP	Maximum Service FAP
MI-FAP	Minimum interference FAP
MIP	Mixed Integer Programming
MO-FAP	Minimum order FAP
MS-FAP	Minimum span FAP
TDMA	Time Division Multiple Access
TRX	Transmitter/Receiver Unit
WI-FI	Wireless Fidelity
WLAN	Wireless Local Area Network

## References

- [1] Karen I. Aardal, Stan P.M. van Hoesel, Arie M.C.A. Koster, Carlo Mannino, and Antonio Sassano. Models and solution techniques for frequency assignment problems. *Annals of Operations Research*, 153(1):79–129, 2007.
- [2] Sara Alouf, Eitan Altman, Jérôme Galtier, Jean-Francois Lalande, and Corinne Touati. Quasi-optimal bandwidth allocation for multi-spot MFTDMA satellites. *IEEE info-com*, 1:560–571, 2005.
- [3] R. Borndörfer, A. Eisenblätter, M. Grötschel, and A. Martin. Frequency assignment in cellular phone networks. *Annals of Operations Research*, 76:73–93, 1998.
- [4] Peter Cheeseman, Bob Kanefsky, and William M. Taylor. Where the really hard problems are. In *12th International Joint Conference on Artificial Intelligence (IJCAI '91)*, pages 331–337, 1991.
- [5] Carla P. Gomes, Bart Selman, Nuno Crato, and Henry Kautz. Heavy-tailed phenomena in satisfiability and constraint satisfaction problems. *Journal of Automated Reasoning*, 24(1-2):67–100, 2000.
- [6] Tad Hogg. Refining the phase transition in combinatorial search. *Artificial Intelligence*, 81(1-2):127 – 154, 1996.

- [7] Tad Hogg and Colin P. Williams. The hardest constraint problems: A double phase transition. *Artificial Intelligence*, 69(1-2):359–377, 1994.
- [8] Malik Magdon Ismail and Amir F. Atiya. The early restart algorithm. *Neural Computation*, 12(12):2991–3010, 2000.
- [9] Haixia Jia and Cristopher Moore. How much backtracking does it take to color random graphs? Rigorous results on heavy tails. In *Principles and Practice of Constraint Programming (CP 2004)*, pages 742–746, 2004.
- [10] A. M. C. A. Koster. *Frequency assignment - models and algorithms*. PhD thesis, University of Maastricht, 1999.
- [11] Zoltán Ádám Mann and Anikó Szajkó. Improved bounds on the complexity of graph coloring. In *Proceedings of the 12th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing*, pages 347–354, 2010.
- [12] Zoltán Ádám Mann and Tamás Szép. BCAT: A framework for analyzing the complexity of algorithms. In *8th IEEE International Symposium on Intelligent Systems and Informatics*, pages 297–302, 2010.
- [13] Janne Riihijävi, Marina Petrova, and Petri Mähönen. Frequency allocation for WLANs using graph colouring techniques. In *Proceedings of the 2nd Annual Conference on Wireless On-demand Network System and Services*, pages 216–222, 2005.
- [14] Tamás Szép and Zoltán Ádám Mann. Graph coloring: the more colors, the better? In *Proceedings of the 11th IEEE International Symposium on Computational Intelligence and Informatics*, pages 119–124, 2010.