Konrad-Zuse-Zentrum
für Informationstechnik Berlin

Takustraße 7
D-14195 Berlin-Dahlem
Germany

KAREN I. AARDAL
STAN P.M. VAN HOESEL
ARIE M.C.A. KOSTER
CARLO MANNINO
ANTONIO SASSANO

Models and Solution Techniques for Frequency Assignment Problems

ZIB-Report 01–40 (December 2001)

Models and Solution Techniques for Frequency Assignment Problems

Karen I. Aardal* Stan P.M. van Hoesel† Arie M.C.A. Koster‡

Carlo Mannino§ Antonio Sassano§

December 28, 2001

Abstract

Wireless communication is used in many different situations such as mobile telephony, radio and TV broadcasting, satellite communication, and military operations. In each of these situations a frequency assignment problem arises with application specific characteristics. Researchers have developed different modeling ideas for each of the features of the problem, such as the handling of interference among radio signals, the availability of frequencies, and the optimization criterion.

This survey gives an overview of the models and methods that the literature provides on the topic. We present a broad description of the practical settings in which frequency assignment is applied. We also present a classification of the different models and formulations described in the literature, such that the common features of the models are emphasized. The solution methods are divided in two parts. Optimization and lower bounding techniques on the one hand, and heuristic search techniques on the other hand. The literature is classified according to the used methods. Again, we emphasize the common features, used in the different papers. The quality of the solution methods is compared, whenever possible, on publicly available benchmark instances.

Research carried out with financial support of the project TMR-DONET nr. ERB FMRX-CT98-0202 of the European Community.

*Department of Mathematics, Utrecht University, P.O. Box 80.010, 3508 TA Utrecht, The Netherlands.

†Department of Quantitative Economics, Universiteit Maastricht, P.O.Box 616, 6200 MD Maastricht, The Netherlands.

‡Konrad-Zuse-Zentrum für Informationstechnik Berlin, Takustraße 7, D-14195 Berlin, Germany. Corresponding author, e-mail: koster@zib.de

§Dipartimento di Informatica e Sistemistica, Università di Roma "La Sapienza", Via Buonarroti 12, 00185, Roma, Italy.

Keywords: frequency assignment

Mathematics Subject Classification (MSC 2000): 90-02, 90C35, 05C90

1 Introduction

The literature on frequency assignment problems, also called channel assignment problems, has grown quickly over the past years. This is mainly due to the fast implementation of wireless telephone networks (e.g., GSM networks) and satellite communication projects. But also the renewed interest in other applications like TV broadcasting and military communication problems inspired new research. These applications lead to many different models, and within the models to many different types of instances. However, all of them share two common features:

(i) A set of wireless communication connections (or a set of antennae) must be assigned frequencies such that data transmission between the two endpoints of each connection (the receivers) is possible. The frequencies should be selected from a given set that may differ among connections.

(ii) The frequencies assigned to two connections may incur interference to one another, resulting in quality loss of the signal. Two conditions must be fulfilled in order to have interference of two signals:

  (a) The two frequencies must be close on the Electromagnetic band (or harmonics—Doppler effects—of one another).

  (b) The connections must be geographically close to each other, so that the interfering signal is powerful enough to disturb the original signal.

Frequency assignment problems (FAPs) first appeared in the 1960s [99]. The development of new wireless services such as the first cellular phone networks led to scarcity of usable frequencies in the radio spectrum. Frequencies were licensed by the government who charged operators for the usage of each single frequency separately. This introduced the need for operators to develop frequency plans that not only avoided high interference levels, but also minimized the licensing costs. It turned out that it was far from obvious to find such a plan. At this point, operations research techniques and graph theory were introduced. Metzger [99] usually receives the credits for pointing out the opportunities to use mathematical optimization, especially graph coloring techniques, for this purpose.

Until the early 1980s, most contributions on frequency assignment used heuristics based on the related graph coloring problem. First lower bounds were derived by Gamst and Rave [57] in 1982 for the most used problem of that time (cf. Section 4). The development of the digital cellular phone standard GSM (General System for Mobile Communication) in the late 1980s and 1990s led to a rapidly increasing interest for frequency assignment (see Eisenblätter [46] for a discussion of the typical frequency planning problems in GSM networks). But also projects on other applications such as military wireless communication and radio/TV broadcasting contributed to the literature on frequency assignment in recent years. So far, we only discussed Fixed Channel Assignment (FCA), i.e., static models where the set of connections remains stable over time. Opposite to FCA, Dynamic Channel Assignment (DCA) deals with the problem, where the demand for frequencies at an antenna varies over time. Hybrid Channel Assignment (HCA) combines FCA and DCA: a number of frequencies have to be assigned beforehand, but space in the spectrum has to be reserved for the online assignment of frequencies upon request. In this survey, we concentrate on FCA. We refer to Katzela and Naghshineh [79] for a recent survey on the topics DCA, HCA, and their relation with FCA.

This paper is not the first survey on the frequency assignment problem. In 1980, Hale [61] presented an overview of the frequency planning problems of that time, with a special focus on modeling the problems. Hale also introduced the relation of the FAP with graph (vertex) coloring. Since then, new applications have led to new variants of the problem. Moreover, the T-coloring problem introduced by Hale and other generalizations of graph coloring motivated by FAPs have received a lot of attention, resulting in many new (graph-theoretic) results. Roberts [115] surveyed the results on T-coloring in the early 1990s. The survey of Murphey et al. [101] also concentrates on the results for coloring generalizations that were motivated by frequency assignment. In Jaumard et al. [75], a brief description of several exact methods is presented. The survey in [85, Chapter 2] served as a starting point of this overview. Finally, Eisenblätter et al. [47] give an overview of the evolution of frequency planning from graph coloring and its generalizations to the models used nowadays, with an emphasis on the GSM practice. In this survey, we also restrict ourselves to models that are directly motivated from practice, and their solution methods.

Our focus is mainly on the practical relevance of mathematical optimization techniques for frequency assignment. In the next section we will discuss the practical settings of frequency planning mentioned above. Moreover, we will model this in such a way that the common features are emphasized. In Section 3, we will categorize the models in four standard classes. These categories mainly differ in the objective to be optimized. For each of the models, the subsequent sections will discuss:

(i) structural properties of the models, including bounding techniques based on (combinatorial) relaxations (Section 4),

(ii) exact optimization methods, such as branch-and-cut, branch-and-price, and combinatorial enumeration (Section 4 as well), and

(iii) heuristic methods, such as local search (including simulated annealing and tabu search), genetic algorithms, neural networks, constraint programming, and ant colony algorithms. (Section 5).

The paper is concluded with a discussion of the results obtained for available benchmark instances in Appendix A. For convenience, all discussed papers are summarized in a schematic way, in Appendix B.

Although we invested much effort in collecting as many papers on the topic as possible, it is impossible to guarantee completeness. Moreover, new publications will reduce the actuality of this survey. Therefore, updates of this survey, in particular of the digest in Appendix B, will appear at the web-site *FAP web* (http://fap.zib.de) [48]. This site also serves as a platform for announcing new papers on frequency assignment.

2 Models and Applications

The models discussed in the literature differ in the frequency choices for connections (or antennae) for two reasons, mainly. The available set of frequencies differs among applications, as well as the ways of handling interference. We will describe the practical settings of known applications, and the abstractions assumed in the accompanying models that lead to the models described in the literature. The models are discussed both in their common features and their differences. This section starts with a description of the most important practical issues involved with frequency assignment. Then an overview of situations in which frequency assignment problems occur is provided, including application specific characteristics. In the final subsection, we discuss the models that are derived from these practical situations.

2.1 Modeling the Frequency Assignment Problem: Practical Background

The availability of frequencies from the radio spectrum is regulated by the national governments, and world-wide by the International Telecommunication Union (ITU). Operators of wireless services are licensed to use one or more frequency bands in specific parts of a country. The frequency band $[f_{min}, f_{max}]$ available to some provider of wireless communication is usually partitioned into a set of channels, all with the same bandwidth Δ of frequencies. For this reason the channels (actually the channels are often called frequencies) are usually numbered from 1 to a given maximum N, where $N = (f_{max} - f_{min})/\Delta$. The available channels are denoted by $F = \{1, \ldots, N\}$. If more than one frequency band is available each band has its own set of consecutively numbered channels. In case of two bands, the frequencies can be numbered by $F = \{1, \ldots, N_1, N_2 + 1, \ldots, N_3\}$, where usually N_1 is so much smaller than N_2 that the two bands do not have any influence on each other. For a particular connection or antenna not all channels from F might be available. For instance, if a connection is close to the border of a country, division rules between the countries involved may lead to a substantial reduction in channel availability. Therefore, the channels available for a connection or antenna v form a subset $F(v)$ of F.

Interference of signals is measured by the signal-to-noise ratio, or interference ratio, at the receiving end of a connection. There, the signal of the transmitting end should be clearly understandable. The noise comes from other signals broadcasted at interfering frequencies. In general, the level of interference rapidly decreases with the distance between the frequencies. But, the actual signal-to-noise ratio at a receiver depends not only on the choice of frequency, but also on the strength of the signal, the direction it is transmitted to, the shape of the environment, and even weather conditions. It is therefore hard to obtain an accurate prediction of the signal-to-noise ratio at receivers. A first simplification is to ignore the environment and assume an omni-directional antenna. Now, consider two signals, one original and some other signal transmitted at the same frequency channel. Then the interference of the second signal at the receiver of the first signal is computed with the following formula: $\frac{P}{d^\gamma}$ where P is the power of the interfering transmitter and d its distance to the disturbed receiver. γ is a fading factor with values between 2 and 4. Its value depends on the frequency used. For instance the 1800 MHz band frequencies fade faster than the 900 MHz band frequencies both used in GSM networks. If the second signal is transmitted on a frequency at a distance of $n \geq 1$ units from the original signal, then an additional filtering factor of $-15(1 + \log^2 n)$ dB is taken into account (see [43]). There may be more than one source that transmits on the same or a close frequency and thus contributes to the total noise experienced at the receiver. The fact that multiple signals may disturb communication quality is ignored in most models where only interference between pairs of connections or antennae is measured. Notable exceptions are [53], in which constraints are developed to determine the total interference from neighboring connections, and [44], where combinations of frequencies for more than two transmitters are forbidden. We will generally ignore multiple interference. So it becomes a binary relation: only two connections or antennae are involved.

In mobile telephony and radio-TV broadcasting, the receivers are spread within a certain area. The standard approach of determining signal strength at all locations in the area is the following.

(i) A grid of squares of predetermined (small) size, the *test points* or *pixels*, is designed to overlap the area.

(ii) For each test point, the levels of the received signals generated by the serving transmitter, typically the one with strongest received signal (*best server)*, and by the interfering transmitters are predicted with a wave propagation model. Test points with same best server can be clustered to service areas, resulting in pictures like the one in Figure 1.

(iii) For a single transmitter A, and a given interfering transmitter B, the noise generated by B in each pixel of the service area of A is aggregated to a single value, which represents the interference of B over A.

The way noise is predicted and aggregated strongly depends on the application considered. For precise descriptions of wave propagation models used for this task see [33].

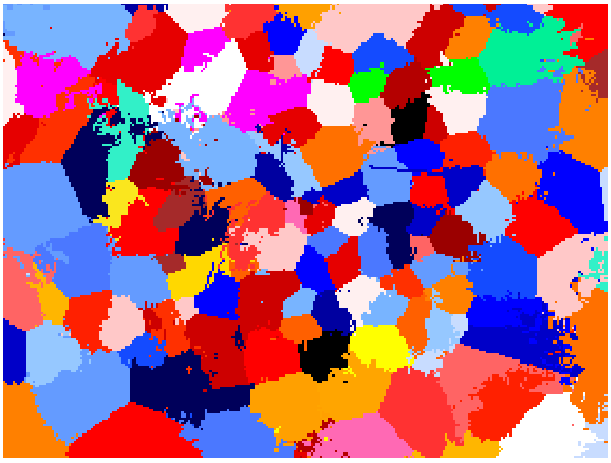


Figure 1: Best-server areas in a GSM network. Provided by E-Plus Mobilfunk GmbH.

In the past, more simplified prediction models were used: a standard approach was to use a grid of hexagons overlapping the area of interest and to consider the transmitters to be located at the center of each hexagon. The well-known Philadelphia instances (cf. Section 2.2) have this structure (see Figure 4, page 8). In the basic model for the hexagonal grids, interference of cells is characterized by a co-channel reuse distance d. No interference occurs if and only if the centers of two cells have mutual distance $\geq d$. In case the mutual distance is less than d (normalized by the radius of the cells), it is not allowed to assign the same frequency to both cells. This pure co-channel case is generalized by replacing the reuse distance d by a series of non-increasing values $d^0, \ldots, d^k$ and corresponding forbidden sets $T^0 \subseteq \ldots \subseteq T^k$. The following relation holds:

$$T_{vw} = T^{j-1} \text{ whenever } d^j \leq d_{vw} < d^{j-1}, j \in \{1, \ldots, k\}$$

where d_{vw} is the distance between the cell centers and T_{vw} denotes the set of forbidden differences for frequencies assigned to v and w, i.e., $|f_v - f_w| \notin T_{vw}$. For the variations of the original Philadelphia instance, the sets T^j are taken as $T^j = \{0, \ldots, j\}$. For example, the values $d^0, \ldots, d^5$ are $2\sqrt{3}, \sqrt{3}, 1, 1, 1, 0$. So, frequencies assigned to the same site should be separated by at least 4 other frequencies, whereas frequencies assigned to adjacent sites should be at a distance of at least 2, and frequencies assigned to a second and third 'ring' of cells should still differ, see Figure 2. In case $T_{vw} = \{0, \ldots, j\}$, alternatively the notation $|f_v - f_w| \geq \delta(v, w)$ is used where $\delta(v, w) = j + 1$, the minimum required difference.

A final aspect to be taken into consideration is two-way traffic. Except for radio and TV broadcasting all traffic is bidirectional, and one needs two channels, one for each direction. In the models considered in the literature the second channel is almost always ignored, with a notable exception in military applications, see Section 2.2. The reasons for ignoring this aspect of the FAP depend on the application. In most applications two bands of N channels are available: one with the channels $\{1, \ldots, N\}$, and one with the channels $\{s + 1, \ldots, s + N\}$, where $s \gg N$. Thus, the backward connection uses a channel which is shifted s channels up. The choice of s prevents any interference of backward channels with forward channels. Moreover, the symmetry of the solution for the backward channels, with the forward channels (plus s channels) leads to (almost) the same interference pattern for the backward channels. If these conditions are not fulfilled, the two-way traffic poses a problem, since interference need not be symmetric. The next example shows that the above conditions are not sufficient for symmetric interference. Consider the geographic positioning of transmitters in Figure 3. Suppose transceiver pair (a, b) transmits on frequencies f from a to b and $f + s$ from b to

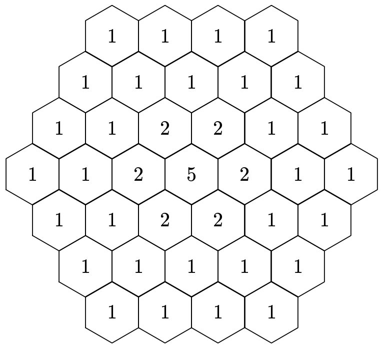


Figure 2: Example of reuse distances in hexagonal cell networks. The values denote the minimum separation distance in relation to the central cell.

a, and another transceiver pair (c, d) transmits on frequencies g from c to d and $g + s$ from d to c where f and g interfere, and $f + s$ and $g + s$ interfere. Now signal strength of g at a is much higher than signal strength of $g + s$ at b, since the receivers have different distances to c and d. Since in most models the assignment of frequencies to the backward direction is ignored, this aspect is not taken into account.

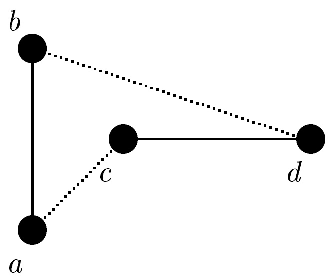


Figure 3: Example of asymmetric interference for bidirectional wireless communication

In mobile telephone networks, the backward interference is not employed for another supplementary reason: the location of the transmitters (the mobile users) is not static but varies over time, which makes it almost impossible to give an accurate prediction of the interference at the receiving end (base station).

Also in mobile telephone networks, in particular in GSM networks, the technique of *frequency hopping* has been introduced to reduce the influence of interference. Frequency hopping permits a transmitter to change the frequency of the signal according to a sequence of assignment frequencies. By rapidly changing the frequency for transmission, the overall interference level can be reduced. In general, still a generalized frequency assignment problem has to be solved. For more information about frequency hopping, we refer to [16, 56, 104].

2.2 Applications

There are various models and problem instances. The practical setting can vary enormously. This leads not only to different variants of the above model, but also to different instance types, see Hale [61]. Some of the settings are:

Mobile telephony In this application one of the endpoints of the connection is a fixed antenna, and the other endpoint is a mobile phone. Each antenna covers a certain area, where it can pick up signals from mobile phones. Each antenna covers a specific region (*cell*) and can serve several mobile units simultaneously. In particular, in TDMA (Time Division Multiple Access) each available frequency can be used to serve n different mobile units; in addition, several frequencies can be assigned to the same antenna (by the use of multiple transmitter/receiver units, TRXs), so that the number of different mobile units that are served can be very large. More antennae are then mounted on a same physical support (*site*) to cover a number of adjacent regions. In GSM networks, typically 8 mobile units can be served simultaneous by TDMA, whereas up to 12 TRXs can be installed at an antenna, for more details, we refer to [46].

The frequencies assigned to each antenna must satisfy a number of requirements that depend on (i) availability, especially at country borders; (ii) interference levels; (iii) technological requirements; and (iv) size of the area with unacceptable interference. Four types of constraints can be specified.

co-cell separation constraint. The frequencies assigned to the same antenna v must differ by at least $\delta(v, v)$ units (typically $\delta(v, v) = 3$).

co-site separation constraint. If u and v are co-site antennae, then typically $\delta(u, v) = 2$.

interference constraint. Due to interference, additional separations can be required between pairs of antennae not at the same site. Typically, such pairs u and v should have different frequencies, i.e., $\delta(u, v) = 1$, or frequencies at distance at least 2.

Constraints that forbid two cells to use the same frequency are often called *co-channel constraints*. Constraints that forbid frequencies with distance 1, usually including distance 0, are called *adjacent channel constraints*.

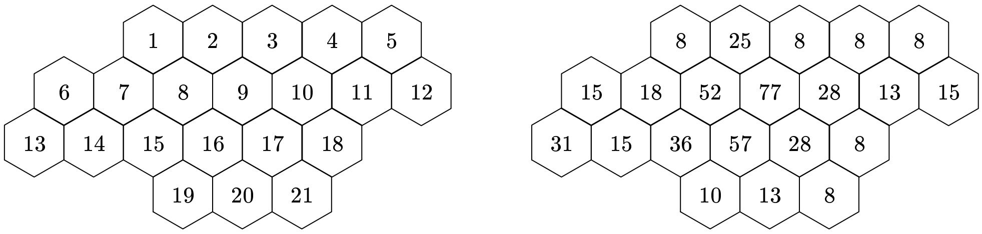
hand-over separation constraint. As the mobile unit moves from a cell u to an adjacent one v, control must be switched from u to v (*hand-over* or *hand-off*), which in turn requires that the broadcasting frequencies used by u and v to serve the mobile, differ by at least one unit. Note that the actual situation, in for instance GSM networks, is more complicated, since the control channels (BCCH) need more protection. This is in some countries translated into a desired distance of 2.

There are several sets of instances available from the literature. The most used sets are the following.

Philadelphia The Philadelphia instances were among the first discussed in the literature [8]. The Philadelphia instances are characterized by 21 hexagons denoting the cells of a cellular phone network around Philadelphia, see Figure 4. Until recently, it was common practice to model wireless phone networks as hexagonal cell systems. Each cell needs a high number of frequencies, the “multiplicity” of the cell. An overview of the results can be found in Appendix A.1 (see also [48] for the most recent results).

COST 259 In the context of the COST (COoperation européenne dans le domaine de la recherche Scientifique et Technique) 259 project (financially supported by the European Union), 32 instances for GSM network planning have been made available. The number of antennae that have to be assigned frequencies ranges from 900 up to almost 4000. Up to 75 frequencies are available at each vertex. The instances are available at [48] together with an overview of the results. A summary of these results can be found in Appendix A.3. More information on the project can be found in the final report [33].

CSELT The CSELT instances have been used by Fischetti et al. [53] and by Mannino and Sassano [94]. These instances have co-channel constraints and adjacent channel constraints. Besides these constraints, multiple interference of antennae is bounded from above by a threshold value L.



(a) network structure

(b) frequency demand of instance `P1`

Figure 4: Hexagonal grid used in the Philadelphia instances

CNET Instances from the French National Research Center for Telecommunications (CNET) have been used in papers by Hao et al. [40–42, 62, 63]. The number of cells is at most 300. To each vertex, one or two frequencies have to be assigned.

CNET 2 Another instance that was also made available by CNET [27] deals with GSM frequency planning. The instance contains raw data about locations of antennae and propagation of the signals.

Bell Mobility Instances provided by Bell Mobility for two Canadian urban areas are made available at [14]. The problems differ in size from almost 700 to more than 5000 transmitters. The instances are used by Jaumard et al. [74–76].

Besides these "realistic" instances, Castelino et al. [30] discussed 6 computer generated instances that have constraints with comparatively high frequency distances among neighboring antennae, and are fairly large with respect to the number of antennae. For every antenna, 50 frequencies are available.

Most of the above mentioned sets of instances consider frequency domains that do not depend on the vertices. The domains are usually represented by one or two sets of consecutive integers. Depending on the objective, the size of this set may or may not vary.

Radio and Television These applications essentially resemble the mobile phone instances. The major difference lies in the used frequency distances. Instances provided by a major Italian radio broadcasting company were made available at [69]. Results for these instances are presented by Mannino and Sassano [94].

There is one set of instances available for UHF TV broadcasting in which the constraints forbid certain differences in frequencies which are not consecutive. For instances, frequency distances $1, 2, 5$, and 14 are forbidden. The practical cause is the frequency band itself, which includes higher harmonics of the frequencies.

Military applications The usage of field phones (or air phones) in the military leads to dynamic (in time and place) frequency assignment problems. These problems have the property that each connection consists of two movable phones. To each connection we must therefore assign two frequencies at a fixed distance of each other, one for each direction of communication. Thus, all frequencies are given in pairs with this fixed distance between them.

In the context of the EUCLID CALMA (Combinatorial ALgorithms for Military Applications) project [26], eleven static real-life instances were provided by CELAR (Centre d'ELectronique de l'ARmement, France), whereas a second set of 14 artificial instances was made available by a group at Delft University of Technology. These GRAPH (Generating Radio Link Frequency Assignment Problems Heuristically) instances were randomly generated by Van Benthem [15], and have the same characteristics as the CELAR instances. There are instances in the CALMA project available that vary over the complete range of models as discussed later. A description of the results achieved in the CALMA project can be found in [2] or [25]. In Appendix A.2, the results are summarized (for updated information see [48]).

The instances of the ROADEF Challenge 2001 [114] are also made available by CELAR and can be viewed as a follow-up of the CALMA project. The frequency assignment problem is extended with polarization constraints. For every connection, a polarization direction (horizontal or vertical) is to be chosen. The interference depends not only on the assigned frequencies but also on the choices for polarization.

Satellite communication In Thuve [125], a frequency planning problem in satellite communication is discussed. In this application, both the transmitters and receivers are ground terminals. They communicate with each other with the help of one or more satellites. Each signal is first transmitted via an uplink to the satellite and next transmitted by the satellite via a downlink to the receiving terminal. The uplink and downlink frequency are separated by a fixed distance, much larger than the bandwidth, which implies that we only have to assign frequencies to the uplink. A set of consecutive frequencies has to be assigned to every transmitter. To avoid interference, every frequency may be used only once. Due to the nature of these constraints the problem does not really fit in the classification presented in the next section.

3 Formulations and Classification

The basic frequency assignment problem consists of assignment constraints, interference constraints (usually packing constraints), and an objective. In this section, we first formulate the basic constraints. In the successive subsections we classify the problem variants, mainly by way of distinct objectives.

The frequency assignment models of Section 2 generally have a predefined set of channels or frequencies, denoted by F. For every antenna or connection v, a subset $F(v) \subseteq F$ of available frequencies is specified, from which a subset of $m(v)$ frequencies must be assigned to v. Generally, the *multiplicity* is equal to one. Higher multiplicities arise in mobile telephony applications, where an antenna represents a cell that may contain multiple TRXs. Each such TRX should be assigned one frequency. Thus, an antenna is to be assigned as many frequencies as there are TRXs. From a modeling perspective it is easy to reduce the multiplicities to one, by viewing each TRX in an antenna as a separate unit. However, the size of a problem instance may then increase quite substantially, and solution methods may not be able to use the special structure, induced by the TRXs, anymore. Note that TRXs in one antenna have high interference restrictions.

Sometimes the frequencies assigned to an antenna is the union of sets belonging to a family of predefined subsets of the available band. These subsets are called *blocks* and a frequency assignment is obtained by assigning to each antenna one or more of such blocks. Such a solution is called a *block assignment.*

A convenient representation of interference is by means of a graph $G = (V, E)$, the *interference graph* or *constraint graph.* Each antenna is represented by a vertex $v \in V$. Two vertices v and w for which the corresponding signals may interfere for at least one pair of transmitting frequencies, are connected by an edge $\{v, w\} \in E$. Multiple frequencies to be assigned to single antennas can be represented by splitting the antenna vertices into a number of copies equal to the desired multiple. Clearly, this may blow up the size of the interference graph, and therefore in some methods we prefer to work with multiplicities on the antenna nodes directly. This extended graph is referred to as the *split interference graph.* Note that loop edges in this graph model the distance requirements of TRXs on the same antenna. In Figure 5, an example of an interference graph from the CALMA project [26] is given.

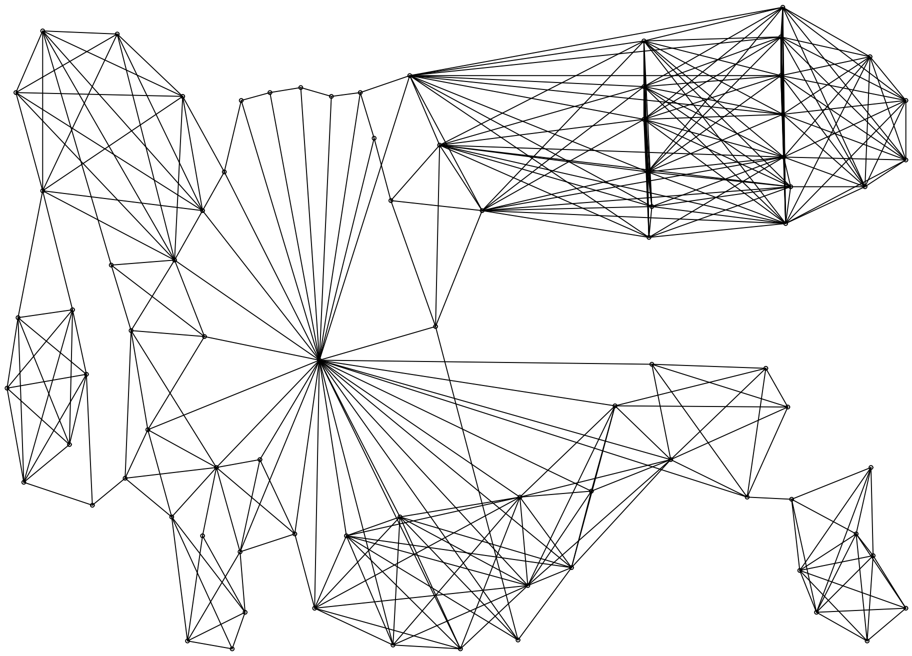


Figure 5: Interference graph of CALMA instance `CELAR [06]`

For each pair of frequencies $f \in F(v)$ and $g \in F(w)$ we penalize the combined choice by a measure depending on the interference level. This penalty is denoted by $p_{vw}(f,g)$ or p_{vwfg}. In most models this penalty has a very specific structure: it depends only on v and w and the distance between the frequencies $|f-g|$. FAPs with this structure are called *distance FAPs* henceforth. Two variants occur frequently in the literature. In the first variant a distance d_{vw} is introduced such that the penalty p_{vw} is incurred if the choices of f and g are such that $|f-g| < d_{vw}$. Note that by selecting very high penalties p_{vw} the distance requirements can turn into hard constraints. In the second variant only a co-channel penalty p_0 (if $|f-g| = 0$) and an adjacent channel penalty p_1 (if $|f-g| = 1$) are incurred, where $p_0 > p_1 \geq 0$.

In many studies the penalty matrices are not used in all detail, but a certain threshold value $p_{\max}$ of interference is allowed. The threshold value corresponds to an acceptable signal-to-noise ratio. This reduces the interference constraints to forbidding certain combinations of frequencies. Moreover, the problem reduces to a binary Constraint Satisfaction Problem (CSP). In case $p_{vw}(f,g)$ only depends on the distance $|f-g|$, this leads, combined with a threshold value, to a set of forbidden distances T_{vw}. This problem is equivalent to the *T-coloring problem*, see Roberts [115], where colors are numbers and certain differences between numbers are forbidden for adjacent vertices. Generally, but not necessarily, the forbidden distances form a set of consecutive integer numbers $\{0, 1, \ldots, d_{vw}-1\}$. In case they do, we will refer to the problem as *distance FAP*.

The mathematical programming formulation of the FAP consists of a set of variables, constraints, and an objective function. A straightforward choice for the variables is to use binary variables representing the choice of a frequency for a certain vertex. For every vertex v and available frequency $f \in F(v)$ we define:

$$x_{vf} = \begin{cases} 1 & \text{if frequency } f \in F(v) \text{ is assigned to vertex } v \in V \\ 0 & \text{otherwise} \end{cases}$$

These variables have been used by the majority of the researchers. They lead to Integer Linear Programming (ILP) formulations that can be solved by Branch-and-Cut methods, for instance. The disadvantage is the size of the variable set. More compact formulations are obtained by using variables f_v for the choice of frequency for vertex $v \in V$. They lead to nonlinear programs which are seldom used explicitly. Moreover, they have the disadvantage that only one frequency can be assigned to a vertex; hence, the split interference graph is essential. Therefore, we will not consider these formulations. On the other hand, even larger formulations used for column generation techniques have been used. They are discussed at the end of this section.

The requirement that $m(v)$ frequencies have to be assigned to a vertex v is modeled by the following constraints, the so-called *multiplicity constraints*:

$$\sum_{f \in F(v)} x_{vf} = m(v) \qquad \forall v \in V \tag{1}$$

The penalty matrices p_{vw} are often used in combination with a threshold value $p_{\max}$. Pairs of frequencies with a penalty exceeding this threshold are forbidden. This is modeled by the following *packing constraints*:

$$x_{vf} + x_{wg} \leq 1 \qquad \forall \{v,w\} \in E, f \in F(v), g \in F(w) : p_{vw}(f,g) > p_{\max} \tag{2}$$

When there is no further objective to be optimized, we obtain the so-called *feasibility* frequency assignment problem (F-FAP). Here, we simply want to find a feasible solution to the FAP, i.e., a solution satisfying the constraints (1) and (2).

In the sequel we consider a variety of objectives for this model. If no feasible solution exists to F-FAP, we can try to assign as many frequencies as possible or minimize the probability that a call will be blocked. Other objectives aim at optimizing operating costs by minimizing the number of frequencies used (until the 70s), or minimizing the bandwidth used (highest minus lowest frequency). All these models use, besides the multiplicity constraints, packing constraints. In case the penalty matrices are used directly, we generally wish to minimize the total penalty incurred. In this model, the packing constraints are replaced by a version that incorporates penalty for certain choices of combinations of frequencies.

3.1 The Maximum Service and Minimum Blocking Frequency Assignment Problems

If feasible solutions to the F-FAP are not available or difficult to find, we can decide to find a partial solution that assigns as many frequencies as possible to the vertices. This problem is known as the Maximum Service FAP, or shortly Max-FAP. Basically, this problem is an F-FAP extended with the objective to assign as many frequencies as possible. To model the problem we introduce the numbers $n(v)$ $(v \in V)$, which denote the number of frequencies assigned to vertex $v \in V$.

$$\max \quad \sum_{v \in V} n(v) \tag{3}$$

$$\text{s.t.} \quad \sum_{f \in F(v)} x_{vf} = n(v) \qquad \forall v \in V \tag{4}$$

$$n(v) \leq m(v) \qquad \forall v \in V \tag{5}$$

$$x_{vf} + x_{wg} \leq 1 \qquad \forall \{v,w\} \in E, f \in F(v), g \in F(w) : p_{vw}(f,g) > p_{\max} \tag{6}$$

$$x_{vf} \in \{0,1\} \qquad \forall v \in V, f \in F(v) \tag{7}$$

$$n(v) \in \mathbb{Z}_+ \qquad\qquad \forall v \in V \tag{8}$$

In contrast to the formulation of F-FAP, the multiplicity constraints (1) need not be satisfied with equality anymore. The objective (3) ensures that as many frequencies as possible are assigned. Jaumard et al. [74, 76] observed that optimal solutions to the Max-FAP tend to assign very few frequencies to some "difficult" vertices, whereas most other vertices obtain all demanded frequencies. Such solutions are not desirable, since this incurs extremely low service in some areas. To cope with this problem, Jaumard et al. introduce a lower bound $l(v)$ on the number of frequencies to be assigned to each of the vertices v to obtain a minimum service guarantee: $l(v) \leq n(v)\ \forall v \in V$. A more realistic way to cope with this problem is to compute the actual blocking probabilities in the vertices as a function of the number of assigned frequencies $n(v)$. This approach has been modeled independently by Mathar and Mattfeldt [95], and Chang and Kim [31], who use a weighted combination of blocking probabilities in the objective function. This problem is known as the Minimum Blocking Frequency Assignment Problem (MB-FAP). Here, we follow the approach taken by Chang and Kim [31]. Let λ_v denote the traffic demand in Erlang for cell v, and $n(v)$ the number of assigned channels. Then the blocking probability of cell v is given by the Erlang B formula as

$$B(\lambda_v, n(v)) = \left(\sum_{k=0}^{n(v)} \frac{(\lambda_v)^k}{k!} \right)^{-1} \frac{(\lambda_v)^{n(v)}}{n(v)!}$$

This function describes the blocking probability for voice traffic with negative exponential distribution of the call inter-arrival time. Note that $B(\lambda_v, n(v))$ is strictly decreasing and convex in $n(v)$. Now, the objective function is a weighted average of the blocking probabilities of all vertices v, given by

$$\sum_{v \in V} w_v B(\lambda_v, n(v)) \tag{9}$$

with $w_v = \lambda_v / \sum_{u \in V} \lambda_u$ being the traffic weighting factor. In contrast to (3), the objective function (9) is to be minimized. Note that the objective of Max-FAP can be viewed as a simplification of this objective: $w_v = 1$, and $B(\lambda_v, n(v))$ is replaced by $m(v) - n(v)$, a linear decreasing function.

Moreover, note that the upper bounds (5) on the number of assigned frequencies are fairly superficial in this model. Their only relevance may come from practical considerations such as a maximum amount of space to install the transmitters. If space is not an issue, by removing the multiplicity constraints one may obtain even better solutions with respect to the objective (9).

3.2 The Minimum Order FAP

If feasible solutions to the F-FAP exist, then we may look for a "cheapest" of the one. The earliest attempt to do so (see Hale [61]), penalizes the usage of frequencies. Thus, the number of different frequencies used should be minimized. This objective dates back to the introduction of mobile telephones in the early seventies, when frequencies were sold per unit and were very expensive. The model is called the minimum order FAP, or shortly MO-FAP.

To formulate the objective, we need extra variables to denote whether a frequency is used or not.

$$y_f = \begin{cases} 1 & \text{if frequency } f \in F \text{ is used} \\ 0 & \text{otherwise} \end{cases}$$

Then, MO-FAP is formulated as follows

$$\min \quad \sum_{f\in F} y_f \tag{10}$$

$$\text{s.t.} \quad x_{vf} \le y_f \qquad \forall v \in V, f \in F(v) \tag{11}$$

$$\sum_{f\in F(v)} x_{vf} = m(v) \qquad \forall v \in V \tag{12}$$

$$x_{vf} + x_{wg} \le 1 \qquad \forall \{v,w\} \in E, f \in F(v), g \in F(w) : p_{vw}(f,g) > p_{\max} \tag{13}$$

$$x_{vf} \in \{0,1\} \qquad \forall v \in V, f \in F(v) \tag{14}$$

$$y_f \in \{0,1\} \qquad \forall f \in F \tag{15}$$

We introduce (11) to force a y-variable to one in case the corresponding frequency is used. The objective (10) determines the number of used frequencies. Note that constraints (11) are node packing constraints: using the complement of the y-variables gives $x_{vf} + \bar{y}_f \le 1$. Note that the *distance* MO-FAP reduces to the standard vertex coloring problem if all distances are equal to 1, and all vertex domains are the same set of consecutive integers (see Cozzens and Roberts [35]).

3.3 The Minimum Span Frequency Assignment Problem

In the Minimum Span Frequency Assignment Problem (MS-FAP), one is supposed to pay for the full set of frequencies between the highest and lowest ones used. Thus, the difference between the maximum and minimum used frequency, the *span*, determines the cost and is therefore to be minimized. To model this problem we introduce two new integer variables, compared to MO-FAP, which denote the largest frequency used $z_{\max}$, and the smallest frequency used $z_{\min}$. The MS-FAP then reads

$$\min \quad z_{\max} - z_{\min} \tag{16}$$

$$\text{s.t.} \quad \sum_{f\in F(v)} x_{vf} = m(v) \qquad \forall v \in V \tag{17}$$

$$x_{vf} + x_{wg} \le 1 \qquad \forall \{v,w\} \in E, f \in F(v), g \in F(w) : p_{vw}(f,g) > p_{\max} \tag{18}$$

$$z_{\max} \ge f y_f \qquad \forall f \in F \tag{19}$$

$$z_{\min} \le f y_f + f_{\max}(1 - y_f) \qquad \forall f \in F \tag{20}$$

$$x_{vf} \le y_f \qquad \forall v \in V, f \in F(v) \tag{21}$$

$$x_{vf} \in \{0,1\} \qquad \forall v \in V, f \in F(v) \tag{22}$$

$$y_f \in \{0,1\} \qquad \forall f \in F \tag{23}$$

$$z_{\min}, z_{\max} \in \mathbb{Z}_+ \tag{24}$$

where $f_{\max} = \max_{f\in D} f$ is the maximum available frequency. The constraints (19) and (20) guarantee that these variables are set to the right values. Note that the second term in the right hand side of (20) is necessary to allow for unused frequencies in F below $z_{\min}$.

An alternative formulation has been presented by Giortzis and Turner [58]. They introduce binary variables instead of $z_{\max}$ and $z_{\min}$. Besides the standard constraints (1) and (2) the new variables introduce additional constraints to set them to the right values.

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

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

With these variables, MS-FAP alternatively reads

$$\begin{aligned}
\min \quad & \sum_{f \in F} f u_f - \sum_{f \in F} f l_f && && (25)\\
\text{s.t.} \quad & \sum_{f \in F(v)} x_{vf} = m(v) && \forall v \in V && (26)\\
& x_{vf} + x_{wg} \leq 1 && \forall \{v,w\} \in E, f \in F(v), g \in F(w) : p_{vw}(f,g) > p_{\max} && (27)\\
& \sum_{f \in F} u_f = 1 && && (28)\\
& \sum_{f \in F} l_f = 1 && && (29)\\
& x_{vf} + u_g \leq 1 && \forall v \in V, f \in F(v), g \in F : f > g && (30)\\
& x_{vf} + l_g \leq 1 && \forall v \in V, f \in F(v), g \in F : f < g && (31)\\
& x_{vf} \in \{0,1\} && \forall v \in V, f \in F(v) && (32)\\
& u_f \in \{0,1\} && \forall f \in F && (33)\\
& l_f \in \{0,1\} && \forall f \in F && (34)
\end{aligned}$$

Constraints (28) and (29) ensure that there is a unique largest and a unique smallest frequency. Constraints (30) forbid to assign frequencies higher than the maximum, whereas (31) forbid to assign frequencies smaller than the minimum.

In Minimum Span FAPs often a set of frequencies $\{1, \ldots, f_{\max}\}$ is available for all vertices, i.e., $F(v) = \{1, \ldots, f_{\max}\}$ $\forall v \in V$. This allows us to set the lower bound to 1, i.e., $z_{\min} = 1$ or $l_f = 1$, in the above MS-FAP model. Thus, minimizing the span is equivalent to minimizing the maximum frequency assigned. In other words, $f_{\max}$ is determined as the minimum frequency for which the MS-FAP has feasible solutions. Doing so with binary search or related techniques F-FAPs or Max-FAPs occur as subproblems.

For the case $F(v) = \{1, \ldots, f_{\max}\}$ yet another formulation is possible. This formulation bases on the formulation of the MO-FAP. Besides the constraints (11)–(15), we introduce the constraints

$$y_{f+1} \leq y_f \qquad \forall f, f+1 \in F \qquad (35)$$

Then minimizing the span is equivalent to

$$\min \quad \textstyle\sum_{f \in F} y_f \qquad (36)$$

This formulation was proposed by Baybars [10] and was probably the first integer linear programming formulation for MS-FAP. It is based on the formulation for graph coloring introduced by Christofides [32].

In case of the *distance* MS-FAP (MS-FAP with hard distance constraints only), a linear ordering of the vertices can be associated with every frequency assignment. In particular, let σ be a linear ordering on the vertices of G. Then the *canonical assignment* of $\sigma = \{v_1, \ldots, v_n\}$ is a feasible frequency assignment with the property that the frequency assigned to v_j is the smallest feasible frequency after $v_1, \ldots, v_{j-1}$ have been assigned. In particular, denoting by $d(v, w)$ the minimum distance required between a frequency assigned to v and a frequency assigned to w, then the canonical assignment $f_1, \ldots, f_n$ corresponding to the linear ordering $\sigma = \{v_1, \ldots, v_n\}$ can be obtained by means of the following recursive equations:

$$f_1 = 1 \tag{37}$$

$$f_k = \min\{f \geq 1 : |f - f_r| \geq \delta(v_r, v_k), r = 1, \ldots, k-1\}, \qquad k = 2, \ldots, n. \tag{38}$$

Note that among all linear orderings, there is one for which the canonical assignment provides the optimal solution to the MS-FAP, since each assignment gives rise to an ordering of the vertices analogous to the ordering of the assigned frequencies.

The relation between MO-FAP and MS-FAP is fairly intimate as follows directly from their formulations. Essentially, the two models only differ by their objective function. In fact, the models coincide under the right circumstances. If we restrict our instances to having co-channel interference constraints only, both problems become standard (list-)coloring problems (see [52, 132]). However, in general a minimum span optimal solution for a problem does not necessarily use a minimum number of frequencies, and vice versa, see Hale [61] or Eissenblätter et al. [47] for examples.

3.4 The Minimum Interference Frequency Assignment Problem

So far, all models simplified the interference data from the penalty matrices, by using them to forbid certain choices of pairs of frequencies. A way to use the penalty data completely is to introduce an objective that minimizes the sum of the penalties incurred by the frequency choices. This is done in the so-called Minimum Interference Frequency Assignment Problem, MI-FAP.

$$\min \sum_{\{v,w\} \in E} \sum_{f \in F(v), g \in F(w)} p_{vwfg} x_{vf} x_{wg} \tag{39}$$

In some instances from the CALMA project, cf. [26], this objective is extended by penalties for the choices of certain frequencies f for v, denoted by q_{vf}. This leads to an extra term in the objective $\sum_{v \in V, f \in F(v)} q_{vf} x_{vf}$, which will be ignored in this section.

Note that the objective contains the quadratic terms $x_{vf} x_{wg}$, resulting in a standard (non-convex) quadratic formulation, cf. Padberg [105], and Warners et al. [135]. To linearize these terms, we define the variables $z_{vwfg} = x_{vf} x_{wg}$, i.e.,

$$z_{vwfg} = \begin{cases} 1 & \text{if } x_{vf} = x_{wg} = 1 \\ 0 & \text{otherwise} \end{cases}$$

To ensure that z_{vwfg} obtains the right value we add the following constraints to the formulation.

$$x_{vf} + x_{wg} \leq 1 + z_{vwfg} \qquad \forall \{v, w\} \in E, f \in F(v), g \in F(w) \tag{40}$$

and if necessary

$$z_{vwfg} \leq x_{vf} \ , \ x_{wg} \qquad \forall \{v, w\} \in E, f \in F(v), g \in F(w) \tag{41}$$

The constraints (41) are usually enforced by (40) and the objective function: if $p_{vwfg} > 0$, then z_{vwfg} is minimized to $\max(0, x_{vf} + x_{wg} - 1)$. The problem with this linearization is that its LP-relaxation is weak. For fractional x-variables, the corresponding z-variable can be small. Fortunately, by using the multiplicity constraints we can replace (and strengthen) the inequalities (40) and (41) with

$$\sum_{g \in F(w)} z_{vwfg} = m(w)x_{vf} \qquad \forall \{v, w\} \in E \ \forall f \in F(v) \tag{42}$$

These constraints are valid by the definition of z_{vwfg} and the multiplicity constraints (1):

$$\sum_{g \in F(w)} z_{vwfg} = \sum_{g \in F(w)} x_{wg}x_{vf} = \left(\sum_{g \in F(w)} x_{wg} \right) x_{vf} = m(w)x_{v,f}$$

On the other hand they imply the definition of the z-variables, i.e., $z_{vwfg} = x_{wg}x_{vf}$. If $x_{vf} = 0$ or $x_{wg} = 0$, then the corresponding variable z_{vwfg} is also equal to 0. Now suppose $x_{vf} = 1$. For w, there exist $m(w)$ frequencies g with $x_{wg} = 1$. Therefore, to satisfy (42) all corresponding z_{vwfg} should be 1.

3.5 Additional Features

There are many more features that can be added to the models presented here, but perhaps the most valuable issue from a practical point of view is the handling of interference caused by multiple sources. The version that we treat here originates from Fischetti et al. [53] and is also used by Mannino and Sassano [94].

The idea is to introduce a local threshold for the interference induced on a vertex v by its neighbors, for each frequency $f \in F(v)$. If the noise produced by all neighboring vertices $N(v) = \{w : \{v, w\} \in E\}$ on a frequency f for v is to be taken into account, we can do so by introducing the following constraints where L_{vf} is an upper bound on the penalty for v if frequency f is chosen:

$$\sum_{w \in N(v)} \sum_{g \in F(w)} p_{vwfg}x_{wg}x_{vf} \leq L_{vf}x_{vf} \qquad \forall v \in V, \forall f \in F(v) \tag{43}$$

We can linearize this constraint by use of an upper bound on the possible interference for any vertex and any frequency, say M.

$$\sum_{w \in N(v)} \sum_{g \in F(w)} p_{vwfg}x_{wg} \leq L_{vf} + M(1 - x_{vf}) \qquad \forall v \in V, \forall f \in F(v) \tag{44}$$

Within the CSELT instances of Fischetti et al. [53] and Mannino and Sassano [94] only co-channel and adjacent channel interference is penalized. Co-channel interference is penalized with I_{vf}, and adjacent channel interference is penalized with $\frac{I_{vf}}{NFD}$, where NFD is a reduction factor called the Net Filter Discriminator. The upper bound on allowable interference, L, is fixed for each vertex frequency pair (v, f).

$$\sum_{w \in N(v)} I_{vf}x_{wf} + \frac{I_{vf}}{NFD}(x_{w,f-1} + x_{w,f+1}) \leq L + M(1 - x_{vf}) \qquad \forall v \in V, \forall f \in F(v) \tag{45}$$

If $f-1$ or $f+1$ do not exist, e.g., in case f is on the border of the spectrum, then the corresponding x-variables should be removed. The constraints (45) are difficult to handle in most optimization methods. Therefore, some authors use other ways to take multiple interference into account. For instance, Dunkin et al. [44] introduce besides forbidden combinations for pairs, forbidden combinations of frequencies for triples of vertices.

3.6 Formulation Variants

Alternative mathematical programming formulations have been proposed in the literature. These formulations are first and for all used to specify the problem at hand. Typically, heuristics inspired by the formulation are explored to solve the problem. In this section, we discuss two such formulations: a column generation one, and the orientation formulation.

3.6.1 Column Generation

Jaumard et al. [76] develop two column generation formulations for the Max-FAP. They consider co-cell distance constraints, and co-channel and adjacent channel constraints. All vertices have the same frequency domains. As mentioned earlier, besides upper bounds, additional lower bounds on the frequency demand (multiplicity) of cells are added to the formulation.

The first formulation is based on a column generation formulation for coloring, such as given in Mehrotra and Trick [98]. The variables correspond to independent sets in the interference graph, i.e., to vertices that may obtain the same frequency simultaneously. For each frequency f these independent sets are denoted by T_f. Note that these sets may differ per frequency, since the frequency domains for the vertices may differ. Let z_t be a binary variable denoting whether or not $t \in T_f$ is chosen. To model the constraints and the objective of the Max-FAP with these variables, we use the relation $x_{vf} = \sum_{t \in T_f : v \in t} z_t$. To ensure that frequency f is chosen at most once we add $\sum_{t \in T_f} z_t \leq 1$. Note that this formulation can also be used for the MO- and MS-FAP: for MO-FAP the latter constraints become $\sum_{t \in T_f} z_t = y_f$. Jaumard et al. [76] solve the LP-relaxation of this formulation with column generation techniques (the pricing problems are weighted independent set problems), and they describe branching strategies as well as cut generation schemes. The authors use their method as a heuristic.

The second formulation is based on admissible sets of frequencies for separate cells. The variables correspond to sets of frequencies that can be assigned to a certain cell. For each cell v, subsets of F denoted by T_v that satisfy the co-cell constraints and lower and upper bounds on the multiplicity are given. Another binary variable z_t specifies whether or not T_v is chosen. The authors show that the LP-relaxation of the formulation based on these variables is, at best, equal to the value of the LP-relaxation of the previous formulation. On the other hand, the pricing problems to be solved in a column generation approach are simple constrained shortest path problems.

3.6.2 Orientation formulation

Borndörfer et al. [20] consider MI-FAPs with co-channel and adjacent channel interference. They model the interference with penalties on combinations of frequencies. Moreover, they forbid combinations of frequencies with penalties above a certain threshold. Among the feasible assignments they seek one with minimum penalty. For each vertex in the interference graph they introduce a variable y_v that corresponds to the frequency number assigned to v. For each pair (v, w) denote the co-channel penalty by p_{vw} and the adjacent-channel penalty by q_{vw}. Now, three more binary variables are introduced:

$$z_{vw} = \begin{cases} 1 & \text{if } |y_v - y_w| = 0 \\ 0 & \text{otherwise} \end{cases} \qquad \overline{z}_{vw} = \begin{cases} 1 & \text{if } |y_v - y_w| = 1 \\ 0 & \text{otherwise} \end{cases} \qquad \Delta_{vw} = \begin{cases} 1 & \text{if } y_v \geq y_w \\ 0 & \text{otherwise} \end{cases}$$

The variables Δ_{vw} determine a partial ordering of the frequencies assigned to the vertices. With these variables one can model all constraints and the objective linearly. The model defined in this way contains much fewer variables, than the formulations given in the previous section. The price for this is a weaker formulation. If the Δ_{vw} are given, the authors show that the problem is solvable in polynomial time, since the constraint matrix is totally unimodular. This result is used in a two-stage heuristic, where the variables Δ_{vw} are adjusted iteratively and then a solution is determined for the new values.

4 Methods for Optimization and Lower Bounding

Since all models for FAP share an important part of their structure (assignment of frequencies and handling of interference), many optimization ideas translate easily from one model to another. This is specifically true for the most extensively used method: tree search. We, therefore, treat this method in a general fashion. We explain the handling of the components of tree search, like branching and subproblem processing, using the F-FAP as the descriptive generic problem type, or Max-FAP if an objective function is needed. We do this for the two versions of tree search, one based on the linear programming relaxation of F-FAP, and one based on combinatorial ideas. Note that these versions latter are used for determining lower bounds on the objective function. The objective function, however, is exactly what the models differ in. Therefore, we treat the (combinatorial) lower bounding techniques separately for each of the models.

The exception to the above is the MI-FAP. Here interference is modeled by using penalties. This makes the MI-FAP much harder to solve than the other variants. This is probably the reason behind a relatively rich set of solution methods for the problem. These methods are therefore treated in a separate subsection.

4.1 F-FAP

In tree search algorithms we distinguish two parts:

(i) Construction of the tree. The variable (or function) choice for branching. The selection of a subproblem from a list L of active subproblems: such as depth-first search, best-first search.

(ii) The processing of a node (or a subproblem) from the tree. This includes instance reduction techniques, and node pruning techniques such as cutting plane algorithms, and combinatorial lower bounding techniques.

The first part of the process structures the tree corresponding to the search algorithm. This part if fairly problem independent. Thus, the ideas used in any of the variants of the FAP can generally be applied directly to other variants. The second part is concerned with actually solving (sub)problems. This part partially depends on the problem at hand, the type of instances, and also on the used technique. The generic ideas with respect to instance reduction, cutting planes, and lower bounds are treated here. The problem specific ideas are treated separately in later subsections.

The F-FAP that we consider in the sequel is, purely for explanatory reasons, restricted to satisfy the following conditions. The frequency domains are equal for all vertices, and consist of a consecutive set of integers $\{1, 2, \ldots, f_{max}\}$, where f_{max} is a given parameter (F-FAP, Max-FAP, MB-FAP, MO-FAP), or a variable to be minimized (MS-FAP). The interference constraints are of the type $|f(v)-f(w)| \geq \delta(v, w)$, where $f(v)$ and $f(w)$ are frequencies assigned to v and w respectively. These restrictions are the ones that are most frequently encountered in the literature. So, most techniques are developed for problems with these characteristics. Moreover, the ideas described in the sequel often allow for straightforward generalization to other characteristics.

4.1.1 Branching rules

The standard branching rule used in combinatorial optimization is to divide the domain of a variable into two (or more) disjoint subsets. For binary variables this rule reduces to setting the variable to either zero or one. In frequency assignment, this implies that a vertex and a frequency have to be chosen. Most branching rules are only occupied with selecting a vertex. The majority of them is based on the relation between FAP and graph coloring ($\delta(v, w) = 1$ for all $\{v, w\} \in E$) and on Constraint Logic Programming (CLP). Vertex selection is done either statically or dynamically. A selection mechanism is a *static ordering* if the ordering is independent of the actual tree search. Such an ordering can be computed at the start. A popular one is the *highest degree first* ordering, which orders the vertices according to their degree (including multiplicities) in the constraint graph. A related ordering that is applied frequently is to iteratively select the highest degree vertex, simultaneously removing it from G. This ordering can also be applied backward, selecting and removing the smallest degree vertices, i.e., *smallest degree latest* ordering.

The CLP approach of Kolen et al. [84] and the branch-and-cut approach of Aardal et al. [1] solve the MO-FAPs from the CALMA project. Both consider the smallest degree latest ordering as the most successful one. Kolen et al. [84] also specify the choice of a new frequency as the one with the highest distance to the already chosen frequencies. Though static, the above orderings all aim at isolating the possibly present hard part of an instance. Mannino and Sassano [94] carry the idea of selecting the difficult part of the interference graph a little further by identifying a hard subgraph (called the *core* in [94]) for the Max-FAP instances of CSELT. After solving the partial problem restricted to the core they hope that the remainder can be solved without influencing the objective function.

Dynamic orderings depend on the subproblem at hand. A simple example of dynamic ordering is *saturation degree vertex* selection. It is attributed to Brelaz [24] who described the idea for graph coloring problems. During the tree search process the number of available frequencies for the vertices decreases due to previously made choices. Assignment of frequencies to a vertex v generally becomes harder if this number is smaller. Brelaz' rule therefore selects the vertex with a minimum number of frequencies available. Clearly, the multiplicity of v, and the multiplicity of its neighbors, also influences the level of difficulty of assigning frequencies to v. Moreover, the distances play a role. The higher the distances the more combinations are forbidden. Giortzis and Turner [58], who consider the Max-FAP and the MS-FAP, devised a branching rule that uses the latter two observations. They dynamically select vertices v for which $m(v) \cdot \sum_{w \in N(v)} m(w)\delta(v, w)$ is maximum. Many variants of these ideas are, of course, possible. An overview of many ordering ideas can be found in Hurley et al. [68].

The choice of variable on which to branch in LP-based methods is fairly standard. One can take variables that have values closest to 0.5 or closest to 0 or 1. In Fischetti et al. [53] branching is done with three such rules used randomly: 1: variables with value in the interval [0.4, 0.6], where the actual choice is determined by the largest degree (number of interfering cells) of the corresponding vertex; 2: variables with value smaller than 0.05, where again the actual choice is determined by the largest number of interfering cells; 3: variables with value closest to 0.5 are chosen. Thus, the standard branching rules for binary variables are mixed randomly. In Aardal et al. [1] standard LP-based branching is combined with a partial ordering of the variables: the frequency variables y_f are considered first. Strangely enough, none of the studies on LP-based methods for FAP uses constraints for branching, although SOS constraints make up a significant part of the formulation.

4.1.2 Subproblem choices

The standard strategies for subproblem selection are depth-first search (DFS) and best-first search (BFS). In applying DFS one attempts to find good solutions quickly. DFS involves little implementation overhead, since the stored part of the search tree resembles a path. Among others, Giortzis and Turner [58], and Kolen et al. [84] use DFS. If a lower bounding method is available, one may select a subproblem with a small lower bound to be processed first, anticipating better solutions to be available, and quicker increase of the overall lower bound. Implementations of BFS are found in Aardal et al. [1] and Fischetti et al. [53]. Mannino and Sassano [94] incorporate a backtracking idea from CLP in their tree search, called back-jumping, which attempts moving back multiple levels at once in the search tree, once an inconsistency is found that that can be traced back.

4.1.3 Reduction techniques

Instance reduction techniques attempt to remove frequencies from the domains of vertices or even complete vertices. The ideas to do so are based on similar ideas in CLP (arc-consistency) and coloring. Consider, for instance a vertex v with neighbors $N(v)$. In the process of assigning frequencies to the neighbors of v, a certain number of frequencies from v will be blocked. If the maximum number of blocked frequencies still leaves enough space (free frequencies) to assign all necessary frequencies to v, then we can remove v from the constraint graph. For example, in the standard instances we consider here, for each vertex $w \in N(v)$ a frequency chosen for w can only block $2\delta(v,w)-1$ frequencies. Thus, in total at most $\sum_{w\in N(v)} m(w)(2\delta(v,w)-1)$ frequencies can become unavailable for usage by v. If the number of remaining available frequencies for v is at least $m(v)\cdot(\delta(v,v)-1)+1$, we can always select enough frequencies for v. This idea is applied dynamically to the MO-FAP in [1] and [84], and to the MS-FAP in [94].

One way that remains to remove frequencies from domains is by consistency checking. In its simplest form we check whether, for a particular frequency $f \in F(v)$, there is a feasible choice of frequencies for the neighbors of v. If not, we can remove f from $F(v)$. This idea is used in [84] and [94].

4.1.4 Cutting planes

Techniques using the LP-relaxation of the formulation from F-FAP generally strengthen the relaxation by using additional constraints, so-called *valid inequalities*. The inequalities that are used are typically derived from the relaxation of FAP obtained by considering the packing constraints (2). These constraints can be illustrated by a graph $H = (W, F)$ of the binary variables x_{vf} known as the *conflict graph*. For each variable we introduce a node (v, f). Two nodes (v, f) and (w, g) are connected by an edge if at most one of the variables may obtain value 1. Now, consider a clique (a complete subgraph) in H with vertex set S. Then clearly, no two of the variables of S may have value 1, and therefore $\sum_{(v,f)\in S} x_{vf} \le 1$ is valid. In general, the most powerful such constraints come from maximal cliques, e.g., cliques that cannot be extended with other vertices. Finding such cliques in H is usually a tremendous task due to the size of H, and therefore most researchers resort to finding certain cliques in H, that are easier to find. Such cliques can, for instance, be found by considering cliques in the interference graph. Consider a clique $\bar{V} \subset V$ in the interference graph G and let $\delta = \min_{v,w\in\bar{V}} \delta(v,w)$. Then the following valid inequality can be formulated:

$$\sum_{v\in S} \sum_{f\in\{k+1,\dots,k+\delta\}} x_{vf} \le 1 \tag{46}$$

where $k = 0, \dots, f_{max} - \delta$. Rouskas et al. [116] consider the MB-FAP with co-channel constraints. Their formulation includes all clique constraints (46) from the start. Fischetti et al. [53] consider a subclass of the clique constraints (46), with $\delta = 1, 2, 3$, for the Max-FAP. They add them to the formulation with a separation algorithm, in a Branch-and-cut framework.

Aardal et al. [1] consider cliques from the conflict graph that can be viewed as lifted versions of (46). Consider, for example, the clique in Figure 6. This clique induces the following valid inequality

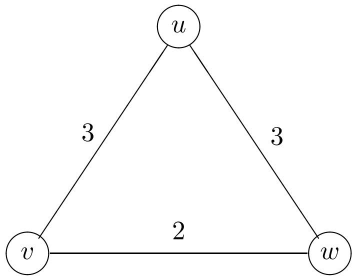


Figure 6: Example 3-clique with $\delta = 2$

(u, v, w are nodes and f, g, h are frequencies).

$$x_{uf} + x_{ug} + x_{uh} + x_{uf} + x_{vg} + x_{vh} + x_{wf} + x_{wg} \leq 1$$

Finally, in Kazantzakis et al. [80] the linear programming relaxation of the Max-FAP is tightened by using cuts derived from rounding the objective function during the (complete) tree search.

4.2 Max-FAP and MB-FAP

4.2.1 Cutting planes

The Max-FAP of Fischetti et al. [53] includes multiple interference constraints (45). These constraints allow for generation of cutting planes based on knapsack covers (see [102]), which are used in their branch-and-cut scheme as well.

4.3 MO-FAP

4.3.1 Instance reduction

The CALMA instances contain nonadjacent pairs of vertices v, w, such that for any neighbor u of v we have $\delta(uv) \leq \delta(uw)$. Moreover $F(v) \subseteq F(w)$. Then the choice made for w is also available for v. Thus, vertices such as v can be removed. Though rare in general, this situation does occur in the CALMA instances.

4.3.2 Valid inequalities

For any clique $\bar{V}$ in the constraint graph G, the following inequality is valid with respect to MO-FAP.

$$\sum_{v \in S} x_{vf} \leq y_f$$

These are used in [1].

4.3.3 Lower bounds

Clearly, the clique and coloring number of the constraint graph are lower bounds of the MO-FAP. If the domains (available frequencies) differ among the vertices, sometimes a list coloring bound may improve upon such bounds. This occurs in some of the CALMA instances. For an overview of these bounds see [2].

4.4 MS-FAP

4.4.1 Valid inequalities

The variables introduced in the model of Giortzis and Turner [58] for MS-FAP give rise to special cliques:

$$\sum_{f \in F(v): f \leq g} x_{vf} + \sum_{f \in F(v): f > g} l_f \leq 1 \quad \forall v \in V, g \in F(v)$$

$$\sum_{f \in F(v): f \geq g} x_{vf} + \sum_{f \in F(v): f < g} u_f \leq 1 \quad \forall v \in V, g \in F(v)$$

The use of such cliques has not been reported on, so far.

4.4.2 Lower bounds

The fairly direct relation between MS-FAP and MO-FAP allows some lower bounding techniques to be used for both models. This applies for instance to the simplest lower bound: the clique bound. Each subgraph of G induced by $W \subset V$ that forms a clique determines a lower bound $|W|$ for MO-FAP and $|W| - 1$ for MS-FAP. This bound, though applicable to MS-FAP, is especially suitable for MO-FAPs. There are, however, more general and more powerful lower bounds available for MS-FAP. The standard clique bound can be generalized as was first observed by Gamst [55]. Let all multiplicities of the vertices be equal to one. If a clique of size k in the interference graph contains edges with minimum distance d only, then the range of frequencies must be at least $(k-1)d + 1$. Over the years, lower bounds for more and more complex structures have been derived (cf. [101] for an overview). Recently, Janssen and Wentzell [73] showed that many of these bounds can be derived within a general theoretical framework called *tile covers*.

The clique bound has been further generalized by Raychaudhuri [112]. They consider a subgraph of the (splitted) interference graph. For an assignment, the vertices can be ordered such that the assigned frequencies form a non-decreasing sequence. If we extend the subgraph to a complete graph by the introduction of edges with distance zero, this order forms a path with length less than or equal to the span of the assignment. Hence, the minimum Hamiltonian path in an arbitrary subgraph (completed by zero-value edges) provides a lower bound on the minimum span for the MS-FAP defined on that subgraph, and thus, on the minimum span for the MS-FAP defined on the whole graph. Note that this bound indeed generalizes the clique bound of Gamst [55]: in a clique of size k with minimum distance d the shortest Hamiltonian path has length $(k-1)d$. Note that the Hamiltonian path bound can also be shown to be a lower bound by use of the canonical assignment, generated by an optimal solution. The Hamiltonian path bound is obtained by using the recursion (38), where we relax the minimization by taking the distance to the last vertex in the ordering.

Jansen and Kilakos [72] compute a lower bound of the Hamiltonian path bound, by considering a limited, but carefully chosen set of subgraphs. For each subgraph they use the following procedure. First, they reformulate the problem into a minimum Hamiltonian cycle problem (TSP). Then, they solve the LP-relaxation of the TSP. Their procedure is powerful enough to prove optimality of some of the Philadelphia problems. Tcha et al. [124] solve the 2-matching relaxation of the TSP to obtain lower bounds.

The above procedure has one serious drawback, namely, that it is difficult to select the right subgraphs. Note that considering the whole constraint graph may give a very short Hamiltonian path due to the existence of many short edges.

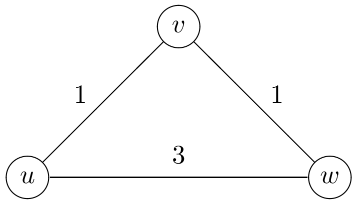


Figure 7: Example bad walk

For example, if edges $\{u, v\}$, and $\{v, w\}$ have distance 1, and $\{u, w\}$ has distance 3, the path (u, v, w) leads to a bound of 2 (its length), whereas the span is equal 3. We will refer to such paths as *bad paths*. The edge $\{u, w\}$ is not used in the bound.

Allen et al. [7] add *excess* variables e_{vw} to the edges, and force these variables to positive values when appropriate, by adding the constraints

$$[\delta(uw) - (\delta(uv) + \delta(v, w))](x_{uv} + x_{vw} - 1) \leq e_{uv} + e_{vw} \tag{47}$$

for all paths (u, v, w) of length 2.

The excess variables e_{uv} and e_{vw} cause the objective function to increase. The authors develop inequalities for larger paths that in principle would allow for exact solution of the MS-FAP. However, these have not been used in their approach. They proceed by solving their problem using Lagrangean relaxation, where constraints (47) are relaxed and added , with a multiplier, to the objective function.

The above bounds have only been used in a stand-alone fashion, i.e., to compute a single lower bound on problem instances. It should be said, though, that the lower bounds are, in general, very close to the optimal span.

Recently, Avenali et al. [9] have devised an optimization algorithm using techniques that are comparable to the Allen et al. [7] method. Here, path variables are used to forbid certain concatenations of paths. A bad path, like for example (u, v, w) in the example above, is avoided by forbidding the concatenation of edge $\{u, v\}$ with edge $\{v, w\}$; when necessary, a new binary variable is introduced to represent the forbidden path P and its weight is set equal to the span of P. This idea is the basis of a column generation approach.

The MS-FAP has initiated a lot of research on the T-coloring problem, where a prespecified set of distances is forbidden between frequencies of neighboring vertices. Roberts and Cozzens [115] develop a theory on lower bounds for special graphs using T-coloring arguments. An overview of the most important lower bounds is given in Murphey et al. [101]. These lower bounds, however, are hardly used in practice since MS-FAP with specific T-coloring type interference constraints are rare.

4.5 The MI-FAP

The MI-FAP model is much more difficult than the previously mentioned variants of the FAP. This is due to the fact that hard interference constraints are turned into soft constraints by the use of penalties. This hardness has caused a large diversity in solution methods. For instance, there are only two papers, to our knowledge, that use some sort of tree search. Other methods are based on dynamic programming, the structure of the interference graph, and (in case of lower bounding) combinatorial relaxations.

The earliest attempt to solve the MI-FAP is from Verfailly et al. [131] who developed a procedure called the Russian doll algorithm. This algorithm is perhaps best described as a backward tree search in combination with lower bounds. For a certain static ordering of the vertices of the interference graph, say from 1 to n, we consider n iterations. In a backward fashion, in each iteration all assignments of vertices $\{k+1,\ldots,n\}$ are considered. Lower bounds and upper bounds on the penalties are computed for all subsets $\{l,\ldots,n\}$ $(l > k)$ of vertices, which are used in subsequent iterations. Thus, in the iteration of vertex k we do a complete tree search with the vertices $\{k,\ldots,n\}$, using the produced lower bounds for the subsets $\{l,\ldots,n\}$ $(l > k)$. Although the paper says little about the choice of the ordering of the vertices, it probably uses rules similar to the branching rules of subsection 4.1.1, such as "smallest degree last". The Russian-doll procedure has been used to solve `CELAR 06` (an instance from CALMA) to optimality.

Koster et al. [86] combine tree search with the linear programming relaxation of the MI-FAP. They solve the problem, formulated as a Partial Constraint Satisfaction Problem (PCSP), with branch-and-cut, using standard branching rules and variable selection mechanisms, and valid inequalities based on the boolean quadric polytope (cf. Padberg [105]). The valid inequalities are derived from structures in the interference graph, such as cycles and cliques. In a cycle C (or clique), the set $F(v)$ for $v \in C$ is partitioned into two sets A_v and B_v. For cycles the following inequalities are valid.

$$\begin{aligned}\sum_{i=1}^{k-1} \big(z(v_i, A_{v_i}, v_{i+1}, A_{v_{i+1}}) + z(v_i, B_{v_i}, v_{i+1}, B_{v_{i+1}})\big)& \\ +z(v_0, A_{v_0}, v_k, B_{v_k}) + z(v_0, B_{v_0}, v_k, A_{v_k}) \qquad &\leq k-1\end{aligned} \tag{48}$$

where $z(v, A_v, w, A_w) = \sum_{f \in A_v} \sum_{g \in A_w} z_{vwfg}$. Figure 8 shows a 3-cycle inequality and a 4-cycle

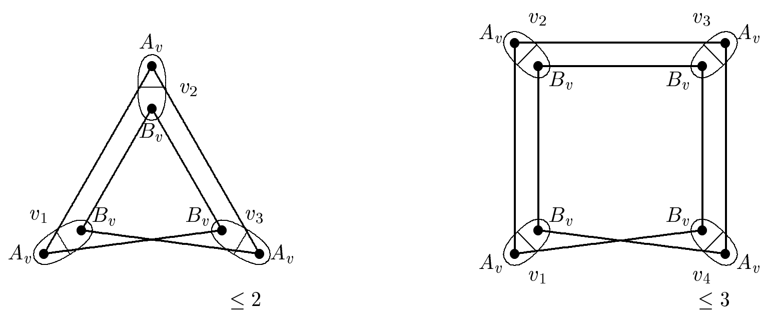


Figure 8: Cycle Inequalities

inequality. A line between two dots indicates that the coefficient corresponding to the indicated subsets is equal to one.

For cliques, we take a coefficient $1 \leq \gamma \leq k-1$, where k is the size of the clique, and we get the following (γ, k)-clique inequalities.

$$\gamma \sum_{v \in C} x(v, A_v) + \sum_{\{v,w\} \in E[C]} z(v, B_v, w, B_w) \geq \gamma k - \tfrac{1}{2}\gamma(\gamma+1) \tag{49}$$

where $x(v, A_v) = \sum_{f \in A_v} x_{vf}$. See Figure 9 for examples of 3-clique and 4-inequalities.

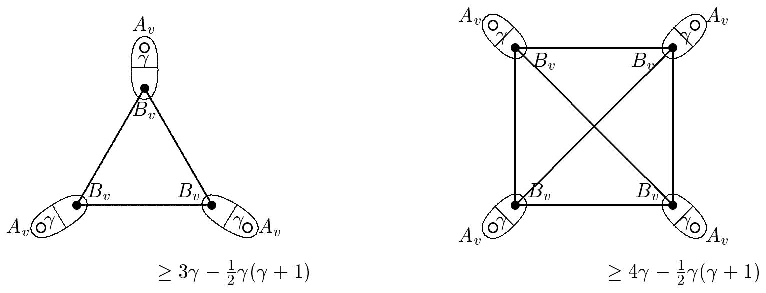


Figure 9: (γ, k)-Clique-Cycle Inequalities

The Branch-and-Cut method using these inequalities solves the problem well for instances with domain sizes up to 6 frequencies, especially with dominance criteria and reduction methods incorporated. It has been used as a subroutine (with domain sizes 2) in a genetic algorithm by Kolen [83]. For larger domain sizes the method returns fairly poor lower bounds.

In Koster et al. [87] make use of the structure of the interference graph $G = (V, E)$. They observe that assigning frequencies to a cut-set of G decomposes the problem into two (or more) independent problems. They generate a sequence of small cut-sets by using a tree decomposition (see Bodlaender [17]) of the interference graph. Note that a small cut-set induces a relatively small number of assignments to the vertices of the cut-set.

A series of reduction methods were developed to limit the number of assignments. These ideas led to the solution of some quite large MI-FAP from the CALMA project. For some remaining instances Koster et al. [87] improved the known lower bounds by introducing a relaxation where the vertex domains are partitioned in a small number of subsets. Each such subset is then treated as a single frequency. By considering a sequence of relaxations, better and better lower bounds could be derived for the CALMA instances. The relaxations are solved with the above described tree decomposition approach. Alternatively, the cutting plane algorithm of [86] can be applied (cf. [88]).

4.5.1 Lower bounds for MI-FAP

Lower bounding for MI-FAP has started with the work of Hurkens et al. [126] who use a quadratic programming relaxation that can be solved by clever enumeration. Non-trivial bounds are reported on a limited set of CALMA instances, namely those where next to the interference penalties, also single frequency penalties are used to favor certain frequencies for vertices.

Another lower bound is derived by Eisenblätter [46] using semidefinite programming. He studies the semidefinite programming relaxation of the minimum k-partition problem. MI-FAP reduces to a min k-partition problem in case only the co-channel interference is considered. Like the related max cut problem, the min k-partition problem can be modeled as a semidefinite program. The relaxation of this semidefinite program can be solved in polynomial time. For GSM networks of the COST 259 project (cf. Section 2.2), the first lower bounds were computed in this way.

In Maniezzo and Montemanni [92], the MI-FAP formulation forms the starting-point for deriving lower bounds. These bounds are inspired by similar bounds for the quadratic assignment problem. In addition, reduction and dominance rules are presented. A tree search algorithm is based on these bounds and rules. The algorithm is tested on the CALMA and Philadelphia instances (taking a fixed spectrum) as well as on some graph coloring benchmarks.

In Montemanni et al. [100], a refinement of the orientation formulation (cf. Section 3.6.2) by Koster [85] is used. Valid inequalities are derived that bound the interference in subgraphs from below. A cutting plane algorithm is exploited on realistic GSM network instances and some Philadelphia instances (by limiting the spectrum).

5 Heuristic Methods

Due to the difficulty of the diverse FAPs, the majority of research papers have been on heuristic approaches. In this section, we discuss these approaches in the form of an annotated bibliography. This section is organized as follows. We start in Section 5.1 with constructive algorithms that build a solution in a greedy manner. In the subsequent subsections we consider local search methods (cf. Section 5.2), that start with a given solution and, with iteratively doing small changes (moves), try to find good solutions. Standard local search methods only allow improving moves (*downhill*). To increase the chances for improving solutions also worsening moves *uphill* may be allowed. Tabu search (cf. Section 5.3) allows worsening moves under certain conditions (neighborhood restriction with tabu list) and simulated annealing (cf. Section 5.4) allows worsening with a probability that typically depends on the size of the worsening, and generally decreasing in time. Genetic algorithms (cf. Section 5.5) start with a whole set of solutions, called *generation*, and iteratively builds new generations by recombination of solutions from the previous generation. Artificial Neural Networks (cf. Section 5.6) generate new solutions by emulating the behavior of a grid of neurons, where each neuron represents a "piece" of solution and its state is dynamically determined by the states of its neighboring neurons. Ant Colony Optimization (cf. Section 5.7) is a meta-heuristic that is inspired by the behavior of ants.

Finally, the section is closed with application specific heuristics. In Section 5.8, heuristics based on mathematical programming formulations are discussed, whereas heuristics based on graph theory and constraint programming are the topic of Section 5.9. Throughout the section, we assume that the reader is familiar with the classical meta-heuristic algorithms mentioned above. Still, in the beginning of each subsection we briefly recall the main components of the described class. The various implementations of a specific scheme mainly differ in the way these components are handled.

5.1 Greedy Algorithms

A greedy algorithm constructs a frequency assignment by iteratively selecting a vertex (an antenna), and then assigning a feasible frequency to it. The selection and assignment follow some rule based on local characteristics that has the aim to optimize the global objective function. An important feature of this (greedy) algorithm is the irrevocable nature of the greedy choice which is performed at each iteration. Many versions of the greedy algorithm have been proposed in the literature to solve FAP, often in conjunction with more sophisticated local search methods. Several greedy heuristics are described and compared in the early work of Zoellner and Beall [138]. The model is MS-FAP. First, an ordering of the vertices of the interference graph is performed: then the frequencies are assigned to the vertices following this ordering. In particular, three different orders are considered:

(i) *highest degree first*: the vertices are ordered by non-increasing degree;

(ii) *smallest degree last*: the vertices are ordered so that the degree of v_j in the graph induced by the set $\{v_1, v_2, \ldots, v_j\}$ is (one of) the smallest;

(iii) *random order*.

Two types of assignments are compared: *frequency exhaustive*, which corresponds to the canonical assignment, and *uniform* where the current vertex is assigned the least used available and feasible frequency.

Sivarajan et al. [119] propose a slight modification of the above defined highest degree first order, taking into account multiple demands and distance requirements. In particular, the *degree* of a vertex $v \in V$ is defined as the quantity $\sum_{u=1,\ldots,|V|} m(v)\delta(u,v) - \delta(v,v)$. Frequencies are again assigned by canonical assignment. Tests were performed on the Philadelphia instances.

Adjakplé and Jaumard [3] make use of block assignments (cf. Section 3) in order to solve distance MI-FAP. At each iteration, the greedy weight of a vertex is defined as a function of the unsatisfied demand, of the size of the feasible blocks, and of the number of forbidden channels (due to several types of interference constraints). Once a vertex is selected, it is assigned the feasible block which maximizes the marginal variation, with respect to the cardinality of the block, of the interference level.

Generalized Saturation Degree. This methodology generalizes the well known DSATUR procedure for graph coloring (see Brelaz [24]) to FAPs. The basic definition of *saturation degree* of a vertex v is simply the number of *blocked* frequencies, i.e., the number of frequencies in the available band which cannot be assigned to v (in consequence of the violation of some hard constraints (2)). At each iteration, the greedy choice consists of selecting the vertex with largest saturation degree and assigning to it the smallest non-blocked frequency. This is the scheme adopted in Costa [34] (where the model is MO-FAP). A slight modification is presented by Borndörfer et al. [19]: here the model is MI-FAP; after a vertex v has been selected in the standard way, it is assigned the non-blocked frequency which minimizes cost increase. Carlsson and Grindal [29] reinforce the basic scheme by adding several mechanisms borrowed from constraint programming, such as propagation, lifting, intelligent backtracking, redundancy avoidance and iterative deepening. The model is MI-FAP and frequencies are assigned in blocks rather than singularly. Finally, a *Generalized Saturation Degree* (GSD) is defined in Valenzuela et al. [130]. If frequency f is blocked for vertex v, then the *weight* of f is the largest penalty cost $p_{vw}(f,g)$ for all $w \in V$, g assigned to w. The GSD of v is the sum of the weights of its blocked frequencies.

In Zhang and Yum [137] the model is MB-FAP. The vertices are clustered according to their geographical distance. In particular, each cluster is *compact* in the sense that the average distance between all pairs of vertices in the cluster is minimum. There is only a limited number of such clusters. At each iteration of the algorithm a new frequency is assigned to a cluster where the cluster is chosen so as to minimize the blocking probability.

Sequential Packings

The algorithm presented in Sung and Wong [123] deals with (distance) MS-FAP. All frequency domains are equal to $\mathbb{Z}_+$. Two heuristic procedures are proposed. The first considers only $\delta(v,w) \leq 1$, for all $v, w \in V$. It finds a family of stable sets $S_1, \ldots, S_q$ such that each vertex v is contained in exactly $m(v)$ stable sets in the family. All vertices in stable set S_i are assigned frequency i. The stable sets are built in sequence, i.e., the construction of S_{i+1} begins after S_i is completed. The first vertex in the current stable set S_i is the one with largest unsatisfied demand. Next, a vertex w which maximizes $|N(S_i) \cap N(w)|$ is selected. Ties are broken by the weight of the maximum weight clique (where the weights on the vertices are equal to the residual demands) in the set $N(S_i) \cap N(w)$. The authors are able to prove that for a 3-stripe cellular system (a particular hexagonal network) the assignment produced by their procedure is optimal.

The second heuristic tackles the case where $\delta(v,w) = 2$ for some $v, w \in V$. As in the previous case, it finds a family of stable sets $S_1, \ldots, S_q$ such that each vertex v is contained in exactly $m(v)$ stable sets in the family and all vertices in stable set S_i are assigned frequency i. However, now stable sets are built in pairs, i.e., S_i and S_{i+1} are found simultaneously. The choice of the vertex to add to the current pair of stable sets is made according to criteria analogous to the previous case.

5.2 Local Search (LS)

Local Search is certainly the most basic improving heuristic developed for combinatorial problems. We refer here to the classical definition as described, for example, in [106]. According to this definition one starts with a given solution and replaces it with a better one (*improving solution*) selected from a restricted subset of the solution set. If no improving solution exists in the restricted subset, then LS stops. Otherwise, the whole process is iterated with the old solution replaced by the improving one. The restricted solution subset depends on the current solution and is defined as the set of solutions (*neighborhood*) that can be obtained from the current solution by a predefined set of small changes (*moves*). Observe that a crucial issue in LS, as well as in other neighborhood-based methods, is the ability of efficiently optimizing over the neighborhood space. So, on the one hand we would like the neighborhood to be large, so as to increase the chances to find improving solutions, but on the other hand, large neighborhoods correspond in general to exponentially increasing search times. In FAP efficiency is reached by enumerating over small cardinality neighborhoods. Observe that a frequency assignment is a partition of the vertex set, where each class corresponds to a frequency in the available spectrum (additional empty classes take into account still unassigned frequencies). The *1-exchange* neighborhood is the set of solutions which may be obtained from the current one by selecting a vertex and moving it into a different class. Such an operation is called *1-exchange move.* If n is the number of vertices and f_{max} is the number of frequencies, then the number of different solutions in the 1-exchange neighborhood is simply $n(f_{max}-1)$. The *2-exchange* neighborhood is the set of solutions which may be obtained from the current one by selecting two vertices and swapping their frequencies.

Even though very few algorithms can be classified as "pure local search", still LS is often used as a building block, such as in the paper of Castelino et al. [30] devoted to the solution of distance MI-FAP with unit penalties. An initial solution is generated at random. Then a sequence of moves is performed (*passes*). Each pass consists of $|V|$ iterations. At the i-th iteration, the i-th vertex is selected and an improving solution is searched in the 1-exchange neighborhood restricted to v_i. This process is interrupted if a 0-cost solution is generated or a fixed number of passes has been performed.

Another good example of an application of pure LS is the algorithm by Park and Lee [107] for MI-FAP. Here, both the 1-exchange and the 2-exchange neighborhoods are used and applied to a set of randomly generated instances.

Guided Local Search

Guided Local Search is a meta-heuristic technique proposed by Tsang and Voudouris [128] that helps Local Search to escape local optima. First of all a number of *features* of a solution has to be defined. Each feature is a mapping $I(s)$ from the solution set S to the set $\{0,1\}$. With the j-th feature I_j we associate a cost c_j and a penalty p_j. Penalties are updated during the search. Finally, the fitness function (to be minimized) is given by the sum of two terms, $g(s) + \lambda \sum_j p_j I_j(s)$. The first term $g(s)$ is typically the original objective function, while the second term is proportional to the sum of the penalties associated with the features exhibited by solution s. At each step we choose the neighbor minimizing the fitness function. When we are trapped in a local minimum s_*, some of the penalties are increased. In particular we increase by one unit those penalties whose associated feature is exhibited by s_* (in practice only a subset of such penalties are upgraded, based on the cost of the associated feature and on the current penalty value).

In [128] this technique is applied to MI-FAP, MO-FAP and MS-FAP. The neighborhood is defined by a move that consists of changing frequencies to a pair of *coupled* vertices in all possible ways. Note that in some problems the frequencies assigned to particular pairs of coupled vertices must differ (exactly) by a specified quantity. The underlying local search algorithm is the so-called *Fast*

Local Search (FLS), which is simply a restricted neighborhood search. Basically, once a move is performed, the pair of vertices involved in the move is declared tabu. The tabu status of a vertex can be reset when specific conditions are verified (all involving the fact that an adjacent pair has been processed in the last move). For the Guided Local Search, three sets of features are considered. Each feature in the first set is associated with a constraint of type (2): the cost of each feature is the corresponding (interference) cost in the objective function of MI-FAP. A second set is associated with mobility costs. When solving MO-FAP, a third set of features is associated with constraints (15). Benchmark instances are the CALMA instances.

Canonical assignments

Here we describe local search algorithms based on neighborhood structures related to orderings of the vertices. The corresponding frequency plan is determined by the corresponding canonical assignment (i.e., according to the ordering, assign the smallest available frequency, cf. Section 3.3).

The basis of the heuristic method presented in Wang and Rushforth [134] to solve MS-FAP is the canonical assignment associated with the linear orderings of the split graph introduced in Section 3.3. The neighborhood of a solution is the set of all solutions obtained by swapping the positions of two distinct vertices. A neighborhood restriction is obtained by fixing the first vertex in the pair: such a vertex is randomly selected among the set of vertices with largest assigned frequency. The second vertex is randomly selected. In a first phase, a new solution is accepted if and only if the associated span is strictly smaller than the previous one. In a second phase, executed when a fixed maximum number of non-decreasing tentative moves has been examined, the acceptance criterion is relaxed by accepting non-increasing solutions. Finally, the initial solution is obtained by ordering the vertices by non-increasing weighted degrees. When tackling large instances, the authors propose the following decomposition scheme: find a vertex coloring of G, identify all vertices belonging to the same color class to obtain a new graph G', solve the frequency assignment problem for G' and then extend the solution to G.

In Rushforth and Wang [117] the above algorithm is enhanced by partitioning the network into the *minimum network* and the *difference network*. Initially, the original network is partitioned into a number of k-cell-clusters. Each cluster contains k vertices, labeled from 1 to k. All vertices in different clusters labeled by the same integer can be assigned the same frequency. Now, for each cluster, we can define a *cluster frequency demand* as the minimum demand of all vertices in the cluster. The *minimum network* is the original network with modified frequency demands: in particular, the demand of vertex i is the demand of the cluster containing i. The *difference network* is the original network where the demand of vertex i is the difference between the demand of vertex i in the original network and the demand of vertex i in the minimum network. First an assignment is found for the minimum network. Then, by considering a sufficiently large "guard" interval between the largest frequency assigned to the minimum network and the smallest frequency available for the difference network, a feasible assignment is also found for the difference network.

The paper by Box [23] makes use of the same strategy: solutions are found by associating the canonical assignment to orderings of the vertices of the graph. The aim is to find a feasible assignment using a fixed number of consecutive frequencies. The initial ordering is randomly selected. If this ordering is associated with a feasible canonical assignment, we are done. Otherwise, some of the vertices of the graph cannot be assigned. These vertices receive random weights (increments in the subsequent iterations) belonging to the interval [0.15,0.45], and a new order is found according to these weights. The procedure is repeated until a feasible solution is found (or a stopping criterion is fulfilled). In the final part of the paper, the method is extended to handle additional constraints such as distinct vertex frequency domains, pre-assignments, inter-modulation, etc.

5.3 Tabu Search (TS)

Tabu Search is a local search method that, in contrast with standard LS, allows for non-improving moves. At each iteration the best solution in the neighborhood is selected as the new current solution. Notice that this solution can be worse than the current solution. In order to try to avoid cycling, the solutions selected in the last k iterations are declared *tabu solutions* and cannot be selected again. In fact, checking the tabu status of a solution may require excessive computing time. So, rather than prohibiting solutions, it is in general preferred to avoid inverting any of the last k moves. That is, if vertex v has been moved from class f_1 to class f_2 within the last k iterations, then v cannot be moved back from class f_2 to class f_1. Parameter k is called the *tabu list length.* The algorithm typically stops after a fixed number of non-improving moves has occurred.

In order to classify the different tabu search approaches, we will consider the following basic ingredients:

(i) The way the initial solution is generated.

(ii) Fitness function, i.e., the function to be minimized (maximized), which also determines the best solution in the neighborhood. In general the original objective function of the problem plays this role, but sometimes the objective function is adapted.

(iii) Neighborhood definition. Most of the algorithms presented in the literature adopted the 1-exchange neighborhood. Unless otherwise specified, this is the default neighborhood.

(iv) Neighborhood restriction. Additional mechanisms introduced to reduce the size of the neighborhood.

Other parameters, such as tabu list length or number of iterations to termination will not be discussed here, since they do not really lead to different algorithms.

MS-FAP Costa [34] deals with (distance) MS-FAP. However, MS-FAP is solved by solving a sequence of (distance) F-FAPs, where at each iteration the size of available band is reduced. The fitness function is the interference cost of the current solution (expressed as the sum of distance violations). The initial solution is found by generalized DSATUR (see Section 5.1). The neighborhood is restricted by fixing the maximum number of tentative moves. The vertices with largest *local violation*, the sum of the terms in the objective function involving v, are chosen first. Finally, the test instances are randomly generated.

In Hao and Perrier [64], the model is MS-FAP, which is solved (as in [34]) by solving a sequence of (distance) F-FAPs. The initial solution is generated by applying standard local search to a random assignment. The fitness function is the standard objective of MI-FAP. The 1-exchange neighborhood is restricted by randomly selecting a small subset. Finally, the tabu status of a move depends on how recent and how often it is preformed. Test instances are 45 randomly generated mobile telephony instances provided by French CNET. In Hao et al. [63] the authors enhance the quality of the algorithm presented in [64] by implementing the following improvements: 1) an efficient data structure to quickly compute best moves; 2) the neighborhood is restricted by only considering moves involving vertices with positive local violation; 3) co-cell constraints are treated separately from other type of interference constraints, i.e., only solutions which are feasible with respect to co-cell constraints are considered; 4) the algorithm solves a sequence of problems in order to minimize the span. At each iteration the best solution of the previous iteration is used to initialize the tabu-search rather than producing a new random solution.

MI-FAP In Bouju et al. [21] the model is distance MI-FAP with two alternative definitions of penalties: either unit penalties or penalties proportional to the distance requirement. The initial solution is randomly generated, the fitness function is the number of violated constraints, the neighborhood restriction is performed by selecting the k vertices with largest local violation, where parameter k is increased during the search. Before the tabu search is started, an arc consistency preprocessing is performed to reduce the size of the instances, see Section 4. A subset of the CALMA data set is adopted to test the algorithm.

In Castelino et al. [30] the model is distance MI-FAP with unit penalties. The fitness function is the objective of MI-FAP. No restriction is applied to the neighborhood. The tabu state of a move is determined both by how recent a move has been performed and by the number of times a move has been performed. Test problems are six artificial instances.

Block assignment is proposed in Adjakplé and Jaumard [3] to solve distance MI-FAP, where penalty costs are integers ranging from 0 to a maximum of 10. The initial solution is found by a greedy block assignment, see Section 5.1. The fitness function is the standard objective of MI-FAP. The neighborhood of a block assignment is defined by means of two different moves. The first move consists of changing exactly one block assigned to a vertex, replacing it with one or more available blocks (observe that blocks are not equally sized). The neighborhood is restricted by fixing the maximum number of tentative moves; blocks with largest local violation are chosen first, where the *local violation* of a block B with respect to a given solution is the sum of the terms involving B in the fitness function. In order to diversify the search, a second type of move is periodically performed. That is, a vertex with largest local violation is selected and all of its frequencies are reassigned from scratch by a greedy block assignment heuristic. Test instances are real-life instances provided by the Canadian Bell Mobility [14].

A more sophisticated definition of move is proposed in the work by Borgne [18]. Frequencies are assigned in blocks and besides the standard move consisting of changing exactly one block assigned to a vertex, Borgne proposes an adaption of the so called *Kempe Chains Interchange*. In terms of graph coloring, a *Kempe Chain* is simply a connected component of the subgraph induced by two color classes, say C_1 and C_2 corresponding to colors c_1 and c_2. A *Kempe interchange* consists of interchanging the colors of the vertices in the two color classes, i.e., assigning color c_2 to the vertices in C_1 and color c_1 to the vertices in C_2. The extension to frequency assignment with only co-channel constraints is straightforward. The idea is to first select two adjacent vertices u and v that are assigned a same frequency, say f_1, and that have strong co-channel interference. Select another frequency f_2 and suppose that no vertex assigned to f_2 is adjacent to u. Then we can assign f_1 to all vertices which are assigned to f_2 and assign f_2 to all vertices which are assigned to f_1 but u. In this new assignment, u and v have different frequencies and no other edge violation is created. This leads to a reduction of the overall interference cost. This definition of move can be generalized by taking into account other types of interference constraints such as adjacent channel constraints, and by allowing the interchange of three or more frequencies. Experiments on two real-life cellular network problems provided by Ericsson show the effectiveness of generalized Kempe chain interchanges.

Finally, Capone and Trubian [28] solve MI-FAP by minimizing the interference level directly evaluated on the grid of test points introduced in Subsection 2.1 (rather than on the interference graph). They use the standard neighborhood, i.e., the exchange of a frequency to a single vertex. However, exploring and evaluating all the solutions in the standard neighborhood can be too expensive due to the large number of test points in an average sized instance. So, once a vertex v is chosen, the neighborhood is restricted by first evaluating a simplified objective function which makes it possible to remove some of the frequencies available for v. Ad hoc generated test instances have been used to test the algorithm.

5.4 Simulated Annealing (SA)

Analogous to TS, also Simulated Annealing allows for non-improving moves. However, the anti-cycling strategy consists of a randomized selection mechanism. Specifically, the best solution in the neighborhood is accepted as a new current solution either if it is better then the old one, or with a probability which depends on its value. This probability increases as the difference between the current value and the new value decreases. In addition, the acceptance probability is controlled by another parameter, the *temperature*. This parameter decreases as the number of iterations increases (*cooling*). Ceteris paribus, lower temperatures correspond to lower acceptance probabilities. When the temperature is very low, non-improving solutions will not be accepted anymore and the algorithm terminates (*freezes*).

The main ingredients of SA are:

(i) initial solution,

(ii) neighborhood structure,

(iii) fitness function, and

(iv) cooling strategy.

The last parameter is determined by the initial temperature and the cooling rate.

In most algorithms, the neighborhood is defined by 1-exchange moves. Typically a single vertex is chosen at random, and moved into the least costly alternative frequency class. In some cases, the new frequency is randomly assigned.

The initial temperature is chosen so as to ensure that a given percentage of tentative moves is accepted (recall that the acceptance rate grows with the temperature).

Finally, the temperature decreases only after a specified number of iterations has been performed at constant temperature. In the following we denote by an L*-loop* the inner loop of SA, i.e., a block of iterations performed at constant temperature.

MS-FAP In Costa [34] the model is (distance) MS-FAP. The initial solution is found by generalized DSATUR (see subsection 5.1). The fitness function is the sum of the distance violations. The vertex is chosen among those involved in some positive local violations. The number of iterations of the L-loop is increased at each temperature update. The cooling rate is linear. Test instances are randomly generated.

MI-FAP In Duque-Antón et al. [45] the model is distance MI-FAP. A dummy frequency is introduced to represent (partially) unsatisfied demand. Substituting a dummy frequency with an available one or vice versa (*single flip*) corresponds to increasing or decreasing the violation of traffic demand. Both the vertex as well its new frequency are randomly chosen. In order to increase the performance of the algorithm the new frequency is chosen now and then as the most frequently assigned to close (with respect to their actual geographical location) non-interfering vertices. Another technique used to extend the neighborhood consists of performing a sequence of moves before the acceptance test. The cooling rate is chosen so that the difference between the average solution cost of two consecutive L-loops at temperature t_1 and t_2 is no more than the standard deviation of the solution costs at temperature t_1. The system is frozen when the current solution does not change during the last L-loop. Ad hoc test instances are proposed.

In Knälmann et al. [82, 110] the model is distance MI-FAP. The move is performed by randomly selecting one vertex and by randomly changing its frequency. No hard constraints are considered. Experiments are performed on a 10 transmitter FM network from the German broadcaster Südwestfunk.

In Beckmann and Killat [11] neighborhood restriction is obtained by selecting one vertex v at random and replacing the frequency assigned to v causing the largest amount of interference by a frequency in the domain of v causing the smallest amount of interference. An initial solution satisfying all hard constraints is obtained by applying the genetic algorithm presented by the same authors in [12] (described in Section 5.5). The algorithm was applied to large real-life instances of the COST 259 test-bed.

An interesting variant of SA is presented in Zerovnik [136]. The algorithm is inspired by the graph coloring algorithm of Petford and Welsh [109]. The model is distance MI-FAP with all penalties being unit penalties. To handle multiple demands the split interference graph is used. An initial solution is found by a uniform random assignment. The main difference with a standard SA approach is that the initial temperature T is never changed. At each iteration a vertex v involved in a large number of violated constraints is selected. Then a new frequency f is assigned with probability $e^{-S_f/T}$, where S_f amounts to the number of constraints that will be violated by assigning f to v. Benchmark instances are 7-cluster hexagonal torus [45] and triangular lattice graphs with random demand.

A second variant of SA, called *threshold accepting* is applied by Heller and Hellebrandt [65]. The minimal differences between this technique and SA are not discussed here. The initial solution is found so as to fulfill all of the hard constraints. The initial temperature is chosen such that the acceptance rate is between 0.8 and 0.9. The move is the standard one, i.e., exchange of a frequency for a single vertex, but the neighborhood is restricted by only considering moves that do not violate hard constraints. An important feature of the algorithm is *one-cell optimization*, which is performed at the end of every L-loop. The authors show that, by a simple dynamic program, it is possible to efficiently optimize over the neighborhood of the current solution obtained by letting *all* of the frequencies assigned to a vertex be changed simultaneously. In fact, it is possible to show that this corresponds to looking for a minimum cost k-cardinality stable set in interval graphs, where k is the demand of the vertex. Benchmark instances are taken from the COST 259 test-bed. The authors propose to extend this idea to clusters of vertices, but they are not able to describe efficient search algorithms - here they propose to use a greedy search. Finally, Mannino et al. [93] describe a generalization to cliques of vertices of the dynamic programming approach presented in [65]. In particular, they show that finding an optimum assignment for a clique of vertices with multiple demands can be reduced to finding fixed cardinality stable sets in a generalization of interval graphs, and this task can be performed in polynomial time.

MB-FAP In Mathar and Mattfeldt [95] the model is (distance) MB-FAP with only hard co-channel constraints. The authors propose a non standard definition of a neighborhood in order to guarantee convergency of the method. A solution is represented by m distinct orderings of the vertices, where m is the number of available frequencies. A frequency assignment is then generated as follows. If $\pi = \pi_1, \ldots, \pi_m$ is a solution (where π_i is the ordering of the vertices associated with the i-th frequency) then the corresponding assignment is obtained by assigning frequency i to vertex v if and only if none of the vertices adjacent to v precedes v in π_i. The move is defined by suitable permutations of the current solution. Namely, it consists of randomly selecting an available frequency r, and a permutation Φ from a set Ψ of pre-defined feasible permutations. Ψ must be a generator of the set of all permutations of the set $\{1, \ldots, |V|\}$. The move simply consists in applying permutation Φ to π_r. Three different choices for Ψ are discussed and compared with the standard definition of move.

A hybrid model (MI-FAP + MB-FAP) is addressed in Al-Khaled [4]. The main novelty with respect to standard SA consists of an adaptive cooling rate. In particular the cooling rate depends on the difference between the average of the accepted solution values of two consecutive L-loops. Ad hoc hexagonal test instances are considered.

5.5 Genetic Algorithm (GA)

Genetic Algorithms are inspired by the natural process of reproduction. Metaphors as *chromosomes* and *population* stand for solutions and solution set, respectively. Analogously, a single variable is often indicated as a *gene*. Mechanisms as recombination and mutation give rise to new offspring by manipulating the current population of solutions. Specifically, mutation applies to a single solution (chromosome) while crossover creates new solutions from a pair of solutions selected in the current population. Following a standard Darwinistic approach, selection extracts the most promising individuals in the current population.

The main features of a genetic algorithm are the following:

(i) Chromosomal representation. The correspondence between chromosomes and solutions.

(ii) Initial population. An initial set of solutions (chromosomes).

(iii) Fitness function. The function used to evaluate the quality of candidate chromosomes.

(iv) Selection. A mechanism to select promising chromosomes (in conjunction with fitness function).

(v) Crossover and mutation. Mechanisms to generate new solutions from the currently selected chromosomes.

The most common way to represent a solution is as follows: each chromosome is a vector $s \in \mathbb{Z}^{|V|}$, where s_j is simply the frequency assigned to v_j. The split graph model is adopted when multiple demands are considered. We denote this representation by ($R1$). In a second rather common representation ($R2$) each chromosome is a partition of the vertices in a family of f_{max} (eventually empty) subsets $S_1, S_2, \ldots, S_{f_{max}}$, called the *genes*, where S_f is the set of vertices that are assigned frequency f for $f = 1, \ldots, f_{max}$, and f_{max} is the maximum available frequency. Simple adaptations are required when not all frequencies in the interval $[1, f_{f_{max}}]$ are available. Often such a chromosome is represented by a binary string of $f_{max} \times |V|$ elements. The set of vertices that is assigned frequency f (called the *f-th gene*) is stored in the bits in the interval $[(f-1) \times |V|+1, f \times |V|]$: specifically, a 1 in position $(f-1) \times f_{max} + k$ means that frequency f is assigned to vertex k. In a third representation ($R3$) each chromosome is a permutation σ of the $|V|$ vertices, and represents the canonical assignment associated with σ (see subsection 3.3). Again, the split graph model is considered when multiple demands are considered.

Several types of crossover operators have been applied in the literature. As an example, consider the *one point* crossover, applied to (R1). Let $(f_1, \ldots, f_n)$ be the first parent chromosome, and let $(g_1, \ldots, g_n)$ be the second parent chromosome. Now, an index k is chosen and two new children are generated: the first is $(f_1, \ldots, f_k, g_{k+1}, \ldots, g_n)$, while the second is $(g_1, \ldots, g_k, f_{k+1}, \ldots, f_n)$. More sophisticated cross-over operators are the two-point crossover, where each parent is split into three parts, and the uniform cross-over, where the each gene is copied either from the first parent or from the second parent according to a pre-defined scheme.

MS-FAP In Valenzuela et al. [130] the representation is (R3). The initial population is randomly generated. The generalized Saturation Degree greedy algorithm is applied to each element of the initial population to obtain good quality initial solutions. The mutation operator consists of exchanging the position of two vertices. The fitness function is the span of the permutation. Selection is made by round-robin for the first parent, i.e., all chromosomes are selected in turn according to a circular ordering, while the second parent is chosen with probability proportional to its fitness value. Testing is performed on Philadelphia instances.

MO-FAP In Kapsalis et al. [77] the model is a variant of MO-FAP. The fitness function is a weighted sum of (i) the number of distinct frequencies, (ii) the weighted sum of violated constraints (different types of constraints different weights), (iii) mobility costs. Two chromosomal representations are adopted: the first is a simple variant of (R1), the second is (R2). Besides standard operators, a number of specialized crossover operators were tested. The first one, applied to (R1), consists of a repeated application of the following steps:

(i) Select a constraint (and the corresponding pair of vertices u and v) and check whether this constraint is satisfied by any of the two currently selected parents.

(ii) If it is satisfied, then interchange the frequencies assigned to u and v in the first parent with those assigned in the second parent.

The second one is a single-point operator applied to (R2): similar to [38] (described above), the genes are interchanged taking care of the hard constraints. A first mutation operator is applied to (R1) and consists of choosing a pair of vertices and interchanging the frequencies assigned to them. Also the second mutation operator applies to (R1) and consists of choosing a pair of vertices whose assignment violates a hard constraint, then randomly changing the assigned frequencies with a new pair of available frequencies which do not violate such constraint. The last mutation applies to (R2). A set of vertices with the same frequency are reassigned a commonly available new frequency. Finally, several strategies to static and dynamic modification of the fitness function are proposed. Experiments are performed using the CALMA data set.

In Dorne and Hao [40] the model is (distance) MO-FAP and the authors adopt the standard representation (R1). The fitness function is the number of unsatisfied constraints. Mutation in a given chromosome is performed by selecting an infeasible assignment, i.e., a vertex v and a frequency assigned to v violating one or more constraints, and then replacing this frequency with the best alternative. The selection mechanism extracts one chromosome from the current population by favoring elements not yet trapped in local optima. The child obtained by applying the mutation operator is accepted only if (i) its fitness function is not worse than the fitness of the parent or (ii) randomly with a given probability. No crossover is applied. Test instances are a set of 18 real-life problems from CNET. The algorithm favorably compares with Constraint Programming and Simulated Annealing.

Dorne and Hao [41] extend the algorithm presented in [40] to the multi-demand case. Co-cell constraints are used to limit the size of the search space by a suitable adaptation of the mutation operator. Again, crossover is not implemented. The GA algorithm is repeated several times after the generation of a new population. Testing is performed on 10 instances.

Finally, Hao and Dorne [62] extend the algorithm presented in [41] (described above) by applying three different crossovers: single-point, uniform, and *conflict based*. The last consists of changing only those genes representing frequencies violating one or more constraints.

MI-FAP In Cuppini [39] the encoding scheme is (R2), while the fitness function is the weighted sum of two terms: the first takes into account the global interference level, while the second is a measure of the span. Only *asexual crossover* is used to produce new generations: it consists of choosing two genes $G1, G2$ in a chromosome and two crossover points (the same for each gene). Then, two new genes are obtained by breaking the old genes in the crossover point and then by completing the first part of $G1$ with the second part of $G2$ and the first part of $G2$ by the second part of $G1$. A chromosome is chosen with probability proportional to its fitness value. Computational experiments are performed on a number of ad hoc instances.

In Lai and Coghill [90] the chromosomal representation is (R1). Three-point crossover is applied with some specialization required to avoid conflicts in reconstructing the new offspring. Random mutation is applied to each gene of a new chromosome with probability equal to 0.01.

Ngo and Li [103] use the representation (R2). Crossover and mutation are designed to maintain the number of one's in the chromosome unchanged (*Genetic-Fix Algorithm*) so that the number of frequencies assigned to each vertex is unchanged. In particular, two-point crossover is applied by selecting an initial gene g_1 and a final gene g_2 and by swapping only a subset of the genes between g_1 and g_2. The solution is mutated by randomly substituting a frequency assigned to a given vertex with a different one. A compressed encoding scheme which takes co-cell distance constrains implicitly into account is also proposed in order to reduce the search space. To diversify the search, a local search procedure is sporadically applied to the current best solution in order to increase the quality of the solution. Basically, a suitable subset of the most interfered frequencies currently assigned to some vertices is selected and randomly replaced by new frequencies.

In Crisan and Mühlenbein [36] the encoding scheme is (R1). The mutation operator consists of replacing a frequency assigned to one vertex by a randomly chosen frequency from a candidate set. The new frequency must respect co-cell constraints. Crossover is obtained by selecting a *good* vertex v in the first parent, i.e., a vertex with no local violation, and then constructing a new assignment in the following way: (i) for v and for all vertices belonging to the neighborhood of v, the new assignment is equal to the assignment in the first parent; (ii) for all other vertices the assignment is as in the second parent. The rationale behind this choice is that, if we partition the network into two subnetworks G_1 and G_2, two distinct assignments may have opposite performances in terms of fitness function in G_1 and G_2. In this way an attempt is made to get the best out of the two parents. Of course, after crossover, local adjustments are required to minimize the fitness function. Tests were performed on real-life instances.

Beckmann and Killat [12] adopted (R3) as representation. The model is MS-FAP and the span is optimized by iteratively reducing the number of available frequencies. The fitness function takes into account the number of "blocked calls" (vertices that cannot be assigned an available frequency without violating hard constraints) plus an additional tie-breaking term that takes the number of vertices receiving large frequencies into account. In the selection mechanism, a chromosome is chosen with probability proportional to its fitness value. Two different mutation operators are considered: (i) one of the blocked calls is selected and randomly displaced, or (ii) a group of 4 contiguous calls is randomly rearranged - in fact, at each application of this mutation operator, 3 different groups are randomly selected. Uniform crossover produces two children from a pair of parents. First, the vertices are randomly partitioned into two subsets. In one child, the vertices in the first subset obtain the positions they occupy in the first parent, while the other vertices obtain the residual positions according to the ordering induced by the second parent. For the other child, the role of the parents is interchanged. The test bed is a subset of the Philadelphia instances.

In Jaimes-Romero et al. [70], MI-FAP is solved by a standard genetic algorithm blended with a local search heuristic which is used to generate the new population. In particular, after a solution with 0 blocking probability is found by a standard genetic algorithm, a new phase is started where the algorithm tries to minimize the overall interference level (number of violated constraints). Once a chromosome C is selected, a mutation is generated by looking to the best neighbor, where the neighborhood is the 1-exchange restricted to vertices which are endpoints of an edge violated by C. The chromosomal representation is (R2). Testing is performed over ad hoc instances.

An innovative approach to mutation and crossover is presented by Kolen [83] to solve MI-FAP. The fitness function is the cost of the solution. The initial population is generated at random. The mutation operator consists of a 1-opt local search that transforms the input solution into a 1-optimal solution. This operator is applied to every new entry and to every chromosome in the initial population. So, at any stage, every solution in the current population is 1-optimal. The most relevant difference with former genetic approaches is that crossover is an optimal operator, i.e., once the two parents are selected, the best possible combination of their genes is calculated to generate a single child. More precisely, the selection operator is a random operator which selects a parent with a probability inverse proportional to its fitness value. Every vertex in the child solution will then obtain its frequency either from one parent or from the other - so that there are only two possible choices for each vertex. The best solution is then computed by applying a branch-and-cut procedure based on the partial constrained formulation of MI-FAP, see Section 3.4.

In Crompton et al. [38] the two different representations (R1) and (R2) along with corresponding crossover and mutation operations are compared and applied to a hybrid model. The single point crossover operation, applied to (R2), produces children C^1, C^2 from parents P^1, P^2 by first selecting a so-called cross frequency x. Then $C^1_f = P^1_f$ and $C^2_f = P^2_f$ for $1 \leq f \leq x$. The remaining classes are interchanged with some care, since simply letting $C^2_f = P^1_f$ and $C^1_f = P^2_f$ for $f > x$ would result in infeasible solutions (a feasible chromosome is a partition of the vertices). Two point and uniform crossover are simple extensions. Mutation consists of randomly selecting a vertex and reassigning it to a different class. The fitness function is a weighted sum of four terms, representing the number of violated constraints, the span, the order and the larger violation. This scheme is incorporated into a parallel algorithm, where each node of a parallel computer runs a sequential algorithm with its own population of chromosomes. Occasionally exchanges of chromosomes can take place among various populations. The authors illustrate the superior effectiveness of the (R2) scheme over (R1) on two different instances.

5.6 Artificial Neural Networks (ANN)

Again, the natural learning process is a useful paradigm to define heuristic algorithms for combinatorial problems. In the ANN context, solutions are generated by a network of *neurons*, whose *states* represent the values of the variables involved in the model. In order to minimize an *energy function*, which represents the objective of the problem, the neurons change their states dynamically as a function of the states of the neighboring neurons.

Basic ingredients in an ANN algorithm are the following:

(i) Neuron definition. The mapping between neuron states and solutions.

(ii) Energy function. The objective to be minimized.

(iii) Synapses (coupling weights). Weighted connections between two neurons.

(iv) Local updating rule. Function of the neighbor states and the coupling weights used to update the state of a neuron.

A standard way to define the neural network for FAP is the following: associate a neuron V_{if} with each pair (i, f) where $i \in V$ and $f \in F_v$. Two neurons are coupled if the corresponding vertices are adjacent in the interference graph. The energy function is the weighted sum of several terms, representing different types of interference constraints (co-cell, co-site, etc.), demand constraints (number of required frequencies) and sometimes instance specific requirements.

In Kunz [89] the model is distance MI-FAP. The coupling weight between two distinct neurons V_{if} and V_{jr} depends on the type of interference relation (co-cell, co-site, etc.) between the corresponding two assignments. Computational experiments were performed on random hexagonal networks and a real-life network from the city of Helsinki.

In Funabiki and Takefuji [54] the model is distance MI-FAP. The local updating rule consists of two terms. The first term is proportional to the demand deficiency while the second is proportional to the distance violations. Several heuristics are proposed to increase the ability of escaping from local minima.

The dynamic MI-FAP is considered in Del Re et al. [113]. At each iteration the current assignment is updated to take into account new requests of connection. To speed up the process, only those vertices involved in new connections are re-optimized. The energy function is the weighted sum of several terms to handle interference level, unsatisfied demand, (Hamming) distance from former assignment, (Hamming) distance from predefined reuse schemes, number of distinct frequency assigned. Computational experiments are performed on hexagonal networks with non-uniform traffic distribution. A special type of dynamic channel allocation is the *borrowing channel assignment* (BCA) where one vertex is allowed to borrow frequencies from its neighbors. A standard ANN algorithm to solve (BCA) is presented in Sandalidis et al. [118].

In Kim et al. [81] the neurons can only assume binary values. The energy function takes into account several types of interference constraints and the level of unsatisfied demand. In particular, the latter term is translated into an additional input to each neuron, which forces the assignment of new frequencies to vertices with unsatisfied demand. Several initialization and updating methods are proposed. Benchmark instances are ad hoc hexagonal networks.

Model MI-FAP is addressed in the connectionist algorithm presented in Bouju et al. [22]. The algorithm is compared on the CALMA test-bed.

Finally, Smith and Palaniswami [122] modified the Hopfield network to incorporate some hill climbing mechanism for escaping from local minima. Experiments with a self-organizing neural network are also performed (however, the Hopfield network with hill climbing appears to be the best method). Benchmark instances are the Philadelphia instances and the Helsinki instances proposed in [89].

5.7 Ant Colony Optimization

Ant Colony Optimization (ACO) is a class of constructive meta-heuristic algorithms inspired by real ants behavior. We have a fixed number of ants, where each ant can be interpreted as a sequential greedy algorithm which iteratively generates a solution by upgrading partial solutions *moves*. A move is controlled by two parameters: the *attractiveness* which is based on the structure of the problem (costs and constraints); the *pheromone trail level* which takes into account how many times a given move has been successful. Pheromone trails are updated when all ants have completed the construction of their solution, increasing (decreasing) the level for those moves which led to good (bad) solutions. A lower bound is required to fix the initial level of pheromones. Finally, the solution generated by each ant is possibly improved by local search.

Maniezzo and Carbonaro [91] solve MI-FAP by means of ACO. A partial solution consists of an assignment of frequencies to a subset of the vertices. In order to complete a given partial solution, every ant selects a new vertex and a new frequency to be assigned to this vertex at each iteration. This assignment is defined as a move. The initial lower bound is computed by solving the relaxation of the (orientation) formulation proposed in [20]. The solution produced by each ant is locally upgraded by local search. The authors use this solution to define a partial order on the vertex set (as in the orientation model). The proposed local search algorithm looks for solutions with the same underlying order. Benchmark test problems are the CALMA instances and the Philadelphia instances.

5.8 Formulation based relaxations

A standard way to solve an optimization problem consists of describing it by means of a mathematical formulation which in turn may be solved by some standard techniques. However, for most practical instances such formulations are too large to be solved to optimality and we must content ourselves with heuristic solutions. A classical heuristic approach consists of removing some "difficult" constraints so that the residual problem can be handled. Removed constraints can be accounted for in several ways or simply ignored.

5.8.1 Lagrangian Relaxation

In Chang and Kim [31] the adopted model is MB-FAP, with co-channel and adjacent channel constraints. A family of stable sets (*patterns*) of the interference graph is generated. To simplify the model, frequencies are assigned to patterns, rather than to single vertices. The objective function is linearized by exploiting the fact that the Erlang B formula is piecewise linear. The problem is formulated as a mixed-integer linear program, by means of two types of variables: assignment variables (to select patterns and assign a frequency to each selected pattern), and variables associated with the linearization of the Erlang B formula. The Lagrangian relaxation is obtained by relaxing all the constraints coupling the two types of variables. The residual problem is a maximum weighted stable set problem which is solved by branch-and-bound. The Lagrangian multipliers are updated by a standard subgradient approach. Test problems are randomly generated.

5.8.2 Orientation formulation

The orientation formulation (cf. Section 3.6.2) is exploited in the method proposed by Borndörfer et al. [20] for MI-FAP. The authors show that, under mild assumptions, the relaxation of the orientation formulation has integral solutions once all variables of type Δ_{uv} are fixed either to 0 or 1, i.e., an orientation of the edges has been fixed. This observation leads to a two-stage heuristic. In the first stage an orientation Δ is chosen. In a second stage, a minimum cost assignment is found by solving the associated linear program. A new partial orientation is then generated by exploiting information associated to the solution of the relaxation.

5.8.3 Potential reduction

An approximation algorithm both for MI-FAP and for MO-FAP based on Karmarkar's potential reduction approach to combinatorial optimization is proposed by Warners et al. [135]. First the problem is formulated as a quadratic non-convex optimization problem (see Section 3.4). It is then shown that any feasible (optimal) fractional solution to this formulation can be converted to multiple feasible (optimal) integer solutions of MI(MO)-FAP. Now, solving quadratic non-convex optimization is a difficult task which can be approximately performed by applying potential reduction. To do this, the constraints of the formulation that ensure that each vertex receives exactly one frequency are relaxed and the objective function is modified by adding a weighted logarithmic barrier potential function. Namely, for each constraint i if we denote by s_i its slack variable and by w_i an associated positive weight, the new term has the form $\sum_i w_i \log s_i$. This new non-convex optimization problem is solved by an adaption of the method developed by Karmarkar et al. [78]. Several implementation details of this approach, such as starting point, alternative rounding schemes, techniques to escape local minima are discussed in deep. Also, several preprocessing techniques are used to reduce the size of the instances. Experiments are performed on the CALMA instances.

5.9 Ad hoc

5.9.1 Solve and Extend

The *solve and extend* strategy consists of two phases. In the first phase, a suitably hard subproblem of the original one is selected and solved. The subproblem should be small enough to be handled easily. In the second phase the solution found in this way is extended to a solution to the original problem. The two phases are iterated until a "satisfactory" solution to the whole problem is found or until a stopping criterion is met. The term "satisfactory" gives rise to different interpretations in different papers. In Smith et al. [121] (MS-FAP) the initial subproblem is found by selecting a p-level clique, i.e., a subset C of the vertices of G with the property that $\delta(u, v) \geq p$ for all $u, v \in C$. Such a clique is selected in order to maximize the associated lower bound for MS-FAP, see Section 4.4.2. A solution to $G[C]$ is found heuristically. This solution is then (heuristically) extended to a solution of G. If such a solution is "satisfactory", we are done. Otherwise new vertices are added to C and the method is iterated. The new vertices are chosen according to decreasing saturation degrees, i.e., the number of different frequencies assigned in the neighborhood.

A similar approach is presented in Mannino and Sassano [94] to solve MS-FAP; the solution to the subproblems and the extension are found by an exact implicit enumeration algorithm. A "satisfactory" solution is a solution whose span does not exceed a pre-defined quantity. New vertices are selected and added to the subproblem by using connectivity criteria. Namely, if S is the set of vertices in the subproblem, then the new vertex v maximizes the quantity $\sum_{u \in S} \delta(u, v)$.

5.9.2 Constraint Programming

The models considered by Walser [133] are MS-FAP and MO-FAP. The original interference graph is heuristically shrunk by constructing a suitable covering of the vertices with stable sets. All the vertices in a stable set are identified into a single vertex, which corresponds to assigning the same frequency to all of the original vertices. To compute a feasible assignment of the shrunk graph, the author uses canonical assignments, see Section 3.3. To obtain the solution constraint programming techniques are used. A restricted backtracking technique, called *limited discrepancy search* is exploited, which consists of visiting only those leaves of the branching tree which are not "too far" from the first solution achieved by depth first search.

6 Concluding Remarks

In this survey, we have given an overview of the frequency assignment literature of the last 10–15 years. To conclude the paper, we like to make the following remarks.

The classification given in section 3 shows that *the* frequency assignment problem does not exist. In this survey, we classified the FAPs according to their objective function, given two constraint types: assignment and interference. However, there are more relevant technical and practical issues not accounted for, such as multiple interference and dynamic channel allocation. This makes it very difficult to seek out the relevant literature. The fact that the problem is at the crossroad of multiple disciplines (graph theory, management science and telecommunication technology), does not make it easier either to keep track of new developments. The digest in Appendix B uses this classification to sort the publications, to support the search for relevant literature. This digest is also published at the FAP web-site [48] and will be regularly updated with the newest publications in the field. In this way, it hopefully serves as the first source for all those who are interested, now and in the future.

Another drawback that holds for most optimization problems motivated by practice, and therefore also for FAPs, is the limited availability of benchmark instances. Most of the proposed algorithms are tested on specific, solitary instances, often unavailable to the public, or even on randomly generated instances. Exceptions are the CALMA and COST 259 projects, the papers devoted to the solution of the Philadelphia instances, and the effort made by Caminada [27] to make available a real GSM network instance, with all its technical side constraints. It would be a tremendous improvement when authors test their algorithms not only on their solitary instances, but also on at least one of these sets of benchmarks instances. Alternatively, new/existing data sets can be made available so that other groups can test their algorithms on these instances as well. Also in this context, FAP web likes to serve as an intermediary, and as the place for comparing the results.

Finally, many of the proposed methods, especially the heuristics are sometimes small variations on standard themes. For the sake of completeness, a brief description of most of these technical papers have been reported in the survey. On the other hand, a limited number of original or seminal papers have appeared, and they have been discussed in more detail. We hope that our categorization of these methods gives insight into their contribution, and helps in positioning future research as well.

All above mentioned reasons for confusing the study of the frequency assignment literature, applied to the authors of this survey as well. Over and over again, we discovered relevant (new or old) papers on the topic. Therefore, we do not claim that this survey gives a complete overview of the literature. We, however, have confidence that all important developments in the field have been covered. Although we plan to keep track of future trends in frequency assignment, this intention can only be realized with the help of all working in the field. Also at this point, FAP web plays an important role. By providing information about new papers and results to FAP web, the FAP community can be informed extremely fast, and newcomers are always provided with the latest information in the field.

Altogether, this survey should not be seen as a final report on frequency assignment, but as an evaluation of the field. There is still enough room for improvement of the solution and lower bounding techniques for all variants of frequency assignment problems. The practical relevance of, in particular, the minimum interference and minimum blocking FAPs can be an additional motivation to do so. This survey offers the possibility to detect the areas where advances can be successful.

Acknowledgement

We would like to thank Andreas Eisenblätter for his many helpful comments, both on the form and the contents of this article.

References

[1] K. I. Aardal, A. Hipolito, C. P. M. van Hoesel, and B. Jansen, *A branch-and-cut algorithm for the frequency assignment problem.*, Research Memorandum 96/011, Universiteit Maastricht, 1996, Available at http://www.unimaas.nl/~svhoesel/.

[2] K. I. Aardal, C. A. J. Hurkens, J. K. Lenstra, and S. R. Tiourine, *Algorithms for radio link frequency assignment: The CALMA project*, Operations Research **to appear** (2002).

[3] P. M. Adjakplé and B. Jaumard, *Greedy and tabu search heuristics for channel block assignment in cellular systems*, Tech. Report G-97–45, École Polytechnique de Montréal, July 1997.

[4] F. S. Al-Khaled, *Optimal radio channel assignment through the new binary dynamic simulated annealing algorithm*, International Journal of Communication Systems **11** (1998), 327–336.

[5] J. D. Allen, R. V. Helgason, and J. L. Kennington, *The frequency assignment problem: A solution via nonlinear programming*, Naval Research Logistics **34** (1987), 133–139.

[6] S. M. Allen, S. Hurley, D. H. Smith, and S. U. Thiel, *Using lower bounds in minimum span frequency assignment*, Meta-Heuristics: Advances and Trends in Local Search Paradigms for Optimization (S. Voss et al., ed.), Kluwer, 1999, pp. 191–204.

[7] S. M. Allen, D. H. Smith, and S. Hurley, *Lower bounding techniques for frequency assignment*, Discrete Mathematics **197/198** (1999), 41–52.

[8] L. G. Anderson, *A simulation study of some dynamic channel assignment algorithms in a high capacity mobile telecommunications system*, IEEE Transactions on Communications **21** (1973), 1294–1301.

[9] A. Avenali, C. Mannino, and A. Sassano, *Minimizing the span of d-walks to compute optimum frequency assignments*, Mathematical Programming **91** (2002), no. 2, 357–374, previously published as techincal report 04–00, DIS-Universit di Roma "La Sapienza", Rome, Italy.

[10] I. Baybars, *Optimal assignment of broadcasting frequencies*, European Journal of Operations Research **9** (1982), 257–263.

[11] D. Beckmann and U. Killat, *Frequency planning with respect to interference minimization in cellular radio networks*, Tech. Report TD(99) 032, COST 259, Vienna, Austria, Jan. 1999.

[12] ———, *A new strategy for the application of genetic algorithms to the channel-assignment problem*, IEEE Transactions on Vehicular Technology **48** (1999), no. 4, 1261–1269.

[13] ———, *A powerful hybrid algorithm for the channel-assignment problem basing on evolutionary optimization*, Proc. "3rd European Personal Mobile Communicactions Conference", EPMCC'99 (Paris, France), 1999.

[14] Bell-Mobility, *Frequency assignment benchmarks Canadian urban areas*, URL: http://www.crt.umontreal.ca/~brigitt/telecom/TEST_PROBS/prob1.html, 1998.

[15] H. P. van Benthem, *GRAPH generating radio link frequency assignment problems heuristically*, Master's thesis, Delft University of Technology, 1995.

[16] P. Björklund, P. Värbrand, and D. Yuan, *Optimal gsm network planning with frequency hopping*, Presented at the 5th INFORMS Telecommunication Conference, Boca Raton, USA, March 5–8 2000.

[17] H. L. Bodlaender, *Treewidth: Algorithmic techniques and results*, Lecture Notes in Computer Science **1295** (1997), 29–36, Proceedings 22nd International Symposium on Mathematical Foundations of Computer Science, MFCS'97.

[18] L. Borgne, *Automatic frequency assignment for cellular networks using local search heuristics*, Master's thesis, Uppsala University, 1994.

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

[20] ———, *The orientation model for frequency assignment problems*, Tech. Report TR 98–01, Konrad-Zuse-Zentrum für Informationstechnik Berlin, 1998.

[21] A. Bouju, J. F. Boyce, C. H. D. Dimitropoulos, G. Vom Scheidt, and J. G. Taylor, *Tabu search for the radio links frequency assignment problem*, Applied Decision Technologies (ADT'95) (London), 1995.

[22] A. Bouju, J. F. Boyce, C. H. D. Dimitropoulos, G. Vom Scheidt, J. G. Taylor, A. Likas, G. Papageorgiou, and A. Stafylopatis, *Intellegent search for the radio links frequency assignment problem*, Int. Conf. For Digital Signal Processing (DSP'95) (Limassol, Cypres), 1995.

[23] F. Box, *A heuristic technique for assigning frequencies to mobile radio nets*, IEEE Transactions on Vehicular Technology **27** (1978), 57–74.

[24] D. Brélaz, *New methods to color the vertices of a graph*, Communications of the ACM **22** (1979), 251–256.

[25] B. Cabon, S. De Givry, L. Lobjois, T. Schiex, and J. P. Warners, *Benchmarks problems: Radio link frequency assignment*, Constraints **4** (1999), 79–89.

[26] *EUCLID CALMA project*, Publications and instances available at FTP Site: ftp://ftp.win.tue.nl/pub/techreports/CALMA/, 1995.

[27] A. Caminada, *CNET France Telecom frequency assignment benchmark*, URL: http://www.cs.cf.ac.uk/User/Steve.Hurley/f_bench.htm or http://www.crt.umontreal.ca/~brigitt/telecom/TEST_PROBS/prob2.html, 2000.

[28] A. Capone and M. Trubian, *Channel assignment problem in cellular systems: a new model and a tabu search algorithm*, IEEE Transactions on Vehicular Technology **48** (1999), no. 4, 1252–1260.

[29] M. Carlsson and M. Grindal, *Automatic frequency assignment for cellular telephones using constraint satisfaction techniques*, Proceedings of the Tenth International Conference on Logic Programming, 1993, pp. 648–665.

[30] D. J. Castelino, S. Hurley, and N. M. Stephens, *A tabu search algorithm for frequency assignment*, Annals of Operations Research **63** (1996), 301–319.

[31] K.-N. Chang and S. Kim, *Channel allocation in cellular radio networks*, Computers and Operations Research **24** (1997), 849–860.

[32] N. Christofides, *Graph theory: An algorithmic approach*, Academic Press, London, 1975.

[33] Luis M. Correia (ed.), *Wireless flexible personalized communications - COST 259: European co-operation in mobile radio research*, John Wiley & Sons Ltd., 2001, COST Action 259—Final Report.

[34] D. Costa, *On the use of some known methods for t-colourings of graphs*, Annals of Operations Research **41** (1993), 343–358.

[35] M. B. Cozzens and F. S. Roberts, *T-colorings of graphs and the channel assignment problem*, Congressus Numerantium **35** (1982), 191–208.

[36] C. Crisan and H. Mühlenbein, *The breeder genetic algorithm for frequency assignment*, Lecture Notes in Computer Science **1498** (1998), 897–906.

[37] ______, *The frequency assginment problem: A look at the performance of evolutionary search*, Lecture Notes in Computer Science **1363** (1998), 263–274.

[38] W. Crompton, S. Hurley, and N.M. Stephens, *A parallel genetic algorithm for frequency assignment problems*, Proceedings IMACS/IEEEInt. Symp. on Signal Processing, Robotics and Neural Networks (Lille, France), April 1994, pp. 81–84.

[39] M. Cuppini, *A genetic algorithm for channel assignment problems*, European Transactions on telecommunications and related technologies **5** (1994), 285–294.

[40] R. Dorne and J.-K. Hao, *An evolutionary approach for frequency assignment in cellular radio networks*, IEEE Int. Conference on Evolutionary Computing (Perth, Australia), 1995.

[41] ______, *Constraint handling in evolutionary search: A case study of the frequency assignment*, Lecture Notes in Computer Science **1141** (1996), 801–810.

[42] ______, *A new genetic local search algorithm for graph coloring*, Lecture Notes in Computer Science **1498** (1998), 745–754.

[43] N. Dunkin and S. M. Allen, *Frequency assignment problems: Representations and solutions*, Tech. Report CSD-TR-97–14, Royal Holloway, University of London, 1997, Available at http://www.dcs.rhbnc.ac.uk/research/constraints/publications/index.shtml.

[44] N. Dunkin, J. Bater, P. Jeavons, and D. Cohen, *Towards high order constraint representations for the frequency assignment problem*, Tech. Report CSD-TR-98–05, Royal Holloway University of London, 1998, Available at http://www.dcs.rhbnc.ac.uk/research/constraints/-publications/index.shtml.

[45] M. Duque-Antón, D. Kunz, and B. Rüber, *Channel assignment for cellular radio using simulated annealing*, IEEE Transactions on Vehicular Technology **42** (1993), 14–21.

[46] A. Eisenblätter, *Frequency assignment in GSM networks: Models, heuristics, and lower bounds*, Ph.D. thesis, Technische Universität Berlin, Berlin, Germany, 2001.

[47] A. Eisenblätter, M. Grötschel, and A. M. C. A. Koster, *Frequency assignment and ramifications of coloring*, ZIB-report 00–47, Konrad-Zuse-Zentrum für Informationstechnik Berlin, Berlin, Germany, 2000, To be published in Discussiones Mathematicae Graph Theory.

[48] A. Eisenblätter and A. M. C. A. Koster, *FAP web - A website devoted to frequency assignment*, URL: http://fap.zib.de, 2000.

[49] A. Eisenblätter and T. Kürner, *Benchmarking frequency allocation strategies*, Tech. Report COST 259, TD (00) 44, COST 259, April 2000.

[50] A. Eisenblätter, T. Kürner, and R. Fauß, *Radio planning algorithms for interference reduction in cellular networks*, Communications for the Millennium. COST 252/259 Joint Workshop, University of Bradford, 21–22 apr 1998. (COST 252/259, ed.), 1998, Available at http://fap.zib.de, pp. 87–92.

[51] R. Enders, *Unpublished method*, Siemens AG, Germany, 2000.

[52] P. Erdős, A. L. Rubin, and H. Taylor, *Choosability in graphs*, Congressus Numerantium **26** (1979), 125–157.

[53] M. Fischetti, C. Lepschy, G. Minerva, G. Romanin-Jacur, and E. Toto, *Frequency assignment in mobile radio systems using branch-and-cut techniques*, European Journal of Operational Research **123** (2000), 241–255.

[54] N. Funabiki and Y. Takefuji, *A neural network parallel algorithm for channel assignment problems in cellular radio networks*, IEEE Transactions on Vehicular Technology **41** (1992), 430–437.

[55] A. Gamst, *Some lower bounds for a class of frequency assignment problems*, IEEE Transactions on Vehicular Technology **35** (1986), 8–14.

[56] ______, *Application of graph theoretical methods to GSM radio network planning*, Proceedings of IEEE International Symposium on Circuits and Systems, vol. 2, 1991, pp. 942–945.

[57] A. Gamst and W. Rave, *On frequency assignment in mobile automatic telephone systems*, Proceedings of GLOBECOM'82, IEEE, 1982.

[58] A. I. Giortzis and L. F. Turner, *Application of mathematical programming to the fixed channel assignment problem in mobile radio networks*, IEE Proceedings - Communications **144** (1997), 257–264.

[59] S. De Givry, G. Verfaillie, and T. Schiex, *Bounding the optimum of constraint optimization problems*, Proceedings of the 3rd International Conference on Principles and Practice of Constraint Programming (CP-97), 1997.

[60] U. Gotzner, A. Gamst, and R. Rathgeber, *Statial traffic distribution in cellular networks*, Proc. IEEE VTC'97 Ottawa Canada, May 1997, pp. 1994–1998.

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

[62] J.-K. Hao and R. Dorne, *Study of genetic search for the frequency assignment problem*, Lecture Notes in Computer Science **1063** (1996), 333–344.

[63] J.-K. Hao, R. Dorne, and P. Galinier, *Tabu search for frequency assignment in moblie radio networks*, Journal of Heuristics **4** (1998), 47–62.

[64] J.-K. Hao and L. Perrier, *Tabu search for the frequency assignment problem in cellular radio networks*, European Journal of Operational Research (1999), (submitted), previously published as Technical report LGI2P, EMA-EERIE, Parc Scientifique Georges Besse, Nimes, France.

[65] M. Hellebrandt and H. Heller, *A new heuristic method for frequency assignment*, Tech. Report TD(00) 003, COST 259, Valencia, Spain, Jan. 2000.

[66] H. Heller, *Lower bound for philadelphia instance p6*, Result announced by D. H. Smith at the SPECMAN mailing list, February 2000.

[67] C. A. J. Hurkens and S. R. Tiourine, *Upper and lower bounding techniques for frequency assignment problems*, Tech. Report COSOR 95-34, Eindhoven University of Technology, 1995, Available at http://www.win.tue.nl/math/bs/comb_opt/hurkens/calma.html.

[68] S. Hurley, D. H. Smith, and S. U. Thiel, *FASoft: A system for discrete channel frequency assignment*, Radio Science **32** (1997), 1921–1939.

[69] RADIO instances, *Radio-broadcasting frequency assignment instances*, URL: ftp://ftp.dis.uniroma1.it/PUB/OR/mannino/freq/inst/radio/, 1998.

[70] F. J. Jaimes-Romero, D. Munoz-Rodriguez, and S. Tekinay, *Channel assignment in cellular systems using genetic algorithms*, Proceedings of the 46th IEEE Vehicular Technology Conference (Atlanta, USA), 1996, pp. 741–745.

[71] J. Janssen and K. Kilakos, *A polyhedral analysis of channel assignment problems based on tours (extended abstract)*, Proceedings of the 1997 IEEE International Conference on Communications, 1997, previously published as report CDAM-96-17, London School of Economics.

[72] ———, *An optimal solution to the "Philadelphia" channel assignment problem*, IEEE Transactions on Vehicular Technology **48** (1999), no. 3, 1012–1014.

[73] J. Janssen and T. Wentzell, *Lower bounds from tile covers for the channel assignment problem*, Tech. Report G-2000-09, GERAD, HEC, Montreal, Canada, 2000, preprint available at http://www.mscs.dal.ca/~janssen/cv/research.html.

[74] B. Jaumard, O. Marcotte, and C. Meyer, *Estimation of the quality of cellular networks using column generation techniques*, Tech. Report G-98-02, Ecole Polytechnique de Montréal, January 1998.

[75] ______, *Mathematical models and exact methods for channel assignment in cellular networks*, Telecommunications Network Planning (B. Sansáo and P. Soriano, eds.), Kluwer Academic Publishers, Boston, 1999, pp. 239–255.

[76] B. Jaumard, O. Marcotte, C. Meyer, and T. Vovor, *Comparison of column generation models for channel assignment in cellular networks*, Discrete Applied Mathematics **112** (2001), 217–240.

[77] A. Kapsalis, P. Chardaire, V. J. Rayward-Smith, and G. D. Smith, *The radio link frequency assignment problem: A case study using genetic algorithms*, Lecture Notes on Computer Science **993** (1995), 117–131.

[78] N. Karmarkar, M.G.C. Resende, and K.G. Ramakrishnan, *An interior point algorithm to solve computationally difficult set covering problems*, Mathematical Programming **52** (1991), 597–618.

[79] I. Katzela and M. Naghshineh, *Channel assignment schemes for cellular mobile telecommunication systems: A comprehensive survey*, IEEE Personal Communications Magazine (1996), 10–31.

[80] M. G. Kazantzakis, P. P. Demestichas, and M. E. Anagnostou, *Optimum frequency reuse in mobile telephone systems*, International Journal of Communications Systems **8** (1995), 185–190.

[81] J.-S. Kim, S. Park, P. Dowd, and N. Nasrabadi, *Cellular radio channel assignment using a modified hopfield network*, IEEE transactions on vehicular technology **46** (1996), no. 4, 957–967.

[82] A. Knälmann and A. Quellmalz, *Solving the frequency assignment problem with simulated annealing*, IEE conference publication **396** (1994), 233–240.

[83] A. W. J. Kolen, *A genetic algorithm for frequency assignment*, Tech. report, Universiteit Maastricht, 1999.

[84] A. W. J. Kolen, C. P. M. van Hoesel, and R. van der Wal, *A constraint satisfaction approach to the radio link frequency assignment problem*, Tech. Report 2.2.2, EUCLID CALMA project, 1994.

[85] A. M. C. A. Koster, *Frequency assignment – models and algorithms*, Ph.D. thesis, Universiteit Maastricht, 1999, Available at http://www.zib.de/koster/thesis.html.

[86] A. M. C. A. Koster, C. P. M. van Hoesel, and A. W. J. Kolen, *The partial constraint satisfaction problem: Facets and lifting theorems*, Operations Research Letters **23** (1998), no. 3-5, 89–97.

[87] ______, *Solving frequency assignment problems via tree-decomposition*, Tech. Report RM 99/011, Universiteit Maastricht, 1999, Available at http://www.zib.de/koster/.

[88] ______, *Lower bounds for minimum interference frequency assignment problems*, Ricerca Operativa **30** (2001), no. 94–95, 101–116, previously published as Research Memorandum 99/026, Universteit Maastricht; available at http://www.zib.de/koster/.

[89] D. Kunz, *Channel assignment for cellular radio using neural networks*, IEEE Transactions on Vehicular Technology **40** (1991), 188–193.

[90] W. K. Lai and G. G. Coghill, *Channel assignment through evolutionary optimization*, IEEE Transactions on Vehicular Technology **45** (1996), 91–95.

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

[92] V. Maniezzo and R. Montemanni, *An exact algorithm for the min-interference frequency assignment problem*, Tech. Report WP-CO0003, Scienze dell'Informazione, University of Bologna, Cesena, Italy, 2000.

[93] C. Mannino, G. Oriolo, and A. Sassano, *Weighted stable set problem in k-thin graphs*, Tech. Report 09–00, Universitá di Roma "La Sapienza", 2000.

[94] C. Mannino and A. Sassano, *An enumerative algorithm for the frequency assignment problem*, Discrete Applied Mathematics (2001), to appear, previously published as technical report 1096, Università di Roma La Sapienza.

[95] R. Mathar and J. Mattfeldt, *Channel assignment in cellular radio networks*, IEEE Transactions on Vehicular Technology **42** (1993), 647–656.

[96] S. Matsui and K. Tokoro, *A new genetic algorithm for minimum span frequency assignment using permutation and clique*, Proc. of GECCO-2000 (Genetic and Ecoluionary Computation Conference, Morgan Kaufmann Publishers, 2000, pp. 682–689.

[97] ———, *Improving the performance of a genetic algorithm for minimum span frequency assignment problem with an adaptive mutation rate and a new initialization method*, Proc. of GECCO-2001, Morgan Kaufmann Publishers, 2001, pp. 1359–1366.

[98] A. Mehrotra and M. A. Trick, *A column generation approach for graph coloring*, INFORMS Journal on Computing **8** (1996), 344–354.

[99] B. H. Metzger, *Spectrum management technique*, Fall 1970, Presentation at 38th National ORSA meeting (Detroit, MI).

[100] R. Montemanni, D.H. Smith, and S.M. Allen, *Lower bounds for fixed spectrum frequency assignment*, Annals of Operations Research **submitted to special issue on Frequency Assignment** (2000).

[101] R. A. Murphey, P. M. Pardalos, and M. G. C. Resende, *Frequency assignment problems*, Handbook of combinatorial optimization (D.-Z. Du and P. M. Pardalos, eds.), vol. Supplement Volume A, Kluwer Academic Publishers, 1999.

[102] G.L. Nemhauser and L.A. Wolsey, *Integer and combinatorial optimization*, Wiley, N.Y., 1988.

[103] C. Y. Ngo and V. O. K. Li, *Fixed channel assignment in cellular radio networks using a modified genetic algorithm*, IEEE Transactions on Vehicular Technology **47** (1998), 163–171.

[104] T. Toftegaard Nielsen and J. Wigard, *Performance enhancements in a frequency hopping GSM network*, Kluwers Academic Publishers, 2000, ISBN: 0 7923 7819 9.

[105] M. Padberg, *The boolean quadric polytope: Some characteristics, facets and relatives*, Mathematical Programming **45** (1989), 139–172.

[106] C. H. Papadimitriou and K. Steiglitz, *Combinatorial optimization: Algorithms and complexity*, Prentice-Hall, New Jersey, 1982.

[107] T. Park and C. Y. Lee, *Application of the graph coloring algorithm to the frequency assignment problem*, Journal of the Operations Research Society of Japan **39** (1996), 258–265.

[108] D. V. Pasechnik, *An interior point approximation algorithm for a class of combinatorial optimization problems: Implementation and enhancements*, Tech. report, Delft University of Technology, 1998.

[109] A. Petford and D. Welsh, *A randomised 3-colouring algorithm*, Discrete Mathematics **74** (1989), 253–261.

[110] A. Quellmalz, A. Knälmann, and B. Müller, *Efficient frequency assignment with simulated annealing*, IEE conference publication **407-2** (1995), 301–304.

[111] A. Raychaudhuri, *Intersection assignments, t-colourings and powers of graphs*, Ph.D. thesis, Rutgers University, 1985.

[112] ______, *Further results on t-coloring and frequency assignment problems*, SIAM Journal on Discrete Mathematics **7** (1994), 605–613.

[113] E. Del Re, R. Fantacci, and L. Ronga, *A synamic channel allocaiton technique based on hopfield neural networks*, IEEE Transactions on Vehicular Technology **45** (1996), 26–32.

[114] *ROADEF Challenge 2001*, URL: http://www.prism.uvsq.fr/~vdc/ROADEF/CHALLENGES/2001/, 2000.

[115] F. S. Roberts, *T-Colorings of graphs: Recent results and open problems*, Discrete Mathematics **93** (1991), 229–245.

[116] A. N. Rouskas, M. G. Kazantzakis, and M. E. Anagnostou, *Optimal channel assignment in cellular networks*, International Journal of Communication Systems **8** (1995), 359–364.

[117] C.K. Rushforth and W. Wang, *Local search for channel assignment in cellular mobile networks*, DIMACS Series in Discrete Mathematics and Theoretical Computer Science, vol. 35, American Mathematical Society, 1997, pp. 689–709.

[118] H. G. Sandalidis, P. P. Stavroulakis, and J. Rodriguez-Tellez, *Borrowing channel assignment strategies based on heuristic techniques for cellular systems*, IEEE Transactions on Neural Networks **10** (1999), 176–181.

[119] K.N. Sivarajan, R.J. McEliece, and J.W. Ketchum, *Channel assignment in cellular radio*, Proceedings of the 39th IEEE Vehicular Technology Conference, 1989, pp. 846–850.

[120] D. H. Smith, S. Hurley, and S. M. Allen, *A new lower bound for the channel assignmnet problem*, IEEE Transactions on Vehicular Technology **to appear** (2000).

[121] D. H. Smith, S. Hurley, and S. U. Thiel, *Improving heuristics for the frequency assignment problem*, European Journal of Operational Research **107** (1998), 76–86.

[122] K. A. Smith and M. Palaniswami, *Static and dynamic channel assignment using neural networks*, IEEE Journal on Selected Areas in Communications **15** (1997), 238–249.

[123] C. W. Sung and W. S. Wong, *Sequential packing algorithm for channel assignment under cochannel and adjacent channel interference constraint*, IEEE Transactions on Vehicular Technology **46** (1997), 676–685.

[124] D. Tcha, Y. Chung, and T. Choi, *A new lower bound for the frequency assignment problem*, IEEE/ACM Transactions on Networking **5** (1997), 34–39.

[125] H. Thuve, *Frequency planning as a set partitioning problem*, European Journal of Operational Research **6** (1981), 29–37.

[126] S. R. Tiourine, C. A. J. Hurkens, and J. K. Lenstra, *An overview of algorithmic approaches to frequency assignment problems*, Calma Symposium on Combinatorial Algorithms for Military Applications, 1995, Available at http://www.win.tue.nl/math/bs/comb_opt/hurkens/calma.html, pp. 53–62.

[127] ———, *Local search algorithms for the radio link frequency assignment problem*, Telecommunication systems **13** (2000), 293–314.

[128] E. Tsang and C. Voudouris, *Solving the radio link frequency assignment problem using guided local search*, NATO Symposium on Radio Length Frequency Assignment (Aalborg, Denmark), 1998, Available at http://cswww.essex.ac.uk/CSP/papers.html.

[129] L. J. De Urries, M. A. D. Guerra, and I. Berberana, *Frequency planning using simulated annealing*, Tech. report, COST 259, 2000.

[130] C. Valenzuela, S. Hurley, and D. H. Smith, *A permutation based genetic algorithm for minimum span frequency assignment*, Lecture Notes in Computer Science **1498** (1998), 907–916.

[131] G. Verfaillie, M. Lemaître, and T. Schiex, *Russian doll search for solving constraint optimization problems*, Proceedings of the 13th National Conference on Artificial Intelligence (AAAI-96) (Portland, OR, USA), 1996, pp. 181–187.

[132] V. G. Vizing, *Critical graphs with given chromatic class (russian)*, Diskret. Analiz **5** (1965), 9–17.

[133] J. P. Walser, *Feasible cellular frequency assignment using constraint programming abstractions*, Proceedings of the Workshop on Constraint Programming Applications (CP96) (Cambridge, Massachusetts, USA), 1996.

[134] W. Wang and C. K. Rushforth, *An adaptive local-search algorithm for the channel-assignment problem (CAP)*, IEEE Transactions on Vehicular Technology **45** (1996), 459–466.

[135] J. P. Warners, T. Terlaky, C. Roos, and B. Jansen, *A potential reduction approach to the frequency assignment problem*, Discrete Applied Mathematics **78** (1997), 251–282.

[136] J. Zerovnik, *Experiments with a randomized algorithm for a frequency assignment problem*, Tech. Report 97–27, Ecole Normale Supérieure de Lyon, 1997.

[137] M. Zhang and T.P. Yum, *The nonuniform compact pattern allocation algorithm for cellular mobile systems*, IEEE Transactions on Vehicular Technology **40** (1991), 387–391.

[138] J. A. Zoellner and C. L. Beall, *A breakthrough in spectrum conserving frequency assignment technology*, IEEE Transactions on Electromagnetic Compatiblity **19** (1977), 313–319.

A Comparison of Benchmark Results

In this appendix, a description and a comparison of the results for frequency assignment benchmark instances is presented. From the instances mentioned in Section 2.2, the Philadelphia, CALMA, and COST 259 instances are publicly available and used by several research groups. The other instances are not publicly available and/or only used by a limited number of researchers, which makes a comparison difficult. The latest results for the benchmark instances compared here, as well as the instances themselves can be found at FAP web [48].

A.1 Philadelphia benchmarks

Table 1 shows for the Philadelphia instances, the demand vector for the 21 hexagons, see Figure 4(a), page 8, and the reuse distances as explained in Section 2.1. The frequency domains $F(v)$ are simply $\mathbb{Z}_+$, in which case minimization of the span is equivalent to minimization of the maximum used frequency. Note that, there is a difference of one between minimum span and maximum used frequency. In conformity with [130], the instances are denoted by `P1-P9`. Some of them are also known as `E3-E9`, after [68]. In Table 2, the results of both lower bounding and upper bounding techniques are compared.

A.2 CALMA benchmarks

The CALMA instances (cf. Section 2.2) differ from other frequency assignment problems by their specific distance /separation constraints. Besides the minimum distance constraints, the instances also contain equality constraints to model that two frequencies at a fixed distance have to be assigned to the corresponding vertices. The distance is the same for all constraints and every vertex is contained in exactly one equality constraint. Moreover, the domains are constructed in such a way that for every frequency there exists only one 'matching' frequency. Altogether, these characteristics provide the possibility to reduce the size of the instances to half the original size whenever that may be profitable. The number of available frequencies for a vertex is 40 on average. The set of instances contains MO-FAPs, MS-FAPs, as well as MI-FAPs. The number of vertices ranges from 200 to 916, the number of edges from 1200 to 5700. For more information on the instances, we refer to [2, 26].

Table 3 shows the results for the MO-FAP instances, Table 4 the results for the MS-FAP instances, and finally Table 5 the results for the MI-FAP.

A.3 COST 259 benchmarks

In the COST 259 project [33] (cf. Section 2.2) on Wireless Flexible Personalized Communications, 32 realistic GSM network planning scenarios were collected and served as benchmark for comparing algorithms within COST 259. The scenarios together with several contributed frequency plans are made available on the FAP web [48]. Here, we restrict ourselves to a brief description of the scenarios and the key figures of the computed assignments.

The scenarios have been contributed by E-Plus Mobilfunk GmbH (`bradford`-t-p, `bradford_nt`-t-p, and `K`), Siemens AG (`siemens` instances), and Swisscom Ltd. (`Swisscom` instance). The `bradford`, `bradford_nt`, and `K` instances originate from a GSM 1800 Network. The wildcards t and p stand for five different traffic loads and three interference predictions on the basis of different signal prediction models. The basic traffic load is drawn at random according to an empirically observed distribution [60]. This traffic is then scaled with the factors t prior to applying the Erlang-B formula in order to obtain the required number of TRXs per cell. The signal predictions (p) are done assuming free

instance	demand vector c_v	reuse distances
P1 (E3)	$(8, 25, 8, 8, 8, 15, 18, 52, 77, 28, 13, 15, 31, 15, 36, 57, 28, 8, 10, 13, 8)$	$(2\sqrt{3}, \sqrt{3}, 1, 1, 1, 0)$
P2 (E4)	$(8, 25, 8, 8, 8, 15, 18, 52, 77, 28, 13, 15, 31, 15, 36, 57, 28, 8, 10, 13, 8)$	$(\sqrt{7}, \sqrt{3}, 1, 1, 1, 0)$
P3 (E5)	$(5, 5, 5, 8, 12, 25, 30, 25, 30, 40, 40, 45, 20, 30, 25, 15, 15, 30, 20, 20, 25)$	$(2\sqrt{3}, \sqrt{3}, 1, 1, 1, 0)$
P4 (E6)	$(5, 5, 5, 8, 12, 25, 30, 25, 30, 40, 40, 45, 20, 30, 25, 15, 15, 30, 20, 20, 25)$	$(\sqrt{7}, \sqrt{3}, 1, 1, 1, 0)$
P5 (E7)	$(20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20)$	$(2\sqrt{3}, \sqrt{3}, 1, 1, 1, 0)$
P6	$(20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20)$	$(\sqrt{7}, \sqrt{3}, 1, 1, 1, 0)$
P7 (E8)	$(16, 50, 16, 16, 16, 30, 36, 104, 154, 56, 26, 30, 62, 30, 72, 114, 56, 16, 20, 26, 16)$	$(2\sqrt{3}, \sqrt{3}, 1, 1, 1, 0)$
P8	$(8, 25, 8, 8, 8, 15, 18, 52, 77, 28, 13, 15, 31, 15, 36, 57, 28, 8, 10, 13, 8)$	$(2\sqrt{3}, 2, 1, 1, 1, 0)$
P9 (E9)	$(32, 100, 32, 32, 32, 60, 72, 208, 308, 112, 52, 60, 124, 60, 144, 228, 112, 32, 40, 52, 32)$	$(2\sqrt{3}, \sqrt{3}, 1, 1, 1, 0)$

Table 1: Characteristics Philadelphia benchmark instances.

instance	lower bounds						optimal value	upper bounds											
	Graph Theory [55]	TSP bound [71, 72]	Diverse methods [68, 121]	Graph Theory [123]	Graph Theory [66]	TSP + d-walks [9]		Best Sequential [68]	Tabu Search [68]	Simulated Annealing [68]	Genetic Algorithm [130]	Subgraph Extension [6, 68, 120, 121]	TSP + d-walks [9]	Genetic Algorithm [12, 13]	Genetic Algorithm [96, 97]	Sequential Coloring [119]	Generalized Sequential Packing 1 [123]	Generalized Sequential Packing 2 [123]	Local Search [134]
P1	413	426	426	426	-	426	426	447	428	428	426	426	426	-	426	459	440	450	432
P2	413	426	426	426	-	426	426	475	429	438	426	426	426	426	426	446	436	444	-
P3	245	-	257	252	-	257	257	284	269	260	258	257	275	-	257	282	291	273	263
P4	245	252	252	252	-	252	252	268	257	259	253	252	253	252	252	268	273	268	-
P5	239	-	239	177	-	239	239	250	240	239	239	-	239	-	239	-	-	-	-
P6	139	177	178	177	179	177	179	230	188	200	198	179	183	-	179	-	-	-	-
P7	830	-	855	855	-	855	855	894	858	858	856	855	855	-	855	-	-	-	-
P8	449	-	524	427	-	523	524	592	535	546	527	524	524	-	524	-	-	-	-
P9	1,664	-	1,713	1,713	-	1,713	1,713	1,800	1,724	1,724	-	1,713	1,713	-	-	-	-	-	-

Table 2: Results Philadelphia benchmark instances. Values on a gray background indicate optimality.

instance	Clique bound [1]	Coloring number [1]	Generalized Coloring [1]	Linear Programming [1]	Clique bound [127]	Constraint Satisfaction [84]	optimal value (range)	Constraint Satisfaction [84]	Simulated Annealing [127]	Variable Depth Search [127]	Tabu Search [127]	Tabu Search [22]	GENET [22]	Genetic Algorithm [77]	Potential Reduction [135]	Gradient Descent [108]	Evolutionary Search [37]
	lower bounds							upper bounds									
CELAR 01	12	14	16	16	14	16	16	16	16	16	16	18	16	20	16	16	-
CELAR 02	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14	14
CELAR 03	12	14	14	14	14	14	14	14	14	14	14	14	14	16	16	14	14
CELAR 04	-	-	-	46	46	46	46	46	46	46	46	46	46	46	46	46	-
CELAR 11	20	20	20	22	22	22	22	22	24	24	22	24	24	32	-	22	-
GRAPH 01	-	-	-	18	18	18	18	18	-	-	18	18	18	20	18	18	18
GRAPH 02	-	-	-	14	14	14	14	14	-	-	14	16	14	16	14	14	14
GRAPH 08	-	-	-	-	16	-	16-18	-	-	-	20	24	22	-	18	18	-
GRAPH 09	-	-	-	-	18	18	18	18	-	-	22	22	22	28	18	18	-
GRAPH 14	-	-	-	-	8	-	8	10	-	-	10	12	-	14	10	8	-

Table 3: Results CALMA MO-FAP benchmark instances. Values on a gray background indicate optimality.

instance	Linear Programming [1]	Quadratic Programming [127]	Constraint Satisfaction [84]	Tabu Search [127]	Tabu Search [21]	GENET [22]	Potential Reduction [135]	Gradient Descent [108]
	lower bounds			upper bounds				
CELAR 05	792	792	792	792	792	792	792	792
GRAPH 03	-	-	380	380	-	-	-	380
GRAPH 04	-	-	394	394	-	-	-	394
GRAPH 10	-	-	394	394	-	-	394	394

Table 4: Results CALMA MS-FAP benchmark instances. All results derived are optimal.

instance	Linear Programming [1]	Quadratic Programming [127]	Constraint Satisfaction [59]	Tree Decomposition [85,87]	Integer Programming [85,88]	optimal value (range)	Genetic Algorithm [83]	Genetic Algorithm [77]	Simulated Annealing [127]	Variable Depth Search [127]	Potential Reduction [135]	Gradient Descent [108]
	lower bounds						upper bounds					
CELAR 06	5	-	3,389	3,389	2,321	3,389	3,389	3,456	3,671	3,532	4,539	4,564
CELAR 07	5	-	-	300,000	180,525	300,000 - 343,592	343,592	1,670,572	567,949	344,103	-	831,926
CELAR 08	-	-	-	87	150	150 - 262	262	612	276	299	-	533
CELAR 09	-	14,969	-	15,571	15,571	15,571	15,571	15,599	15,571	15,573	15,775	15,770
CELAR 10	-	31,204	-	31,516	31,516	31,516	31,516	31,517	31,516	31,516	32,460	31,517
GRAPH 05	-	-	-	221	221	221	221	293	223	-	-	452
GRAPH 06	-	-	-	4,123	4,123	4,123	4,123	16,020	4,189	-	-	15,047
GRAPH 07	-	-	-	4,324	4,324	4,324	4,324	5,990	4,324	-	-	14,183
GRAPH 11	-	-	-	2,553	3,016	3,016 - 3,080	3,080	30,312	3,513	-	-	14,692
GRAPH 12	-	-	-	11,827	11,827	11,827	11,827	15,208	11,827	-	-	17,372
GRAPH 13	-	-	-	8,676	9,925	9,925 - 10,110	10,110	49,205	11,130	-	-	41,784

Table 5: Results CALMA MI-FAP benchmark instances. Values on a gray background indicate optimality.

space propagation with a decay factor of 1.5 (`free`), using a Modified Okumura-Hata model (`race`), and using a model developed by E-Plus (`eplus`), see [50] for details. The `siemens` and `Swisscom` scenarios originate from GSM 900 networks. More characteristics of the COST 259 instances are collected in Table 6. Double spectrum size values mean that the spectrum consists of two part of consecutive frequencies.

Scenario	#sites	#cells	avg. #TRXs/cell	max. #TRXs/cell	spectrum size	$\|V\|$	density (%)
`K`	92	264	1.01	2	50	267	56.57
`bradford-0-`p	649	1886	1.00	1	75	1886	13.59
`bradford-1-`p	649	1886	1.05	3	75	1971	13.46
`bradford-2-`p	649	1886	1.17	5	75	2214	13.50
`bradford-4-`p	649	1886	1.47	9	75	2775	13.44
`bradford-10-`p	649	1886	2.02	12	75	4145	13.41
`Siemens1`	179	506	1.84	4	20, 23	930	9.03
`Siemens2`	86	254	3.85	6	4, 72	977	49.17
`Siemens3`	366	894	1.82	3	55	1623	9.18
`Siemens4`	279	760	3.66	5	39	2785	10.50
`Swisscom`	87	148	2.09	4	3, 49	310	8.29

Table 6: Characteristics of COST 259 scenarios

A total of 115 assignments are currently available for the different scenarios. Tables 7 and 8 display the key figures of these assignments: the number of separation violations, the total amount of interference, the maximum level of co-channel interference, and the maximum level of adjacent-channel interference. More statistics of these assignments are available at the FAP web [48] or in Eisenblätter and Kürner [49]. Both Hellebrandt and Heller contributed results on the Threshold Accepting method [65]. These results are presented in different columns, marked (1) and (2). The method used by Enders [51] has not been published. The semidefinite programming approach of Eisenblätter [46] provides a lower bound on the total amount of unavoidable co-channel interference.

B Digest of Frequency Assignment Literature

In Section 3, we classified the frequency assignment problems according to their objective function. For each of the resulting four classes of FAP, we summarized the literature on this problem in Tables 9 to 12. For each paper we depicted the applied technique(s) and the used instances. Table 9 shows an overview of the literature available for the Maximum Service FAP (Max-FAP) and the Minimum Blocking FAP (MB-FAP). Next, Table 10 is devoted to the Minimum Order FAP (MO-FAP). The literature for the Minimum Span FAP (MS-FAP) is collected in Table 11. Finally, the Minimum Interference FAP (MI-FAP) is the topic of Table 12.

Scenario	# separation violations: DSATUR w.cost(5%)+It.1-Opt [19]	Simulated Annealing [129]	Simulated Annealing [11]	Threshold Accepting (1) [65]	Threshold Accepting (2) [65]	Unpublished [51]	total interference: Semidefinite Programming [46]	DSATUR w.cost(5%)+It.1-Opt [19]	Simulated Annealing [129]	Simulated Annealing [11]	Threshold Accepting (1) [65]	Threshold Accepting (2) [65]	Unpublished [51]
K	-	-	-	-	0	-	0.18	-	-	-	-	0.45	-
bradford-0-eplus	0	-	0	0	-	0	0.01	1.10	-	0.80	0.91	-	1.02
bradford-0-free	0	-	0	0	-	-	-	0.00	-	0.00	0.00	-	-
bradford-0-race	0	-	0	0	-	-	-	0.00	-	0.00	0.00	-	-
bradford_nt-1-eplus	0	-	0	0	0	-	0.03	1.78	-	1.04	1.10	0.86	-
bradford_nt-1-free	0	-	0	0	-	-	-	0.00	-	0.00	0.00	-	-
bradford_nt-1-race	0	-	0	0	-	-	-	0.00	-	0.00	0.00	-	-
bradford_nt-2-eplus	0	-	0	0	0	-	0.47	6.23	-	3.79	4.07	3.17	-
bradford_nt-2-free	0	-	0	0	-	-	-	0.00	-	0.00	0.00	-	-
bradford_nt-2-race	0	-	0	0	-	-	-	0.00	-	0.00	0.00	-	-
bradford_nt-4-eplus	0	-	0	0	0	-	0.45	29.17	-	19.00	19.51	17.73	-
bradford_nt-4-free	0	-	0	0	-	-	-	0.00	-	0.00	0.00	-	-
bradford_nt-4-race	0	-	0	0	-	-	-	0.00	-	0.00	0.00	-	-
bradford_nt-10-eplus	0	-	0	0	0	-	2.71	208.14	-	148.12	158.66	146.20	-
bradford_nt-10-free	0	-	0	0	0	-	-	15.34	-	8.63	7.59	5.86	-
bradford_nt-10-race	0	-	0	0	0	-	-	4.39	-	1.73	1.73	1.07	-
bradford-1-eplus	0	-	0	0	-	-	-	56.33	-	33.99	40.52	-	-
bradford-1-free	0	-	0	0	-	0	-	1.51	-	0.16	0.20	-	0.30
bradford-1-race	0	1	0	0	-	0	-	0.16	0.08	0.03	0.04	-	0.08
bradford-2-eplus	0	55	0	0	-	-	-	123.92	111.98	80.03	89.96	-	-
bradford-2-free	0	12	0	0	-	0	-	12.93	9.02	2.95	3.69	-	4.24
bradford-2-race	0	-	0	0	-	0	-	1.45	-	0.42	0.42	-	0.80
bradford-4-eplus	0	-	0	0	-	-	-	253.05	-	167.70	185.42	-	-
bradford-4-free	0	-	0	0	-	-	-	60.90	-	22.09	31.02	-	-
bradford-4-race	0	-	0	0	-	-	-	10.25	-	3.04	3.30	-	-
bradford-10-eplus	1	-	0	0	-	-	-	554.00	-	400.00	435.28	-	-
bradford-10-free	4	-	0	0	-	-	-	213.96	-	117.80	127.33	-	-
bradford-10-race	0	-	0	0	-	-	-	62.35	-	30.22	32.60	-	-
Siemens1	-	-	0	0	0	0	0.12	-	-	2.78	2.53	2.30	3.36
Siemens2	-	-	0	0	0	0	6.95	-	-	15.46	14.75	15.05	17.33
Siemens3	-	-	0	0	0	0	0.41	-	-	6.75	5.63	5.26	8.42
Siemens4	-	-	0	0	0	0	27.63	-	-	89.15	83.57	80.97	105.82
Swisscom	-	-	0	0	6	-	-	-	-	27.36	29.52	17.71	-

Table 7: Key-values of COST 259 assignments (1)

Scenario	max. co-channel interference						max. adj. channel interference					
	DSATUR w.cost(5%)+It.1-Opt [19]	Simulated Annealing [129]	Simulated Annealing [11]	Threshold Accepting (1) [65]	Threshold Accepting (2) [65]	Unpublished [51]	DSATUR w.cost(5%)+It.1-Opt [19]	Simulated Annealing [129]	Simulated Annealing [11]	Threshold Accepting (1) [65]	Threshold Accepting (2) [65]	Unpublished [51]
`K`	-	-	-	-	0.02	-	-	-	-	-	0.00	-
`bradford-0-eplus`	0.01	-	0.03	0.02	-	0.02	0.01	-	0.01	0.02	-	0.01
`bradford-0-free`	0.00	-	0.00	0.00	-	-	0.00	-	0.00	0.00	-	-
`bradford-0-race`	0.00	-	0.00	0.00	-	-	0.00	-	0.00	0.00	-	-
`bradford_nt-1-eplus`	0.01	-	0.02	0.03	0.02	-	0.01	-	0.01	0.00	0.01	-
`bradford_nt-1-free`	0.00	-	0.00	0.00	-	-	0.00	-	0.00	0.00	-	-
`bradford_nt-1-race`	0.00	-	0.00	0.00	-	-	0.00	-	0.00	0.00	-	-
`bradford_nt-2-eplus`	0.02	-	0.06	0.04	0.04	-	0.02	-	0.01	0.01	0.01	-
`bradford_nt-2-free`	0.00	-	0.00	0.00	-	-	0.00	-	0.00	0.00	-	-
`bradford_nt-2-race`	0.00	-	0.00	0.00	-	-	0.00	-	0.00	0.00	-	-
`bradford_nt-4-eplus`	0.08	-	0.10	0.11	0.16	-	0.04	-	0.03	0.03	0.01	-
`bradford_nt-4-free`	0.00	-	0.00	0.00	-	-	0.00	-	0.00	0.00	-	-
`bradford_nt-4-race`	0.00	-	0.00	0.00	-	-	0.00	-	0.00	0.00	-	-
`bradford_nt-10-eplus`	0.36	-	0.34	0.46	0.34	-	0.09	-	0.15	0.10	0.08	-
`bradford_nt-10-free`	0.17	-	0.07	0.10	0.05	-	0.09	-	0.02	0.02	0.01	-
`bradford_nt-10-race`	0.22	-	0.07	0.09	0.06	-	0.08	-	0.01	0.02	0.01	-
`bradford-1-eplus`	0.12	-	0.15	0.31	-	-	0.09	-	0.03	0.07	-	-
`bradford-1-free`	0.05	-	0.01	0.02	-	0.02	0.06	-	0.00	0.00	-	0.00
`bradford-1-race`	0.01	0.01	0.00	0.01	-	0.01	0.00	0.00	0.00	0.00	-	0.00
`bradford-2-eplus`	0.27	0.34	0.25	0.28	-	-	0.16	0.19	0.06	0.07	-	-
`bradford-2-free`	0.15	0.16	0.05	0.05	-	0.05	0.20	0.25	0.01	0.06	-	0.06
`bradford-2-race`	0.06	-	0.02	0.02	-	0.04	0.08	-	0.00	0.01	-	0.03
`bradford-4-eplus`	0.41	-	0.45	0.45	-	-	0.20	-	0.09	0.15	-	-
`bradford-4-free`	0.27	-	0.15	0.21	-	-	0.25	-	0.10	0.23	-	-
`bradford-4-race`	0.16	-	0.08	0.09	-	-	0.16	-	0.03	0.07	-	-
`bradford-10-eplus`	0.70	-	0.52	0.68	-	-	0.40	-	0.18	0.25	-	-
`bradford-10-free`	0.39	-	0.38	0.37	-	-	0.37	-	0.20	0.38	-	-
`bradford-10-race`	0.31	-	0.38	0.30	-	-	0.33	-	0.16	0.15	-	-
`Siemens1`	-	-	0.04	0.03	0.03	0.05	-	-	0.04	0.03	0.03	0.04
`Siemens2`	-	-	0.07	0.06	0.11	0.08	-	-	0.02	0.02	0.02	0.02
`Siemens3`	-	-	0.05	0.03	0.04	0.21	-	-	0.03	0.03	0.03	0.05
`Siemens4`	-	-	0.24	0.18	0.17	0.27	-	-	0.03	0.04	0.05	0.05
`Swisscom`	-	-	0.00	0.00	0.00	-	-	-	0.96	0.96	0.97	-

Table 8: Key-values of COST 259 assignments (2)

source	Tree Search	Integer Linear Programming	Greedy	Simulated Annealing	Other	CSELT	Other	comments
	applied techniques					instances		
Chang and Kim [31]					⋆		⋆	Lagrangian relaxation and subgradient optimization
Fischetti et al. [53]		⋆				⋆		
Jaumard et al. [74, 76]		⋆					⋆	Column generation
Kazantzakis et al. [80]		⋆					⋆	
Mannino and Sassano [94]	⋆					⋆		Solve and Extend
Mathar and Mattfeldt [95]				⋆				
Rouskas et al. [116]		⋆					⋆	
Zhang and Yum [137]			⋆					

Table 9: Overview literature Max-FAP and MB-FAP

source	Tree Search	Integer Linear Programming	Genetic Algorithms	Other	CALMA	Other	comments
	applied techniques				instances		
Aardal et al. [1]	⋆	⋆			⋆		Graph theoretical lower bounds
Baybars [10]		⋆				⋆	
Dorne and Hao [40, 41]			⋆			⋆	
Giortzis and Turner [58]	⋆	⋆				⋆	
Hao and Dorne [62]			⋆			⋆	
Kolen et al. [84]	⋆				⋆		Constraint Logic Programming
Walser [133]				⋆		⋆	Constraint Programming
Warners et al. [135]				⋆	⋆		Potential Reduction Method

Table 10: Overview literature MO-FAP

source	Graph Theory	Integer Linear Programming	Greedy	Tabu Search	Simulated Annealing	Genetic Algorithms	Neural Networks	Other	Philadelphia	CALMA	Other	comments
	applied techniques								instances			
Allen et al. [6, 7]	$\star$	$\star$							$\star$			TSP bound, subgraphs
Avenali et al. [9]	$\star$								$\star$			TSP bound, d-walks
Beckmann and Killat [11]						$\star$			$\star$			
Beckmann and Killat [13]						$\star$			$\star$			Evolutionary Search
Cozzens and Roberts [35, 115]	$\star$											
Gamst [55]	$\star$								$\star$			
Gamst and Rave [57]	$\star$								$\star$			
Hao et al. [63]				$\star$							$\star$	
Hao and Perrier [64]				$\star$							$\star$	
Hurley et al. [68]	$\star$		$\star$	$\star$	$\star$	$\star$				$\star$		FASoft software
Janssen and Kilakos [71, 72]	$\star$	$\star$							$\star$			TSP bound
Janssen and Wentzell [73]	$\star$	$\star$							$\star$			Tile Covers
Maniezzo and Carbonaro [91]								$\star$	$\star$	$\star$		ANT Colony optimzation
Mannino and Sassano [94]								$\star$			$\star$	Solve and Extend
Matsui and Tokoro [96, 97]						$\star$			$\star$			
Park and Lee [107]				$\star$								Kernighan-Lin heuristic
Raychaudhuri [111, 112]	$\star$											TSP bound
Rushforth and Wang [117]								$\star$			$\star$	Local Search
Sivarajan et al. [119]			$\star$						$\star$			
Smith et al. [121]								$\star$	$\star$			Local Search
Smith and Palaniswami [122]							$\star$		$\star$			
Sung and Wong [123]			$\star$								$\star$	
Tcha et al. [124]	$\star$	$\star$							$\star$			
Wang and Rushforth [134]								$\star$			$\star$	Local Search
Zoellner and Beall [138]			$\star$								$\star$	

Table 11: Overview literature MS-FAP

source	applied techniques											instances				comments
	Constraint Programming	Integer Linear Programming	Tree Search	Tree Decomposition	Semidefinite Programming	Greedy	Tabu Search	Simulated Annealing	Genetic Algorithms	Neural Networks	Other	Philadelphia	CALMA	COST 259	Other	
Aardal et al. [1]		⋆											⋆			
Adjakplé and Jaumard [3]						⋆	⋆								⋆	
Al-Khaled [4]								⋆							⋆	
Allen et al. [5]											⋆				⋆	nonlinear programming
Beckmann and Killat [12]									⋆					⋆		
Borndörfer et al. [19]						⋆									⋆	
Borndörfer et al. [20]											⋆				⋆	orientation model
Bouju et al. [21]							⋆						⋆			
Bouju et al. [22]										⋆			⋆			
Box [23]						⋆				⋆						
Carlsson and Grindal [29]	⋆															
Castelino et al. [30]							⋆				⋆				⋆	
Crisan and Mühlenbein [36, 37]									⋆				⋆		⋆	Evolutionary Search
Crompton et al. [38]									⋆						⋆	
Cuppini [39]									⋆						⋆	
Del Re et al. [113]										⋆					⋆	
Duque-Antón et al. [45]								⋆							⋆	
Eisenblätter [46]					⋆	⋆					⋆			⋆		Local Search
Funabiki and Takefuji [54]										⋆					⋆	
Hellebrandt and Heller [65]								⋆						⋆		Threshold Accepting
Hurkens and Tiourine [67]							⋆	⋆					⋆			Quadratic Programming
Jaimes-Romero et al. [70]									⋆						⋆	
De Urries et al. [129]								⋆						⋆		
Kapsalis et al. [77]									⋆				⋆			
Kim et al. [81]										⋆					⋆	
Knälmann et al. [82, 110]								⋆							⋆	
Kolen [83]									⋆				⋆			
Koster et al. [86]		⋆											⋆			See also [85]
Koster et al. [87]				⋆									⋆			See also [85]
Koster et al. [88]		⋆		⋆									⋆			See also [85]
Kunz [89]										⋆					⋆	
Lai and Coghill [90]									⋆						⋆	Evolutionary Search
Maniezzo and Montemanni [92]			⋆										⋆			Lower bounding techniques
Mannino et al. [93]											⋆					Local Search
Montemanni et al. [100]		⋆										⋆			⋆	
Ngo and Li [103]									⋆						⋆	
Pasechnik [108]											⋆		⋆			Interior Point Method
Tiourine et al. [67, 127]							⋆	⋆					⋆			
Tsang and Voudouris [128]											⋆		⋆			Local Search
Valenzuela et al. [130]						⋆			⋆			⋆				
Verfaillie et al. [131]	⋆												⋆			Russian Doll Search
Zerovnik [136]								⋆							⋆	

Table 12: Overview literature MI-FAP