Upper and lower bounds for the fixed spectrum frequency assignment problem

Roberto Montemanni

Division of Mathematics and Statistics
School of Technology
University of Glamorgan

A submission presented in partial
fulfilment of the requirements of the
University of Glamorgan/Prifysgol Morgannwg
for the degree of Doctor of Philosophy

November 2001

Acknowledgements

I would like to take this opportunity to express my sincerest gratitude to my Director of Studies, Prof. Derek Smith and my Supervisor, Dr. Stuart Allen, for their invaluable guidance throughout this research.

Abstract

The frequency assignment problem involves the assignment of discrete channels (frequencies) to the transmitters of a radio network. A separation between the frequencies assigned to transmitters close to each other is required to avoid interference. Unnecessary separation causes an excess requirement for spectrum, which is a valuable resource. Consequently good assignments minimise both interference and the spectrum required.

The subject of this thesis is the fixed spectrum frequency assignment problem, where the spectrum available is given and the target is to minimise the total interference of the system.

Interference is modelled through binary constraints, and consequently the problem, which is treated as a combinatorial optimisation problem, can be represented by an undirected weighted graph.

A summary of some of the integer programming formulations which model the problem is presented, together with a brief dimensional study of them.

An efficient implementation of two well-known metaheuristic algorithms, adapted to the problem treated, is described.

Some novel lower bounding techniques which, given a problem, work by combining lower bounds calculated for some of its clique-like subproblems are presented. The key idea is that it is quite easy to calculate tight lower bounds for problems represented by complete graphs (cliques). The lower bounds for clique-like subproblems are produced by two different methods, the first of which is based on the solution of a linear program, while the second is based on a closed formula. The most effective method to generate estimates for general problems is based on a linear program which is reinforced with inequalities derived from the lower bounds calculated on its clique-like subproblems.

The last part of the thesis is dedicated to improvements to the lower bounding techniques, both for those working on general problems and for those developed for cliques only.

Detailed computational results, obtained on a wide range of benchmarks, are reported.

Contents

List of Tables

List of Figures

List of Mathematical Formulations

Chapter 1

Introduction

1.1 A brief history of wireless communication

The history of wireless communication via radio waves begins in the last decades of the Nineteenth Century, when several researchers started experimenting with this new frontier for information transmission. Even before 1900 it was possible to transmit signals across the Atlantic Ocean, and in 1905 it was quite common for ships to communicate with shore stations using wireless telegraphy. After World War I, radio and (later) television broadcasting became part of the everyday life for most people in our society. In the last fifty years many other applications of wireless communication have been introduced, both in the military field and in the civil one. Nowadays some of the most important technologies of modern society are based on wireless communication. In Section 1.3 we will describe some of these applications, following a brief description of how wireless networks work given in Section 1.2.

1.2 A description of wireless networks

Each communication in a wireless network is realised using a transmitter and a receiver. The transmitter emits a signal (encoded information) usually on a specific frequency. This frequency is known to the receiver, which is able to translate the signal back into information. When two transmitters use frequencies that are too close together in the electromagnetic spectrum, their signals may interfere. Some of the factors that affect interference (beside frequency separation) are the respective distances between the transmitters and the receiver, the direction and the power of the transmitted signals, the terrain of the network's area and the weather conditions. By a closer study of the physics, more complex causes of interference can be identified, such as intermodulation products, spurious emissions and spurious responses (see Loxton [68] for a more detailed description of these phenomena).

The available radio spectrum is a limited resource and the growth of demand for it in recent decades has pushed its price up, increasing the importance of good network planning. Such planning is based on frequency reuse within the same network and aims to achieve more efficient use of the frequency spectrum, maintaining the quality level of the network at a high standard. The *Frequency Assignment Problem* (*FAP*) is defined as the optimisation problem whose target is to find the best network planning possible.

A universally accepted convention in the theory of frequency assignment is to see a continuous frequency band $[f_{\min}, f_{\max}]$ (where $f_{\min}$ is the lowest frequency and $f_{\max}$ is the highest one) as a set of discrete and contiguous *channels* $\{0, 1, \ldots, N-1\}$, each with the same bandwidth B, where $N = (f_{\max} - f_{\min})/B$.

This view of the spectrum as a discrete resource is extremely useful, is realistic and is commonly adopted by service providers. In Figure 1.1 we

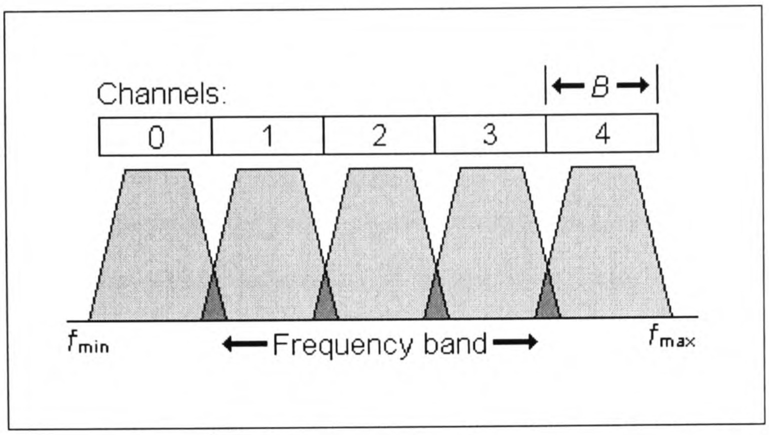

Figure 1.1: Example of the creation of channels from frequencies.

present an example where five channels are created starting from a frequency band $[f_{\min}, f_{\max}]$.

It is customary to ignore the difference in meaning between channels and frequencies just highlighted, and to use the words synonymously.

1.3 Applications of wireless communication

In this section we enumerate some examples of the application of wireless communication. As the reader can see, some of the most common technologies of our time are realised through wireless communications.

1.3.1 Radio and television broadcasting

The most common way to transmit radio and television signals is through the air. The antennae transmit in a radial way (i.e. the signal is sent omnidirectionally), and the information sent is not dedicated to a single user, but to all who wish to receive. Signals transmitted in an area must not interfere with each other and, because of the radial transmission, this can be realised by using each channel only once in the same area.

1.3.2 Mobile telephone networks

The most important application of wireless communication in the last decade has been in mobile phone networks. It is possible to say that the whole of society has been influenced by this innovation. The first commercial mobile phone service was introduced in 1946 by AT&T, but it was very limited in performance. For the first phone network with (relatively) large penetration we have to wait until the 1980's, when AT&T and Motorola Inc. introduced the *Advanced Mobile Phone System* (*AMPS*). In 1992 the *Groupe Speciale Mobile* (*GSM*), a group of European government-owned telephone companies, presented a new digital-based mobile communication standard, called *GSM*. This system has been very successful, not only in Europe but all around the world. A new standard, the *Universal Mobile Telecommunications System* (*UMTS*), is being developed and in the future it should replace *GSM*.

A *GSM* mobile phone network is characterised by a number of base stations, each one including some antennae. Every antenna covers a part of the geographic area to be served, that is consequently divided into smaller areas, called cells. Each antenna operates on a certain frequency. A mobile phone within a cell is connected (on request) with the respective base station via one of the frequencies of the antennae operating on that cell. As a mobile phone proceeds from one cell to another during a call, the frequency should switch to one of those of the new cell without the user realising this. This process is known as *handover*. To aid handover, close cells must be slightly overlapped. Interference can arise when two antennae covering close cells operate on frequencies too close to each other. For a detailed description of *GSM* networks we refer the interested reader to Manni [73], Scourias [85], Kuruppilai et al. [64], Eisenblätter et al. [39] and Grötschel [46].

1.3.3 Fixed wireless telecommunication networks

The cost for a wireless network undercuts, in particular circumstances, the cost for a conventional wired network, making the use of fixed wireless networks economic (see Smith et al. [94]). For this reason, in the near future fixed wireless networks will be adopted to provide fast access for Internet, telephony, video conferencing and home working, which are probably going to become some of the most important applications of telecommunication in the everyday-life of the next years.

The main difference between fixed and mobile wireless networks is that in the first case not only the positions of transmitters are known, as for mobile networks, but also the receivers are fixed. Because of this characteristic the signals of a fixed wireless network are directed, and consequently each receiver must be almost in line of sight of its transmitter.

We refer the interested reader to Allen et al. [6] for a more accurate description of fixed wireless networks and their design.

1.3.4 Military wireless networks

To guarantee communication during military operations, wireless connections have to be dynamically established. This is usually achieved using pairs of transceivers realising radio connections. In this case the interference, which is generally caused by the use of similar frequencies within the same area, has a higher level of tolerance than the other types of wireless networks. The reason for this is that a military network has generally to be engineered and realised very quickly, and anyway it is only temporary.

The interested reader can consult Bradbeer [24] or Williams [105] for a more detailed description of the features of a military wireless network.

1.4 Mathematical modelling of the frequency assignment problem

The frequency assignment problem can be modelled in many ways. In this section we briefly describe the two most famous models, the second of which can be seen as a higher level abstraction of the first.

1.4.1 Multiple interference model

Modelling reality through mathematics implies producing an abstraction of reality. In the case of wireless networks, the first problem to face is how to represent *signal propagation.* There are many models available, each one with particular physical justifications and a certain level of approximation to reality. We can refer the reader interested in a deeper study to Hall et al. [48].

Once a propagation model has been chosen, the natural way to proceed is to define an *attenuation factor* for adjacent channel interference, that measures the decrease of interference as the separation between the frequencies assigned to two transmitters increases (see, for example, Watkins et al. [102]). The next step is to select a set of *reception test points* in the area covered by the network. Given a frequency assignment, it is possible to evaluate at these points the so called *Signal to Interference Ratio* (*SIR*), a ratio between the strength of the wanted transmitter's signal and the strength of the interfering signals. In each network there is a parameter defining the level of tolerable

SIR. When the SIR is over this threshold for a receiver, it is able to recover the desired signal. When the SIR is under the threshold the interference is classified as unacceptable because the desired signal could be lost.

The general *Multiple Interference Model* (Watkins et al. [102], Smith et al. [88] and [89], Montemanni [76] and Montemanni et al. [79][1]) is based on the factors above. Slightly different multiple interference models can be found in Letschy et al. [66], Fischetti et al. [42], Capone and Trubian [26] and Mannino and Sassano [74].

The main problem of this way of modelling interference is in its intrinsic computational complexity. For this reason simpler modelling approaches have been derived, the most famous of which is the binary constraints model[2], described in the next section.

1.4.2 Binary constraints model

The *Binary Constraints Model* can be seen as a generalisation of the well-known *Graph Colouring Problem*, for a description of which the interested reader can refer to Jensen and Toft [57] or to Morgenstern [82]. The frequency assignment problem is modelled by an undirected graph, in which there is a vertex for each transmitter and an edge between each pair of potentially interfering transmitters. Every edge has a label indicating the separation required between frequencies assigned to the two respective transmitters, in order to have acceptable interference.

The binary constraints model is the most common in the literature, and it

[1] The work described in this paper is part of the PhD project of the author of this thesis.

[2] The binary constraints model can be naturally derived from the multiple interference model we have described (see Watkins et al. [102], Montemanni [76] and Bater [12]). For this reason it can be seen as a higher level of abstraction.

has been adopted since the beginning of optimisation in frequency assignment (Hale [47], Leung [67], Cozzens and Roberts [30] and Cozzens and Wang [31]). For a more detailed review of the theories developed until now on this model, the reader can refer to Murphey et al. [83].

Using the binary constraints approach, only interference involving pairs of transmitters is considered. This implies that cumulative interference (i.e. interference generated from more than one transmitter at a time) is simply lost. This, together with the highly digital logic of the model (a constraint can only be in two states: satisfied or violated), is the main drawback of this approach.

Notwithstanding that there are some papers which demonstrate the weakness of the binary constraints model (see Dunkin and Allen [36], Dunkin and Jeavons [38], Dunkin et al. [37], Jeavons et al. [56], Bater [12] and Bater et al. [13], [14] and [15]), it is the most common in practise, especially because the loss of precision in the representation of reality is often justified by the intrinsically lower computational complexity of the model. This makes it possible to address problems of realistic dimensions, which otherwise could not be handled.

In this thesis we will adopt the binary constraints model and we will describe it more formally in Chapter 2.

1.4.3 Other models

Many other models to describe the *FAP* have been presented in the literature. For example in the EUCLID CALMA project[3], a binary model with some extra features is adopted.

[3] EUropean Cooperation for the Long term In Defence Combinatorial ALgorithms for Military Application project, http://www.inra.fr/bia/T/schiex/Doc/CELARE.html.

Another common modification of the binary constraints model is the so called *Cellular Model*, in which every vertex of G requires more than one frequency and a certain separation must exist among the frequencies assigned to the same vertex. The most famous set of benchmark problems for frequency assignment, named the *Philadelphia Dataset* (created on a problem originally proposed by Anderson [10] in 1973, and adopted, for example, in Smith et al. [91] and in Allen et al. [9]) has this feature.

There are also modifications of the binary constraints model where the penalty paid depends on how much a constraint is violated (Eisenblätter [41]). This idea is reasonable because higher violations will logically generate higher interference and consequently higher penalties in the mathematical model.

A different model, the so called *Hypergraph Model*, represents sets of transmitters that cannot assume the same frequency simultaneously through hyper-edges. See Sarkar and Sivarajan [84], Bater [12] and Hurley et al. [53] for more details.

Models considering more physical factors that can generate interference, such as intermodulation products and spurious emissions and responses, have been proposed in Loxton [68] and Smith et al. [93].

1.5 Types of Frequency Assignment Problem

As seen in the previous section, the frequency assignment problem can be modelled in many ways. Once one of these models is adopted, there are many possible optimisation targets. In this section we present some of those studied in the literature, briefly describing some of the results obtained for them so far. The reader interested in a more extensive explanation of these problems and related results can refer to the material concerning the EUCLID CALMA project (especially Aardal et al. [1], [3] and Tiourine et al. [96]).

1.5.1 Minimum Order - Frequency Assignment Problem (*MO-FAP*)

The aim of the *Minimum Order - Frequency Assignment Problem* (*MO-FAP*) is to assign frequencies to the transmitters so that no unacceptable interference occurs and at the same time the number of different frequencies used in the assignment plan (*order* of the assignment) is minimised.

This formulation of the problem is the oldest one and is probably now of little use in practise. In contrast to the situation of a couple of decades ago, the spectrum is now divided and allocated by the regulatory body[4] in contiguous parts (or *bands*) and a network operator will receive one or more of these bands. Neither the authority nor the operators are interested in the number of used (or potentially used) frequencies within these bands, they are more concerned with the dimensions of the required bands.

In the literature it is possible to find good lower bounds for this type of problem, for example, in Hurkens and Tiourine [51] or Aardal et al. [2] (where an efficient branch & bound assignment algorithm is also described). A description of some different heuristic approaches studied within the EUCLID CALMA project is given in Hurkens and Tiourine [51] and Tiourine et al. [96]. Bouju et al. [21] and [22] present a tabu search algorithm and a GEneral NETwork (GENET) algorithm, while Warners et al. [100], [101] describe a potential reduction approach to the problem.

[4] For example in the UK the regulatory body is the Radiocommunications Agency, http://www.radio.gov.uk.

1.5.2 Minimum Span - Frequency Assignment Problem (*MS-FAP*)

In the *Minimum Span - Frequency Assignment Problem* (*MS-FAP*) the aim is, given a threshold for the acceptable interference, to assign frequencies to the transmitters while minimising the *span*, which is defined to be the difference between the maximum and the minimum used frequencies.

This type of problem is considered very important today, especially because of its use when an operator has to establish a network and it asks the authority for a part of the spectrum. Both the operator and the authority must have an approximation of the size of the required interval of spectrum for a fixed level of quality. This is exactly the information supplied by the span.

A lot of material on this problem is available in the literature. In the field of lower bounds some good results are presented in Smith and Hurley [90], Smith et al. [86] and Allen et al. [8] and [9]. Some other lower bounds are described in Tcha et al. [95] and in Janssen and Wentzell [54]. In Hurkens and Tiourine [51] and in Aardal et al. [2], the adaptations of the methods described for the *MO-FAP* are presented. An exact algorithm which produces a lower bound if the computation is stopped before the natural end is described in Avenali et al. [11].

Concerning heuristic algorithms, the most complete work available is FA-SOFT, a powerful system developed in a collaboration among the Radiocommunications Agency, the University of Glamorgan and the University of Wales Cardiff, by Hurley, Smith and Thiel (see [52] and [92]). In this system an assorted collection of heuristic approaches is implemented and used together with some of the lower bounds of the same authors. Some other heuristic approaches (studied in the EUCLID CALMA project) are described in Hurkens and Tiourine [51], in Tiourine et al. [96] and in Aardal et al. [1], [3]. Other methods are due to Costa [29], de Werra and Gay [32], Battiti et al. [16], Hao et al. [49], Dorne and Hao [34] and [35] and Allen et al. [5]. A multi-agent algorithm is described in Abril et al. [4].

1.5.3 Fixed Spectrum - Frequency Assignment Problem (*FS-FAP*)

As the name suggests, in the case of the *Fixed Spectrum - Frequency Assignment Problem* (*FS-FAP*) there is a predefined set of available frequencies for each transmitter and the target is to minimise a measure of the level of interference present in the system. In general it will not necessarily be zero.

This problem is particularly useful for operators when they design real networks, after having obtained an allocation of spectrum.

In this thesis we will focus our attention on this type of frequency assignment problem. We have made this choice essentially because *FS-FAP* is less studied among the problems presented in this section. This lack of research is probably due to the fact that the idea of explicitly accepting interference is relatively recent, being connected with increasing prices for spectrum in the last few years. In particular the lower bound theory is quite poor: good lower bounds have mainly been developed only for problems with particular features.

As well as the lack of a general lower bound technique, there are also no general exact algorithms. A number of heuristic algorithms have been proposed, but without lower bounds it is impossible to evaluate their performance. It is possible to compare them but not to state anything about whether they work well in absolute terms.

In the next chapters we will present a more detailed literature review.

1.5.4 Other optimisation targets

The reader interested in other, quite uncommon, optimisation targets, can refer to Borgne [17] or, for a review, to Koster's PhD thesis [60], Jaumard et al. [55] and *FAP web*[5], a website entirely devoted to frequency assignment problems.

1.6 Outline of the thesis

The thesis is organised as follows.

After the brief introduction to frequency assignment given in the previous sections, in Chapter 2 we formalise the binary constraints model for the fixed spectrum problem. We present a graph theoretical model and some integer programming formulations.

In Chapter 3 we propose two approximate techniques for solving the *FS-FAP*. In particular we describe the adaptation to the *FS-FAP* and the implementation of two well-known metaheuristic algorithms. A description of the benchmarks adopted in this work is also given.

Chapter 4 is dedicated to lower bounds. We propose two different approaches, the most promising of which obtain the estimates (*global* lower bounds) by solving the linear relaxation of one of the integer programs described in Chapter 2, reinforced with inequalities obtained from clique-like subproblems of the original problem. Each inequality is based on a lower bound for the penalty paid (or number of constraint violations present) in a clique-like subproblem (*local* lower bounds).

In Chapter 5 we describe some improvement techniques for the local lower bounds. These techniques are based on inequalities derived from some struc-

[5] FAP web - A website about Frequency Assignment Problems, http://fap.zib.de.

tural characteristics of the problems. These inequalities are used to reinforce the linear programs which are solved to obtain the local lower bounds.

In Chapter 6 some improvements for the most promising of the global lower bounding techniques described in Chapter 4 are presented. In particular we propose a simplification for the linear program on which the bounds are based. This simplification permits shortening of the computation times. A family of reinforcing inequalities, again derived from some structural information of the problems, is also described. These inequalities sometimes allow better estimates to be obtained.

Chapter 7 presents a discussion of the success of the upper and lower bounding techniques and presents suggestions for future work.

Chapter 2

Fixed Spectrum - Frequency Assignment Problem (*FS-FAP*)

In this chapter we formally define the binary constraints model for the fixed spectrum frequency assignment problem, which we will adopt in the remainder of this thesis.

We describe a graph theoretical representation of the *FS-FAP* (see Diestel [33] for an introduction to graph theory) and some mathematical programming formulations to represent it (see Williams [104] for an introduction to mathematical programming). A brief study of the dimensions of these formulations is also included.

2.1 Graph theoretical representation

Adopting the binary constraints model, the *FS-FAP* can be represented by a graph. Formally it can be defined as a 5-tuple $\{V, E, D, P, F\}$ where the elements have the following meaning:

- V: vertex set of an undirected graph G. Every vertex represents a transmitter of the wireless network;

- E: set of edges of the undirected graph G. Edges represent those pairs of transmitters that are constrained. Edges will be written here as $\{v, w\}$, with $v < w$;

- D: set of labels. There is a mapping $E \rightarrow D$ such that $\{v, w\} \mapsto d_{vw}$, with $d_{vw} \in \mathcal{N}_0^+$. d_{vw} is the highest separation between the frequency assigned to the transmitter v and the one assigned to w that may cause the generation of unacceptable interference. If we denote by $f(v)$ the frequency assigned to transmitter v, then if $|f(v) - f(w)| > d_{vw}$, the interference involving the two transmitters is acceptable;

- P: set of labels. There is a mapping $E \rightarrow P$ such that $\{v, w\} \mapsto p_{vw}$, with $p_{vw} \in \mathcal{N}^+$. p_{vw} is a cost to be paid if the separation between the frequencies of transmitters v and w is less than or equal to d_{vw};

- F: set of consecutive frequencies available for every vertex (transmitter) in V.

The objective of the fixed spectrum frequency assignment problem is to find an assignment which minimises the sum of p_{vw} over all pairs $\{v, w\} \in E$ for which $|f(v) - f(w)| \leq d_{vw}$.

A pictorial representation of a graph associated with the model described above is given in Figure 2.1.

2.2 Mathematical formulations

In this section we describe some integer programming formulations of the *FS-FAP*.

A study of the size of these formulations is also presented.

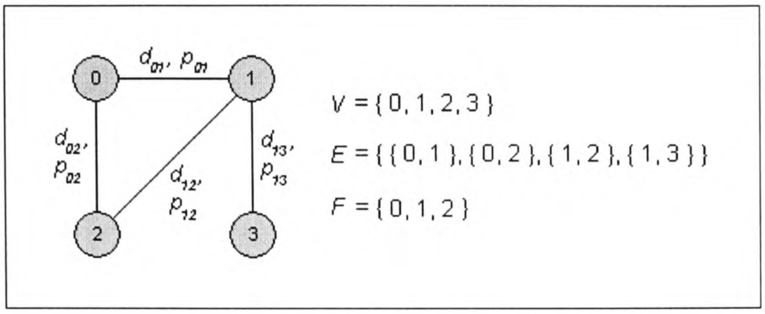

Figure 2.1: Example of the binary constraints model.

2.2.1 Mathematical formulation $FAP1$

This model has been proposed by Koster et al. in [61] (see also [60], [62] and [63]) and it has been developed in the context of the benchmarks adopted within the EUCLID CALMA project. As noted before (see Section 1.4.3) these problems have features that are not present in our model and consequently the version of the formulation presented here is a simplification of the original.

To describe formulation $FAP1$ we need the following definitions:

- x_{viwj}: {0,1} variable defined for every $\{v, w\} \in E$ and for every $i, j \in F$. It is 1 when vertex v is assigned frequency i and at the same time vertex w is assigned frequency j; 0 otherwise;

- y_{vi}: {0,1} variable defined for every $v \in V$ and for every $i \in F$. It is 1 when v is assigned frequency i; 0 otherwise.

$$(FAP1)\ \text{Min} \sum_{\{v,w\}\in E} \sum_{i=0}^{|F|-1} \sum_{\substack{j=0;\\ |i-j|\le d_{vw}}}^{|F|-1} p_{vw} x_{viwj} \tag{2.1}$$

$$\text{s.t.} \sum_{i=0}^{|F|-1} y_{vi} = 1 \qquad \forall v \in V \tag{2.2}$$

$$\sum_{j=0}^{|F|-1} x_{viwj} = y_{vi} \qquad \forall\{v,w\} \in E;\ i = 0,\dots,|F|-1 \tag{2.3}$$

$$\sum_{j=0}^{|F|-1} x_{wjvi} = y_{vi} \qquad \forall\{w,v\} \in E;\ i = 0,\dots,|F|-1 \tag{2.4}$$

$$x_{viwj} \in \{0,1\} \qquad \forall\{v,w\} \in E;\ i,j = 0,\dots,|F|-1 \tag{2.5}$$

$$y_{vi} \in \{0,1\} \qquad \forall v \in V;\ i = 0,\dots,|F|-1 \tag{2.6}$$

The target, minimising the global penalty to be paid, is expressed by (2.1); equations (2.2) hold because exactly one frequency is assigned to each transmitter. Equations (2.3) and (2.4) have been inserted to maintain consistency between the values of the x and y variables; finally set inclusions (2.5) define the domain for the x's and set inclusions (2.6) state that y's must be in $\{0,1\}$.

2.2.2 Mathematical formulation $FAP2$

The formulation described in this section is a modification of the one presented in Aardal et al. [2] within the EUCLID CALMA project.

To describe $FAP2$ we need the following definitions:

- z_{vw}: {0,1} variable defined for every $\{v,w\} \in E$. It is 1 when there is interference involving transmitter v and transmitter w (i.e. $|f(v) - f(w)| \le d_{vw}$);

- x_{vi}: {0,1} variable defined for every $v \in V$ and for every $i \in F$. It is 1 when transmitter v is assigned to frequency i.

$$(FAP2)\quad \text{Min} \sum_{\{v,w\}\in E} p_{vw} z_{vw} \tag{2.7}$$

$$\text{s.t.} \sum_{i=0}^{|F|-1} x_{vi} = 1 \qquad \forall v \in V \tag{2.8}$$

$$x_{vi} + x_{wj} \leq 1 + z_{vw} \qquad \begin{array}{l}\forall\{v,w\} \in E;\ i,j = 0,\ldots,|F|-1; \\ |i-j| \leq d_{vw}\end{array} \tag{2.9}$$

$$z_{vw} \in \{0,1\} \qquad \forall\{v,w\} \in E \tag{2.10}$$

$$x_{vi} \in \{0,1\} \qquad \forall v \in V;\ i = 0,\ldots,|F|-1 \tag{2.11}$$

Here (2.7) expresses the target: minimising the penalty to be paid; equations (2.8) assert that every transmitter has to be assigned to exactly one frequency and inequalities (2.9) activate the z variables in the case of interference. Set inclusions (2.10) and (2.11) state that the z's and the x's have to be boolean variables.

2.2.3 Mathematical formulation $FAP3$

This formulation is a simplification of the one proposed (but not studied) near the end of Koster's PhD thesis [60], where it is stated to be a refinement and extension of the *Orientation Model* (Borndörfer et al. [20]).

To describe $FAP3$ we need the following definitions:

- y_v: integer variable defined for every $v \in V$. It represents the frequency assigned to transmitter v;
- x^1_{vw}: {0,1} variable defined for every $\{v,w\} \in E$. It is 1 when $|y_v - y_w| \leq d_{vw}$;

  0 otherwise;

- x^0_{vw}: {0,1} variable defined for every $\{v,w\} \in E$. It is 1 when $y_v - y_w > d_{vw}$; 0 otherwise;
- x^2_{vw}: {0,1} variable defined for every $\{v,w\} \in E$. It is 1 when $y_w - y_v > d_{vw}$; 0 otherwise.

$$(FAP3) \quad \text{Min} \sum_{\{v,w\}\in E} p_{vw} x^1_{vw} \tag{2.12}$$

$$\text{s.t. } x^0_{vw} + x^1_{vw} + x^2_{vw} = 1 \qquad \forall\{v,w\} \in E \tag{2.13}$$

$$y_v - y_w \geq (d_{vw}+1)x^0_{vw} - d_{vw}x^1_{vw} - (|F|-1)x^2_{vw} \quad \forall\{v,w\} \in E \tag{2.14}$$

$$y_v - y_w \leq (|F|-1)x^0_{vw} + d_{vw}x^1_{vw} - (d_{vw}+1)x^2_{vw} \quad \forall\{v,w\} \in E \tag{2.15}$$

$$y_v \in \{0,1,\ldots,|F|-1\} \qquad \forall v \in V \tag{2.16}$$

$$x^0_{vw} \in \{0,1\} \qquad \forall\{v,w\} \in E \tag{2.17}$$

$$x^1_{vw} \in \{0,1\} \qquad \forall\{v,w\} \in E \tag{2.18}$$

$$x^2_{vw} \in \{0,1\} \qquad \forall\{v,w\} \in E \tag{2.19}$$

Here the target, minimising the global penalty to be paid, is expressed by (2.12); equations (2.13) are introduced to force exactly one among x^0_{vw}, x^1_{vw} and x^2_{vw} to be 1 for every $\{v,w\} \in E$; inequalities (2.14) and (2.15) are introduced to maintain consistency between the values of the x and y variables; set inclusions (2.16) are inserted to define the permitted values for the y's; finally set inclusions (2.17), (2.18) and (2.19) fix the domains for all the x's.

2.2.4 Dimensions of formulations

In this section we estimate, given the problem characteristics (i.e. the number of vertices, the number of edges and the frequency domain dimension), the number of constraints and variables of each formulation presented in the

Formulation	No. of Constraints	No. of Variables
$FAP1$	$\|V\| + 2\|E\|\|F\|$	$\|V\|\|F\| + \|E\|\|F\|^2$
$FAP2$	$\|V\| + \|E\| O(\|F\|)$	$\|V\|\|F\| + \|E\|$
$FAP3$	$3\|E\|$	$\|V\| + 3\|E\|$

Table 2.1: Dimensions of the formulations.

previous sections. We group these results in Table 2.1, where the columns have the following meaning:

- Formulation: names of the three formulations studied in the previous sections;
- No. of Constraints: expressions for the number of constraints of the formulations (number of rows of the problem matrix). Constraints defining variable domains are not counted here;
- No. of Variables: expressions for the number of variables of the formulations (number of columns of the problem matrix).

Analysing Table 2.1, it is possible to have an idea of how the formulations studied modify their dimensions when the characteristics of the problem modelled change. It can be seen that the number of constraints of $FAP1$ and $FAP2$ is dominated by the product between $|E|$ (number of edges of the graph representing the problem) and a quantity proportional to $|F|$ (number of frequencies available)[1]. This result is not encouraging, because it suggests that for most of the problems the number of constraints will be too large to be handled. For the other formulation, $FAP3$, we have $3|E|$ constraints. This

[1] In the estimate for the number of constraints of $FAP2$, $\max_{\{v,w\}\in E}\{d_{vw}\}$ has been considered as limited by a problem-independent constant. This is generally true in reality.

means that $FAP3$ should have a reasonably small number of constraints, especially on sparse problems.

Analysing the third column of Table 2.1 we can see how $FAP1$ has a number of variables proportional to $|V|^2\,|F|^2$, a value that could become much too large even for relatively small problems. Formulation $FAP2$ appears to be the one with the smallest number of variables when problems have a small fixed span, while $FAP3$ should be the smallest one in the case of problems with larger span and which are not too dense.

Concluding the examination of Table 2.1, we can state that $FAP3$ is the most promising formulation from a dimensional point of view. It is probably the only one of the formulations described here which can be used to deal with real problems.

Chapter 3

Upper Bounds

This chapter is dedicated to heuristic algorithms for the *FS-FAP*.

After a brief review of the methods described in the literature, we present an adaptation to the *FS-FAP* of two well-known approximation algorithms. In particular, we describe the first application (as far as we are aware) of a modified version of the original tabu search paradigm to the *FS-FAP*.

The chapter is concluded with some computational tests.

Some of the results presented in this chapter can be found in Montemanni and Smith [77] and Montemanni et al. [78].

3.1 Literature review

Many heuristic algorithms have been presented in the literature for the *FS-FAP*, or for slightly more complex problems which generalise it. In this section we briefly describe some of these approaches, without attempting to be comprehensive. More detailed reviews can be found in Koster [60], Smith et al. [87], Eisenblätter et al. [39] and, for the algorithms developed within the EUCLID CALMA project, in Hurkens and Tiourine [51], Tiourine et al.

[96] and Aardal et al. [3].

Borndörfer et al. [18], [19] and Eisenblätter [40] present some *constructive algorithms* (i.e. algorithms which construct a solution step by step starting from an empty one) together with some basic *local searches* (i.e. algorithms which move from solution to solution, searching for the best possible one). Other notable local search approaches are presented in Koster [60] and Montemanni [75], where a set of constructive algorithms is also proposed. An interior point method applied to a quadratic formulation of *FS-FAP* is described in Warners [100] and in Warners et al. [101].

In the field of *metaheuristic algorithms* (i.e. iterative methods which, using particular strategies, drive a subordinate heuristic algorithm to explore the search space in an intelligent way), an Approximate Non deterministic Tree Search (ANTS) algorithm is presented in Maniezzo and Carbonaro [70], Maniezzo et al. [71] and Montemanni [75]. Some methods based on the genetic algorithm paradigm are proposed in Kolen [59] and in Lau and Tsang [65]. Boyce et al. [23] present a GEneral NETwork (GENET) algorithm together with a tabu search algorithm. Finally, a Guided Local Search (GLS) algorithm is presented in Voudouris [98] and Voudouris and Tsang [99].

In Whitaker et al. [103] a tabu search algorithm developed to deal with binary and non binary constraints (i.e. constraints involving more than two transmitters at a time), called NBS (non binary solver), is described.

Some authors have focussed their attention on unweighted problems ($\forall\{v,w\}\ p_{vw}=1$), where the target is to minimise the number of constraint violations. Some of the most interesting methods working on this model are briefly summarised in the following paragraph.

Two approaches to unweighted problems based on the tabu search paradigm are described in Castellino et al. [27] and Hao et al. [49] respectively (see also Dorne and Hao [35]), while an evolutionary algorithm is presented in Dorne and Hao [34]. A different approach to the problem, based on a randomised algorithm is described in Žerovnik [106]. The most complete collection of heuristic methods for unweighted problems is provided by FASOFT (Hurley et al. [52]), where assorted constructive algorithms, local searches and metaheuristic algorithms are implemented.

3.2 The simulated annealing algorithm

In this section we describe the general framework of the simulated annealing algorithm and an implementation for the *FS-FAP*.

3.2.1 General description

Simulated annealing is a metaheuristic algorithm derived from thermodynamic principles. It has been applied originally to combinatorial optimisation in Kirkpatrick et al. [58]. It can be used to find (near) minimum cost solutions[1] of difficult problems characterised by vast search spaces, on which it is impossible to obtain the optimal solution by running exact algorithms.

The search proceeds with the cost function reducing most of the time, but it is allowed to increase sometimes to permit escape from local minima which are not global minima. The analogy with thermodynamics, and specifically with the way that liquids freeze and crystalise, or metals cool and anneal, is in the strategy adopted to accept or not accept cost-increasing solutions. At high temperatures, the molecules of a liquid move freely with respect to one another. If a liquid metal is cooled quickly (i.e. quenched), it does not reach

[1] Here and in the following we suppose the methods to be applied to minimisation problems. It is trivial to adapt the descriptions for the maximisation case.

a minimum energy state but a somewhat higher energy state corresponding, in the mathematical sense, to a suboptimal solution. On the other hand, if the liquid is cooled slowly, thermal mobility is restricted. The atoms are often able to line themselves up and form a pure crystal that is completely regular. The crystal is the state of minimum energy for the system, which corresponds to the optimal solution in a mathematical optimisation problem. The algorithm is based on the connection of the physical concept of temperature with the mathematical concept of the probability of accepting a cost-increasing solution. The probability will be high initially and will decrease slowly, like the temperature in the annealing process which produces the regular crystal.

The main elements of the algorithm are:

- Solution representation: each feasible solution of the optimisation problem must have a unique representation within the search space;
- Cost function: a function *Cost* mapping each feasible solution into a value representing its cost (analogous to the energy of the system in the thermodynamic case). The goal of the algorithm is to find a solution which minimises the cost;
- Neighbourhood: a function mapping each feasible solution S into a set of other solutions. Each time the algorithm has to consider a new solution it is chosen randomly among those in the neighbourhood of the current solution;
- Temperature t: a control parameter analogous to the temperature in the physical annealing process. It starts with an high value and decreases during the computation. The parameter is used to decide whether or not to accept a new solution;

- Annealing schedule: this indicates how t is lowered from high values to low values during the running of the algorithm. It simulates the physical process of cooling.
- Termination criterion: the algorithm stops when the termination criterion is satisfied. Generally the criterion is represented by a minimum value for the temperature or by a maximum number of consecutive iterations carried out without accepting solutions with cost higher than the actual one.

At each iteration the algorithm selects a new solution S_N from the neighbourhood of the current solution S_O. If the cost of the new solution is less than or equal to the cost of the old one (i.e. $Cost(S_N) \leq Cost(S_O)$) then the new configuration is accepted and becomes the new current solution. If $Cost(S_N) > Cost(S_O)$ then the new configuration may still be accepted, with probability given by:

$$\min\left\{1, e^{-\frac{Cost(S_N)-Cost(S_O)}{t}}\right\} \tag{3.1}$$

This general scheme, which always takes a downhill step and sometimes takes an uphill step, is known as the *Metropolis algorithm.* The simulated annealing procedure described by Kirkpatrick et al. [58] uses the Metropolis algorithm and varies the temperature parameter t during the computation. In the beginning t is high and most of the new configurations are accepted; as the algorithm proceeds t reduces until it reaches a value where non improving configurations are all rejected.

The pseudocode for the general simulated annealing algorithm is presented in Figure 3.1. t_{init} is a parameter representing the initial temperature. I_{sa} is the number of iterations carried out by the algorithm at each temperature value.

```
*SimulatedAnnealing (Pr)*

**INPUT:**
$Pr$ = optimisation problem.
**OUTPUT:**
a solution of $Pr$.

$S_O$ := randomly generated feasible solution of $Pr$;
$Best$ := $S_O$;
$t$ := initial temperature $t_{init}$;
While(termination criterion not met)
    For $k$ := 0 to $I_{sa}$
        $S_N$ := randomly chosen solution in the neighbourhood of $S_O$;
        If(random number in $[0,1) < e^{-\frac{Cost(S_N)-Cost(S_O)}{t}}$)
            $S_O$ := $S_N$;
            If($Cost(S_N) < Cost(Best)$)
                $Best$ := $S_N$;
            EndIf
        EndIf
    EndFor
    reduce $t$;
EndWhile
Return $Best$;
```

Figure 3.1: Simulated annealing algorithm.

3.2.2 A simulated annealing algorithm for the *FS-FAP*

Our adaptation of the general simulated annealing schema to the *FS-FAP* is presented in this section. For the description we refer to the notation introduced in Section 2.1.

3.2.2.1 Solution representation

A frequency assignment S is represented as a list $\langle f_S(0), f_S(1), \ldots, f_S(|V|-1)\rangle$ where element $f_S(v)$ contains the frequency assigned to transmitter v.

3.2.2.2 Cost function

The cost function *Cost* maps an assignment into the sum of the penalties paid in it. Formally we have:

$$Cost(S) = \sum_{\substack{\{v,w\} \in E; \\ |f_S(v) - f_S(w)| \leq d_{vw}}} p_{vw} \tag{3.2}$$

3.2.2.3 Neighbourhood

Given a current solution S_O, another assignment S_N is in its neighbourhood if S_N differs from S_O for the frequency assigned to exactly one *violating transmitter* of S_O, where a violating transmitter is defined as a transmitter involved in at least one constraint violated in S_O. Formally, if we define $V^V_{S_O}$ as the set of violating transmitters in the assignment S_O, S_N is a neighbour of S_O if $\exists\, v \in V^V_{S_O} \mid f_{S_O}(v) \neq f_{S_N}(v)$ and $\forall w \in V, w \neq v\ f_{S_O}(w) = f_{S_N}(w)$.

The neighbourhood described above is often referred to as a *single move full violating neighbourhood* in the literature (see, for example, Hurley et al. [52]).

We have also experimented with a bigger neighbourhood, the so called *double move full violating neighbourhood.* Intuitively this considers as a neighbour of S_O each solution which differs from S_O in the frequencies assigned to exactly two transmitters, of which at least one must be a violating transmitter. The use of this more complex neighbourhood slightly slowed down the algorithm without giving any improvement. For this reason we did not explore it further.

3.2.2.4 Temperature t

In the simulated annealing implementation we propose, the initial temperature t_{init} is a user defined parameter and t decreases from this initial value according to the annealing schedule.

3.2.2.5 Annealing schedule

The annealing schedule we adopt is known as *geometric cooling* in the literature (see, for example, Hurley et al. [52]). Every I_{sa} iterations we reduce t using the following formula:

$$t := \alpha t \tag{3.3}$$

where $0 < \alpha < 1$ is a user defined parameter.

The annealing schedule described is very simple. More complex schemes have been tested, but they did not appear to improve the results of the algorithm.

3.2.2.6 Termination criterion

The algorithm stops when I_{max} consecutive iterations are carried out without accepting a solution with a cost higher than that of the current solution. I_{max} must be a very large number to prevent the phenomenon of interrupting the search process when a long sequence of consequent improving moves are carried out.

3.2.2.7 Implementation details

The implementation technique we adopt is inspired by the one described in Hao et al. [49]. We maintain a table of dimension $|V| \times |F|$, called the *cost change table*, where position (v, f) contains the cost of the solution obtained by changing to f the frequency currently assigned to transmitter v (if f is the frequency currently assigned to v, then the value contained in the entry (v, f) is the cost of the current solution). Each time a move is carried out, the elements of the table affected by the move are updated accordingly. For each transmitter v we also adopt a list containing its adjacent transmitters (transmitters involved in at least one constraint with v). This list is used to speed up the table updating process. Only the transmitters which are adjacent to the one modified are involved in the updating process. Only the positions of the table corresponding to the frequencies which interfere with the old or the new frequency of the reassigned transmitter are modified. An example of the use of the cost change table is given in Figure 3.2. An interval of columns of the rows corresponding to four transmitters, u, v, w and z, appears in the figure. In the graph representing the problem, u and v are not connected, u and w are connected with an edge with separation $d_{uw} = 2$ and u and z are connected with an edge with separation $d_{uz} = 0$. In the figure we depict the effects which derive from the modification of the frequency assigned to transmitter u from j (Old) to i (New). We have highlighted the table entries the values of which are modified because of the frequency reassignment. In blue we have indicated the entries which increase their values (of p_{uw} and p_{uz} for w and z respectively) and in yellow the entries which have their values decreased. Notice that the entry (w, k) (green) is not modified. As transmitter v is not connected with transmitter u, its row is not affected by the frequency reassignment.

Initialising the table has a computational complexity of $O(|V|^2|F|)$. After each move, the matrix can be updated with a theoretical complexity of $O(|V||F|)$. In practise the updating process is extremely fast, because the entries of the table which are updated are generally a small subset.

An implementation of the algorithm without the cost change table would have a theoretical complexity of $O(|V|)$ at each iteration (to compute the variation of cost due to the move selected). The implementation we propose,

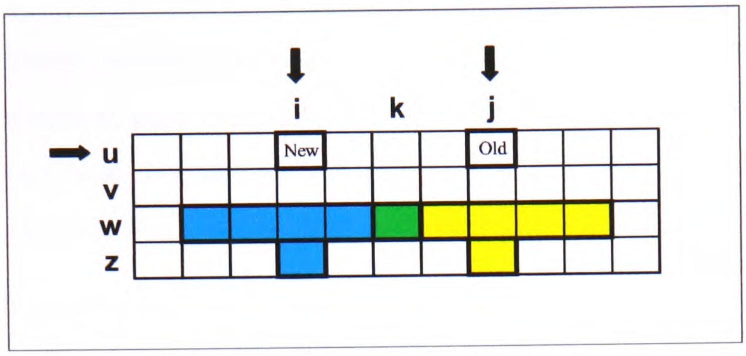

Figure 3.2: Updating the cost change table.

notwithstanding its higher theoretical complexity, appears to be faster (some preliminary tests confirmed this). The reason is that at all of the iterations where the potential move is rejected, no cost calculation of table updating is necessary with this implementation.

3.3 The tabu search algorithm

In this section we describe the general ideas of the tabu search algorithm and an implementation of it we have developed for the *FS-FAP*. This implementation presents some particular features, which are not present in the general schema. These features seem to improve the performance of the tabu search algorithm on the *FS-FAP*s.

3.3.1 General description

Tabu search is a metaheuristic algorithm. It was first suggested by Glover [43] (see also Glover et al. [44]).

The basic idea of the method is to partially explore the search space of all feasible solutions by a sequence of moves. At each iteration, the move carried out is the most promising among those available. A mechanism which forbids a set of moves at each iteration is present, aiming to help the algorithm to escape from local (but not global) minima.

Formally, the main elements of the algorithm are:

- Solution representation: each feasible solution of the optimisation problem must have a unique representation within the search space;

- Cost function: a function *Cost* mapping each feasible solution into a value representing its optimisation cost. The goal of the algorithm is to find a solution which minimises this value;

- Neighbourhood: a function mapping each feasible solution S into a set of other solutions. Each time the algorithm has to consider a new solution, it is chosen from the neighbourhood of the current solution;

- Tabu list: a list containing the last T moves carried out, which for this reason are forbidden. A solution obtained from the current solution S with a move contained in the tabu list, cannot (in general) be a member of the neighbourhood of S;

- Aspiration criterion: if a tabu move (a move which is contained in the tabu list) satisfies this criterion, then the solution obtained by applying it to the current solution S can be considered to be in the neighbourhood of S. The usual criterion is that the move produces the best solution obtained so far.

- Termination criterion: the algorithm stops when the termination criterion is satisfied.

At each iteration the algorithm calculates the neighbourhood of the current assignment. Solutions generated by using a move contained in the tabu list cannot be in the neighbourhood set, unless the respective move satisfies the aspiration criterion. The solution with the minimum cost among those in the neighbourhood becomes the new current solution.

The tabu list, whose dimension strictly depends on the neighbourhood selected, has been inserted to prevent the search becoming trapped in a local minimum, while the aspiration criterion has been introduced to give more flexibility to the algorithm: it makes a move contained in the tabu list feasible in case it would produce a very promising new solution.

In Figure 3.3 the pseudocode of the general tabu search algorithm is presented.

```
TabuSearch(Pr)

INPUT:
$Pr$ = optimisation problem.
OUTPUT:
a solution of $Pr$.

$S$ := randomly generated solution of $Pr$;
$Best$ := $S$;
While(termination criterion not met)
    $S$ := best solution in the neighbourhood of $S^*$;
    If($Cost(S) < Cost(Best)$)
        $Best$ := $S$;
    EndIf
    update tabu list;
EndWhile
Return $Best$;

*the neighbourhood of $S$ does not include solutions obtained using
 those moves which are contained in the tabu list and do not sati-
 sfy the aspiration criterion.
```

Figure 3.3: Tabu search algorithm.

3.3.2 A tabu search algorithm for the *FS-FAP*

In this section we present our adaptation of the tabu search algorithm to the *FS-FAP*. For the description we refer to the notation introduced in Section 2.1.

3.3.2.1 Solution representation

The representation of a frequency assignment S is the same as that described in Section 3.2.2.1 for the simulated annealing algorithm. It is obtained by using a list $\langle f_S(0), f_S(1), \ldots, f_S(|V|-1)\rangle$, where element $f_S(v)$ contains the frequency assigned to transmitter v.

3.3.2.2 Cost function

The function *Cost* is the same as that described for the simulated annealing algorithm in equation (3.2). *Cost* maps each assignment S into the sum of the penalties paid in it.

3.3.2.3 Neighbourhood

An assignment S_N is in the neighbourhood of the current solution S_O if S_N differs from S_O in the frequency assigned to exactly one violating transmitter[2] and the move which produces S_N from S_O is not in the tabu list (no aspiration criterion is used, see Section 3.3.2.5). Defining $V^V_{S_O}$ as the set of violating transmitters in the assignment S_O, S_N is a neighbour of S_O if $\exists\, v \in V^V_{S_O} \mid f_{S_O}(v) \neq f_{S_N}(v)$ and $\forall w \in V, w \neq v\ f_{S_O}(w) = f_{S_N}(w)$ and the move $(v, f_{S_N}(v))$ is not in the tabu list.

[2] Accordingly to the definition given in Section 3.2.2.3, a violating transmitter is a transmitter involved in at least one constraint violated in S_O.

3.3.2.4 Tabu list

The tabu list of our algorithm contains pairs (v, f), where v is a transmitter and f a frequency. Each time a move involving the assignment of frequency f to transmitter v is carried out, we insert (v,f) into the tabu list, where it will remain for approximately T iterations.

Instead of using a tabu list with a fixed length T, as in the original schema, we have decided to dynamically vary T during the running of the algorithm. This choice has been suggested by some tests which indicated the superiority of the dynamic tabu list over the static one (see Section 3.4.3). In particular we have noticed that the best results were achieved by reducing the length of the tabu list in the same way as it is done for the temperature parameter in the simulated annealing algorithm. Every I_{ts} iterations we reduce the length T of the tabu list using the following formula:

$$T := \beta T \tag{3.4}$$

where $0 < \beta < 1$ is a user defined parameter. When T is reduced, the oldest moves which exceed the new length of the list become feasible. The initial value of T, which we will refer to as T_{init}, is defined by the user.

3.3.2.5 Aspiration criterion

We do not use any aspiration criterion in our tabu search algorithm. Some preliminary tests suggested that the use of an aspiration criterion slows down our implementation of the algorithm (because of the extra data structures and their updating) without improving the results. This situation is quite uncommon in the field of combinatorial optimisation, and could be an indication of the peculiarity of the *FS-FAP*.

3.3.2.6 Termination criterion

The algorithm stops when T, the length of the tabu list, becomes smaller than a threshold value T_{min}, which is specified by the user. This termination criterion is quite uncommon for a tabu search algorithm, but it is trivially connected with our strategy of dynamically modifying the parameter T.

3.3.2.7 Implementation details

The implementation technique we adopt is an extension of that described in Section 3.2.2.7 for the simulated annealing algorithm.

In addition to the cost change table (which contains in each position (v, f) the cost of the solution obtained by changing to f the frequency currently assigned to v) there is a second table, again of dimension $|V| \times |F|$, containing in each position (v, f) the last iteration number in which the respective move (assignment of frequency f to transmitter v) has been carried out. This table is used to check in a fast way whether or not a candidate move is tabu: if the iteration number stored in the cell is greater than or equal to the current iteration number minus T, then the move is tabu. We also maintain a list which indicates for each transmitter, the frequency (different from the one currently assigned) that, without generating a tabu move, would produce the lowest cost if assigned to it. This list is used at each iteration to select quickly the best move in the active neighbourhood. After each iteration this list is updated efficiently, by modifying only the entries affected by the last move. We also have a one-dimensional array of length $|T_{init}|$ which contains the conventional tabu list. It is used at each iteration to identify quickly the move which exits from the list (and to update efficiently the data structures affected by this exit). Like for the implementation of the simulated annealing algorithm, for each transmitter we have a list of the adjacent transmitters.

These lists are used to speed up the process of updating the structures.

Initialising the structures has a computational complexity of $O(|V|^2|F|)$. After each move, the tables can be updated with a theoretical complexity of $O(|V||F|)$. In practise the updating process is very fast, but not as fast as for the simulated annealing algorithm, because of the additional overhead of the extra data structures.

3.4 Computational results

In this section we introduce the graphs we will use in the benchmarks adopted in this thesis. We also present the results obtained on these benchmarks by the heuristic algorithms described in this chapter. Section 3.4.3 is dedicated to a study on the effectiveness of the dynamic length tabu list within the tabu search algorithm, while the chapter is closed with a comparison of our algorithms with some other methods presented in the literature.

3.4.1 Description of graphs

In this section we describe the characteristics of the graphs on which the benchmarks of this thesis are based. It is interesting to notice that we will often undertake different tests on the same graph, changing the size of the spectrum available. For this reason there are more benchmarks than the number of graphs we describe in Section 3.4.1.1, Section 3.4.1.2 and Section 3.4.1.3.

3.4.1.1 Graph set 1

The weighted graphs of the first set were originally minimum span problems. We have converted them into the *FS-FAP* format by fixing $\forall\{v,w\}\ p_{vw} = 1$.

Specifically, the scenarios of this first set are created from different families of minimum span problems:

- AC-x-y: scenarios derived from a binary constraint representation of area coverage problems (see Watkins et al. [102]). x is the number of transmitters and y the required *SIR* (signal to interference ratio);
- GSM-x: realistic *GSM* scenarios. x is the number of transmitters in the network;
- Test$x$: graphs generated by Cardiff University (see Castellino et al. [27] and Smith et al. [92]). Again, x is the number of transmitters in the network;
- P06-z: subproblems of the well-known *Philadelphia* problem, originally proposed in Anderson [10] (see also Smith et al. [92]). The generic graph P06-z is obtained by considering for every cell i of the problem a demand of $\left\lceil \frac{m(i)}{z} \right\rceil$, where $m(i)$ is the original demand for cell i.
- P06b-z: graphs obtained from the *Philadelphia* problem with the same method described for *P06-z*, but with a co-cell separation of 3 instead of the original 5. This has been done to more closely match the characteristics of realistic modern frequency assignment problems.

3.4.1.2 Graph set 2

The second family of scenarios is formed by only one type of graph:

- GSM2-x: adaptation to our model of realistic *GSM* scenarios. x is the number of transmitters in the network.

3.4.1.3 Graph set 3

The third family is composed of random scenarios we have generated using a basic graph generator. Our generator randomly places some sites into a rectangular region and assigns to each site a random number of transmitters. Given a required edge density for the graph, we fix the Euclidean distance ϵ below which there will be a constraint (edge). The highest separation δ and highest penalty ϕ which will appear in the problem are specified by the user. These values will be used for co-sited constraints. For each non co-site constraint, if we call σ the Euclidean distance between the two transmitters v and w involved, the required separation d_{vw} is a random integer in the following interval:

$$0 \leq d_{vw} \leq \left\lceil \frac{(\epsilon - \sigma)\delta}{\epsilon} \right\rceil \tag{3.5}$$

Each p_{vw} is generated in a similar way, and it is a random integer in the following interval:

$$1 \leq p_{vw} \leq \left\lceil \frac{(\epsilon - \sigma + 1)\phi}{\epsilon + 1} \right\rceil \tag{3.6}$$

We do not expect the scenarios created using our generator to be very realistic because of the very basic model of reality adopted, but we think they are adequate for our purpose. For the description of better, and more complex, frequency assignment problem generators, we refer the interested reader to Dunkin and Allen [36] and van Benthem [97].

Specifically, the third set of scenarios is composed of the following type of graph:

- $r1$-$r2$-s-x-w-δ-ϕ: *r1* is a random seed used to place sites on the rectangular area; *r2* is a second random seed adopted to calculate separations and penalties; s is the number of sites of the network; x is the number of transmitters to distribute among the s sites; w ($0 < w \leq 1$) is an approximation of the edge density of the graph (number of edges of the graph $\approx \left\lfloor \frac{wx(x-1)}{2} \right\rfloor$); δ is the maximum separation value in the scenario; ϕ is the maximum penalty value in the scenario.

3.4.1.4 Graph characteristics

In Table 3.1, Table 3.2 and Table 3.3 we summarise the characteristics of the graphs previously introduced. The meaning of the columns is as follows:

- Graph: names we will use to refer to the scenarios;
- $|V|$: number of vertices of each graph;
- $|E|$: number of edges of each graph;
- d_{vw}: separation values in each graph. The subcolumns have the following meaning:
  - Max: maximum separation in each graph;
  - Avg: average of the separations of each graph;
- p_{vw}: penalty values in each graph. The subcolumns have the same meaning as in column "d_{vw}".

3.4.2 Results of our heuristic algorithms

In this section we group the results obtained by the algorithms described in Section 3.2.2 and Section 3.3.2 on the benchmarks derived from the graphs previously described.

All the tests of this thesis have been carried out on a computer with an Intel Pentium II 400MHz processor, equipped with 128MB of memory.

Graph	$\lvert V\rvert$	$\lvert E\rvert$	d_{vw}		p_{vw}	
			Max	Avg	Max	Avg
AC-45-17	45	482	1	0.29	1	1.00
AC-45-25	45	801	1	0.34	1	1.00
AC-95-9	95	781	0	0.00	1	1.00
AC-95-17	95	2298	1	0.15	1	1.00
GSM-93	93	1073	1	0.28	1	1.00
GSM-246	246	7611	2	0.32	1	1.00
Test95	95	1214	4	1.37	1	1.00
Test282	282	10430	4	1.38	1	1.00
P06-5	88	3021	4	0.58	1	1.00
P06-3	153	9193	4	0.59	1	1.00
P06b-5	88	3021	2	0.39	1	1.00
P06b-3	153	9193	2	0.40	1	1.00

Table 3.1: Problem characteristics. Graph set 1.

Graph	$\lvert V\rvert$	$\lvert E\rvert$	d_{vw}		p_{vw}	
			Max	Avg	Max	Avg
GSM2-184	184	6809	2	0.20	10^8	$8.946 * 10^6$
GSM2-227	227	10088	2	0.18	10^8	$9.102 * 10^6$
GSM2-272	272	14525	2	0.16	10^8	$7.953 * 10^6$

Table 3.2: Problem characteristics. Graph set 2.

Graph	$\lvert V\rvert$	$\lvert E\rvert$	d_{vw}		p_{vw}	
			Max	Avg	Max	Avg
1-1-50-75-30-2-50	75	835	2	0.26	50	10.81
1-2-50-75-30-4-50	75	835	4	0.62	50	11.01
1-3-50-75-30-0-50	75	835	0	0.00	50	10.97
1-4-50-75-30-2-1	75	835	2	0.25	1	1.00
1-5-50-75-30-2-100	75	835	2	0.26	100	21.35
1-6-50-75-30-0-10000	75	835	0	0.00	10000	2068.48

Table 3.3: Problem characteristics. Graph set 3.

3.4.2.1 Parameter settings

After a parameter tuning phase, in which we have noticed that the tuning of the parameters is not very crucial for the results, we decided on the following settings. For all of the problems we have fixed $\alpha = 0.95$, $I_{sa} = 2 * 10^6$, $I_{max} = 5 * 10^7$, $T_{\min} = 10$, $\beta = 0.96$ and $I_{ts} = 5 * 10^4$. $t_{init} = 0.01$ for the problems based on *AC-45-17*, *AC-45-25*, *AC-95-9*, *AC-95-17*, *GSM-93*, *Test95* and on the graphs of the third set except *1-6-50-75-30-0-10000*; $t_{init} = 0.1$ for the problems based on *GSM-246*, *Test282*, *P06-5*, *P06-3*, *P06b-5* and *P06b-3*; $t_{init} = 1$ for the problems based on the graphs of the second set and on *1-6-50-75-30-0-10000.* $T_{\text{init}} = 500$ for the problems based on *AC-45-17*, *AC-45-25*, *AC-95-9* and on the graphs of the third set; $T_{\text{init}} = 1000$ for the problems based on *AC-95-17*, *GSM-93*, *GSM-246*, *Test95* and on the graphs of the second set. Finally $T_{\text{init}} = 2000$ for the remaining problems (based on *Test282*, *P06-5*, *P06-3*, *P06b-5* and *P06b-3*).

As the algorithms are naturally fast because of the efficient implementation, we had the opportunity to choose quite conservative values for the parameters, which give priority to a careful search instead of the convergence speed. Execution times are anyway under 45 minutes for all of the problems, and under 10 minutes for most of them.

3.4.2.2 Results

In Table 3.4, Table 3.5 and Table 3.6 we summarise the results obtained by our two algorithms on the benchmarks. Both the algorithms have been run five times on each problem. The columns of the tables have the following meaning:

- Problem: names of the problems. Each name is composed of the following two elements:

  - Graph: name of the graph on which the problem is based;
  - $|F|$: number of channels available;

- Simulated annealing: summary of the results obtained by the simulated annealing algorithm in the five runs. The subcolumns have the following meaning:
  - Min: the best result obtained on each problem;
  - Max: the worst result obtained on each problem;
  - Avg: the average of the results obtained on each problem in the five runs;
- Tabu search: summary of the results obtained by the tabu search algorithm in the five runs. The subcolumns have the same meaning as in column "Simulated annealing".

From Table 3.4, Table 3.5 and Table 3.6 the tabu search algorithm appears to be better than the simulated annealing algorithm, both in terms of best solution found and in terms of the average of the costs. It must however be observed that for some of the weighted problems (Table 3.5 and Table 3.6) the simulated annealing algorithms obtains better results than the tabu search algorithm.

It is also interesting to observe that both the approaches have obtained, in particular for the weighted problems of Table 3.5 and Table 3.6, quite scattered results in the five runs (i.e. there is a substantial difference between the worst and the best result of the five runs). This is an indicator of the difficulty of the algorithms to escape from local minima and to converge to a global minimum. For this reason we suspect that the upper bounds may not be very good for some of the problems. The scattered results suggest also

Problem		Simulated annealing			Tabu search		
Graph	$\|F\|$	Min	Max	Avg	Min	Max	Avg
AC-45-17	7	**32**	32	**32.0**	**32**	32	**32.0**
AC-45-17	9	**15**	15	**15.0**	**15**	16	15.4
AC-45-25	11	**33**	33	**33.0**	**33**	33	**33.0**
AC-45-25	19	**8**	8	**8.0**	**8**	8	**8.0**
AC-95-9	6	**31**	31	**31.0**	**31**	31	**31.0**
AC-95-9	10	**3**	3	**3.0**	**3**	3	**3.0**
AC-95-17	15	34	35	34.2	**33**	34	**33.8**
AC-95-17	21	**10**	10	**10.0**	**10**	10	**10.0**
GSM-93	9	**32**	33	**32.4**	**32**	34	33.0
GSM-93	13	**7**	8	7.8	**7**	8	**7.6**
GSM-246	21	82	85	83.0	**79**	84	**82.2**
GSM-246	31	28	30	28.8	**25**	28	**26.6**
Test95	31	**12**	13	**12.8**	**12**	13	**12.8**
Test95	36	**8**	9	8.4	**8**	9	**8.2**
Test282	61	61	65	63.6	**51**	59	**55.8**
Test282	71	37	39	38.0	**27**	34	**31.2**
P06-5	11	143	147	144.8	**133**	136	**134.6**
P06-5	41	**15**	15	**15.0**	**15**	15	**15.0**
P06-3	31	**115**	119	118.2	**115**	119	**116.4**
P06-3	71	**26**	26	**26.0**	**26**	26	**26.0**
P06b-5	21	**52**	52	**52.0**	**52**	52	**52.0**
P06b-5	31	**25**	25	**25.0**	**25**	25	**25.0**
P06b-3	31	**112**	112	**112.0**	**112**	113	112.4
P06b-3	71	**26**	26	**26.0**	**26**	26	**26.0**

Table 3.4: Upper bounds results. Benchmark set 1.

Problem		Simulated annealing			Tabu search		
Graph	$\|F\|$	Min	Max	Avg	Min	Max	Avg
GSM2-184	39	5849	6546	6207.6	**5521**	5869	**5736.4**
GSM2-184	49	**874**	905	**898.8**	999	1247	1118.0
GSM2-227	39	11125	12768	12151.8	**10979**	11984	**11431.4**
GSM2-227	49	2513	2717	**2597.4**	**2459**	3148	2743.4
GSM2-272	39	32210	34252	33654.0	**27416**	30149	**29208.2**
GSM2-272	49	8830	9820	9340.6	**7785**	8629	**8178.8**

Table 3.5: Upper bounds results. Benchmark set 2.

Problem		Simulated annealing			Tabu search		
Graph	$\|F\|$	Min	Max	Avg	Min	Max	Avg
1-1-50-75-30-2-50	5	1247	1259	**1253.4**	**1242**	1304	1266.0
1-1-50-75-30-2-50	10	119	128	121.2	**101**	118	**107.0**
1-1-50-75-30-2-50	11	**59**	81	76.0	68	74	**70.8**
1-1-50-75-30-2-50	15	**11**	13	**12.0**	12	13	12.2
1-2-50-75-30-4-50	11	347	366	355.8	**323**	344	**335.6**
1-3-50-75-30-0-50	11	**36**	40	38.2	**36**	38	**37.0**
1-4-50-75-30-2-1	10	**19**	19	**19.0**	**19**	20	19.2
1-5-50-75-30-2-100	10	204	229	218.8	**186**	209	**199.2**
1-6-50-75-30-0-10000	10	7140	8181	7736.8	**6942**	7464	**7180.0**

Table 3.6: Upper bounds results. Benchmark set 3.

that, given a problem, many short runs could produce a better upper bound than a single long run.

3.4.3 Effectiveness of the dynamic length tabu list in the tabu search algorithm

In this section we compare the results obtained by the tabu search algorithm which incorporates a dynamic length tabu list, with the results obtained by a tabu search algorithm with a fixed length tabu list (i.e. $\beta = 1$), which we will refer to as the *conventional tabu search algorithm.*

The implementation of the conventional algorithm is the same as described in Section 3.3.2.7 except for the exit criterion, which in this case is a maximum computation time of 45 minutes. For each problem considered, this time is longer than the time required by the tabu search algorithm with the dynamic length tabu list, which in the remainder of the chapter we will refer to as the *dynamic tabu search algorithm.*

In Table 3.7, Table 3.8 and Table 3.9 the results achieved by the conventional tabu search are compared with those obtained by the dynamic tabu search. Three different values for the length of the conventional tabu list have been considered for each problem, and for each one of these values the best result achieved in five runs is presented in the table. The columns of the table have the following meaning:

- Problem: names of the problems. Each name is composed of the following two elements:
  - Graph: name of the graph on which the problem is based;
  - $|F|$: number of channels available;
- DTS: best results obtained by the dynamic tabu search algorithm;
- CTS: best results obtained by the conventional tabu search algorithm. Subcolumns have the following meaning:
  - T_i: length of the tabu list;
  - Val$_i$: best upper bound obtained with a tabu list of length T_i.

The advantage arising from the use of the dynamic length tabu list is clear from Table 3.7, Table 3.8 and Table 3.9. The dynamic tabu search algorithm obtains a worse result than the conventional tabu search algorithm in only two cases (problem *GSM2-184* with $|F| = 49$ and problem *1-1-50-75-30-2-50* with $|F| = 15$). It is interesting to notice that these are problems for which the simulated annealing algorithm obtained a better result than the tabu search algorithm in the tests of Section 3.4.2.2. This may be an indication that, for these problems, the five runs of the tabu search algorithm considered in Section 3.4.2.2 were particularly unlucky.

It is also important to observe that the parameter tuning of the dynamic tabu search algorithm is easier because even a greatly overestimated choice

Problem		DTS	CTS					
Graph	$\lvert F\rvert$		T_1	Val_1	T_2	Val_2	T_3	Val_3
AC-45-17	7	**32**	450	36	100	**32**	30	**32**
AC-45-17	9	**15**	450	20	100	**15**	30	**15**
AC-45-25	11	**33**	450	**33**	140	**33**	30	34
AC-45-25	19	**8**	450	**8**	140	**8**	30	**8**
AC-95-9	6	**31**	450	36	140	**31**	30	**31**
AC-95-9	10	**3**	450	**3**	140	**3**	30	**3**
AC-95-17	15	**33**	450	40	100	34	30	34
AC-95-17	21	**10**	450	13	100	**10**	30	**10**
GSM-93	9	**32**	900	44	100	33	50	33
GSM-93	13	**7**	900	13	100	**7**	50	**7**
GSM-246	21	**79**	900	94	500	87	100	100
GSM-246	31	**25**	900	36	500	33	100	34
Test95	31	**12**	900	15	300	**12**	100	**12**
Test95	36	**8**	900	10	300	**8**	100	**8**
Test282	61	**51**	1800	83	400	63	100	68
Test282	71	**27**	1800	52	400	36	100	40
P06-5	11	**133**	1800	161	400	**133**	100	161
P06-5	41	**15**	1800	**15**	400	**15**	100	21
P06-3	31	**115**	1800	121	400	123	100	144
P06-3	71	**26**	1800	**26**	400	29	100	31
P06b-5	21	**52**	1800	**52**	400	**52**	100	61
P06b-5	31	**25**	1800	**25**	400	**25**	100	27
P06b-3	31	**112**	1800	117	300	117	100	117
P06b-3	71	**26**	1800	**26**	300	**26**	100	**26**

Table 3.7: Effectiveness of the dynamic length tabu list in the tabu search algorithm. Benchmark set 1.

Problem		DTS	CTS					
Graph	$\lvert F\rvert$		T_1	Val_1	T_2	Val_2	T_3	Val_3
GSM2-184	39	**5521**	900	6806	300	5881	100	5896
GSM2-184	49	999	900	1054	300	**874**	100	999
GSM2-227	39	**10979**	900	13325	500	11561	100	12996
GSM2-227	49	**2459**	900	3322	300	2517	100	2961
GSM2-272	39	**27416**	900	30775	500	29481	100	39506
GSM2-272	49	**7785**	900	8877	500	8776	100	10165

Table 3.8: Effectiveness of the dynamic length tabu list in the tabu search algorithm. Benchmark set 2.

Problem Graph	$\|F\|$	DTS	CTS T_1	Val_1	T_2	Val_2	T_3	Val_3
1-1-50-75-30-2-50	5	**1242**	450	1347	140	1255	30	1352
1-1-50-75-30-2-50	10	**101**	450	163	140	105	30	165
1-1-50-75-30-2-50	11	**68**	450	101	140	73	30	102
1-1-50-75-30-2-50	15	12	450	18	140	**11**	30	15
1-2-50-75-30-4-50	11	**323**	450	444	140	327	30	441
1-3-50-75-30-0-50	11	**36**	450	48	140	39	30	44
1-4-50-75-30-2-1	10	**19**	450	24	140	20	30	20
1-5-50-75-30-2-100	10	**186**	450	336	140	198	30	349
1-6-50-75-30-0-10000	10	**6942**	450	8982	140	7176	30	9100

Table 3.9: Effectiveness of the dynamic length tabu list in the tabu search algorithm. Benchmark set 3.

for parameters T_{init}, β and I_{ts} does not compromise the quality of the estimates (although the convergence speed of the algorithm may be affected). On the contrary, the choice of parameter T is crucial for the conventional tabu search algorithm.

3.4.4 Comparison with algorithms of other authors

In this section we compare our simulated annealing and tabu search algorithms with two programs developed by other authors.

The first program considered is FASoft (Hurley et al. [52]). It treats only problems where the target is to minimise the number of constraint violations, so we cannot run it on the second and, except for one problem, the third families of benchmarks. As stated in Section 3.1, FASoft contains more than one algorithm. For the tests reported in this thesis we have adopted the tabu search algorithm, which seems to be the best one of the collection.

The second algorithm we consider is the tabu search algorithm NBS, developed by Whitaker et al. [103]. The algorithm has been originally developed to deal with binary and non binary constraints, but we do not have this last type of constraint. Thus we do not use all of its functionality. Unfortunately the method is not able to manage the problems of the second benchmarks family because their penalties are too high. Consequently we have run NBS only on the problems of the first and the third families.

The methods have been tuned in such a way to have computation times similar to those of the algorithms developed by us. Notwithstanding our efforts in the tuning phase, we believe that the developers of the algorithms could have found better parameter configurations, which probably would have produced better results.

In Table 3.10 and Table 3.11 we compare the upper bounds produced by our algorithms with those provided by FASOFT and NBS. The results presented are, for each method, the best obtained in five or more runs. The columns of the tables have the following meaning:

- Problem: names of the problems. Each name is composed of the following two elements:
  - Graph: name of the graph on which the problem is based;
  - $|F|$: number of channels available;
- Simulated annealing: best results obtained by the simulated annealing algorithm described in Section 3.2.2;
- Tabu search: best results obtained by the tabu search algorithm described in Section 3.3.2;
- FASOFT: best results obtained by the tabu search algorithm contained in the system FASOFT. The symbol " - ", which appears in the rows

Problem		Simulated	Tabu	FASoft	NBS
Graph	$\|F\|$	annealing	search		
AC-45-17	7	**32**	**32**	33	**32**
AC-45-17	9	**15**	**15**	16	16
AC-45-25	11	**33**	**33**	34	**33**
AC-45-25	19	**8**	**8**	9	**8**
AC-95-9	6	**31**	**31**	**31**	**31**
AC-95-9	10	**3**	**3**	**3**	**3**
AC-95-17	15	34	**33**	36	36
AC-95-17	21	**10**	**10**	11	**10**
GSM-93	9	**32**	**32**	34	39
GSM-93	13	**7**	**7**	9	9
GSM-246	21	82	**79**	89	93
GSM-246	31	28	**25**	35	36
Test95	31	**12**	**12**	15	13
Test95	36	**8**	**8**	10	9
Test282	61	61	**51**	75	70
Test282	71	37	**27**	45	42
P06-5	11	143	**133**	137	144
P06-5	41	**15**	**15**	16	**15**
P06-3	31	**115**	**115**	123	132
P06-3	71	**26**	**26**	28	29
P06b-5	21	**52**	**52**	**52**	53
P06b-5	31	**25**	**25**	**25**	**25**
P06b-3	31	**112**	**112**	113	116
P06b-3	71	**26**	**26**	**26**	**26**

Table 3.10: Upper bounds comparison. Benchmark set 1.

of the weighted problems, means that no result is available (method cannot manage this type of problem);

- NBS: best results obtained by the tabu search algorithm NBS.

On the benchmarks analysed in Table 3.10 and in Table 3.11, both FASoft and NBS are clearly dominated by our algorithms. On some problems (in particular on those of the third benchmark set) there is a great difference among the upper bounds provided by the different algorithms. This may be

Problem Graph	$\|F\|$	Simulated annealing	Tabu search	FASoft	NBS
1-1-50-75-30-2-50	5	1247	**1242**	-	1296
1-1-50-75-30-2-50	10	119	**101**	-	131
1-1-50-75-30-2-50	11	**59**	68	-	112
1-1-50-75-30-2-50	15	**11**	12	-	17
1-2-50-75-30-4-50	11	347	**323**	-	371
1-3-50-75-30-0-50	11	**36**	**36**	-	44
1-4-50-75-30-2-1	10	**19**	**19**	**19**	21
1-5-50-75-30-2-100	10	204	**186**	-	251
1-6-50-75-30-0-10000	10	7140	**6942**	-	18034

Table 3.11: Upper bounds comparison. Benchmark set 3.

seen as an indication of the difficulty of these problems.

We believe that the better performance of our methods can be explained partly by a better implementation. The two tabu search algorithms of FASoft and NBS are implemented in a classic way, and do not use the special structures adopted in the implementation of our algorithms (see Section 3.2.2.7 and Section 3.3.2.7). A consequence of the adoption of these special structures is that our tabu search is able to use a full neighbourhood, which we believe to be a really important factor in the quality of our results. FASoft and NBS have to use a random (partial) neighbourhood for computational reasons.

Considering the pure speed of the algorithms, we can observe that, notwithstanding that we consider a full neighbourhood instead of a partial one as the other methods, our tabu search is, in terms of number of iterations carried out in a given interval of time, at least 5 times faster than the tabu search of FASoft and at least 20 times faster than the one developed within NBS. It is anyway important to remind the reader that NBS is the only method, among those compared, which is able to deal with non binary constraints. This extra feature contributes to make the algorithm slower.

To understand the role of the speed of our methods in the better results obtained by them, we tested FASOFT and NBS on longer runs. Practically no improvement was found, and this suggests that the superiority of our methods does not depend only on their speed, but probably on the different neighbourhood adopted. However it must be observed that the use of a better neighbourhood in our methods is a direct consequence of the use of an efficient implementation.

Another factor which could make our tabu search algorithm more effective than those implemented in FASOFT and NBS is, we believe, the dynamic length tabu list (as observed in Section 3.4.3).

Chapter 4

Lower Bounds

In this chapter, after a brief review of the literature, we present some lower bounding techniques for the *FS-FAP*.

Some methods calculate lower bounds for the cost paid (number of constraint violations present) in complete subgraphs (*clique-like subproblems*) of a given problem. These estimates are used in other methods to produce lower bounds for the original problem.

The last section is dedicated to computational results and a brief analysis of them.

A preliminary version of some of the methods described in this chapter has been presented in Montemanni et al. [81].

4.1 Literature review

Few lower bounding methods for the *FS-FAP* appear in the literature. Unfortunately those that do appear are based on very basic ideas or developed for problems with particular features, which are used to make the methods work. To the knowledge of the author no general method exists.

In the rest of this section we present an overview of the approaches proposed so far.

4.1.1 Lower bound by Allen et al.

In Allen et al. [7] a lower bound based on the adaptation of the *Travelling Salesman Problem* bound, originally developed for the *MS-FAP* (see, for example, Smith et al. [86]) to the *FS-FAP* is described. The method for the *MS-FAP*, which is to create and solve a linear program, is designed to work on near-clique subproblems of the entire problem. A detailed description of the adaptation of the method to the *FS-FAP* is given in Section 4.2.1.

The main limitation of the bound is that it is non-trivial only for near-clique problems.

4.1.2 Lower bound by Smith et al.

A simple lower bound is described in Smith et al. [86]. The method is based on a closed formula, applied to clique-like subproblems. This formula returns, for a given number of available frequencies and a given clique, a lower bound for the number of vertices that cannot be satisfied in a violation free assignment of the clique. This is also a lower bound for the number of constraint violations in the subproblem. We will give a more detailed description of the formula in Section 4.2.2. A lower bound for the number of constraint violations of the global problem is given by the sum of the bounds obtained on some disjoint clique-like subgraphs of it.

The idea on which this method is based is very simple, but in general the results are poor, especially because the number of violating transmitters (which the formula returns) can be much less than the number of constraint violations.

4.1.3 Lower bounds by Aardal et al.

In Aardal et al. [2] a lower bounding technique developed within the EUCLID CALMA project is presented. It is based on some valid inequalities for the linear relaxation of formulation $FAP2$ (see Section 2.2.2). Non-zero bounds have only rarely been found with this approach.

4.1.4 Lower bounds by Hurkens and Tiourine

Hurkens and Tiourine [51] present a second lower bounding technique developed within the EUCLID CALMA project (see also Tiourine et al. [96] and Aardal et al. [3]). It works only for problems where most of the transmitters have a preassigned frequency (we do not consider this feature in our model) and the penalties to be paid when changing these frequencies dominate the penalties generated by frequency separation constraints. They formulate a new problem, the solution of which provides a lower bound for the original one. The relaxed problem is a non-linear program where for each transmitter it is decided whether or not to change the preassigned frequency.

The results are good, but problems dominated by preassigned frequencies are very uncommon in practice.

4.1.5 Lower bounds by Koster et al.

Two other methods developed on the EUCLID CALMA project benchmarks (after the end of the project) have been presented by Koster et al.. In [60] and [61] the polytope of $FAP1$ (see Section 2.2.1) is studied from a polyhedral point of view and some facet defining inequalities are presented. Computational tests show that the method is practical only for problems with very small frequency domains (less than 7). As the authors state, it is impossible to approach real life problems using this method.

A second lower bounding technique is presented in [60], [62] and [63] (see also Aardal et al. [3]). Here Koster et al. propose first of all some preprocessing rules that reduce the size of the problems by trivial considerations. Sometimes good lower bounds are provided just by applying those rules. The gap between the best known upper bound and the lower bound so obtained is closed by at least 73% on 8 of the 11 problems given (2 of them are solved to optimality). A dynamic programming algorithm is also presented. It works by computing a tree decomposition of the problems resulting after preprocessing, and by partitioning the frequency domains into subsets. Each partition is considered as a frequency in a new problem. This approach is able to obtain the optimum for 5 of the problems and quite good lower bounds for the remaining 4 (2 were solved to the optimum by preprocessing).

The preprocessing rules appear to work only on the benchmarks the authors studied, and the dynamic programming algorithm is strongly based on two peculiarities of the EUCLID CALMA project's problems: the tree-like structure and the natural partition characterising frequency domains (they are separated intervals of frequencies). Realistic problems generally do not have tree-like structures and their frequency domains tend to be (almost) continuous intervals of spectrum.

4.1.6 Lower bounds by Maniezzo and Montemanni

In Maniezzo and Montemanni [72] and Montemanni [75] a family of constructive lower bounds is described. The work is based on the observation that the *FS-FAP* can be seen as a simplification of the *Quadratic Assignment Problem* (see, for example, Maniezzo [69] for a description of this problem).

The lower bounds presented are consequently adaptations of classical lower bounds originally developed for this problem.

The results obtained are acceptable only for the EUCLID CALMA project benchmarks. When more general benchmarks are considered, the method works only for small problems with limited frequency domains, and also in this case it requires long computation times.

4.1.7 Lower bound by Chardaire and Sutter

Chardaire and Sutter [28] describe a lower bounding method for the *Unconstrained 0-1 Quadratic Programming Problem*, through which it is possible to represent the *FS-FAP*.

The method obtains good lower bounds, but unfortunately the computational time grows exponentially with the size of the problem. The largest problem which it is possible to bound with such an approach has 100 variables, which in terms of frequency assignment is a very small problem. The number of variables of the 0-1 quadratic programming representation of *FS-FAP* is given by $|V|\,|F|$.

4.1.8 Lower bound by Helmberg

More recently Helmberg [50] presents a new lower bound based on semidefinite programming (see also Eisenblätter et al. [39]). The method provides a lower bound for the *Max k-Cut Problem*, that is seen as a simplification of the *FS-FAP*.

The main limitations of this approach are:

- it can model only *co-channel* constraints (separations of 1 channel);

- the computation complexity is intrinsically high;

- numerical problems may occur.

In particular, modelling only co-channel constraints could be a great limitation because most of the real networks have *co-site* constraints, that model the separations that must exist among transmitters installed at the same site. These co-site separations usually require 2 or 3 channels separation, and they must be relaxed to 1 channel to apply the method described in [50].

The results presented, which are for problems with co-channel constraints only, are satisfying, but with long computation times, especially for the larger problems.

4.2 Local lower bounds

In this section we describe two methods working on clique-like problems. The first produces a lower bound for the cost paid in the problem, while the second gives a lower bound for the number of constraint violations.

The idea is to apply these methods on clique-like subproblems of a bigger problem. Because they do not work on the whole problem but just on a part of it, we will call these methods *local* lower bounds.

It is interesting to point out that the idea of working on (near) clique subproblems is well-known to be a successful approach for lower bounding minimum span frequency assignment problems (see, for example, Smith and Hurley [90], Smith et al. [86] and Allen et al. [8]).

Before the description of the methods, we need to specify exactly what we will refer to as a clique-like subproblem in the remainder of the thesis. They are the following two types of subgraphs.

Definition 1. *A* level-k-h clique *of G is a complete subgraph in which every edge has label at least k in D and label at least h in P, which is not contained* *in any larger such complete subgraph.*

Definition 2. *A* level-k-h non-maximal clique *of G is a complete subgraph in which every edge has label at least k in D and label at least h in P, which is contained in a larger such complete subgraph.*

In the remainder of the thesis we will refer to *level*-0-1 *non-maximal cliques* simply as *non-maximal cliques.*

4.2.1 *TSP* lower bound

The methods described in this section return a lower bound for the cost (weighted sum of penalties) paid in a level-k-h (non-maximal) clique $C = \{V_C, E_C\}$.

We present the original method, which is described in Allen et al. [7], and a simplified version we have developed.

4.2.1.1 Original method

The original idea (described in Allen et al. [7] and briefly introduced in Section 4.1.1) is to adapt the *Travelling Salesman Problem* bound, originally developed for the *MS-FAP*, to the *FS-FAP*. Ideally the target becomes to find a Hamiltonian path with length equal to the available span, where the length is given by the sum of the separations between each pair of consecutive vertices in the path. The penalty is paid when the separation between the vertices of an active edge is less than the one required not to have interference. In the conversion process some simplifications have been introduced. The most important of these simplifications is the elimination of the so-called *subtour elimination* inequalities, which were originally inserted to force the active variables to form an unique circuit.

In [7] the authors propose an integer program, the target of which is to find the lowest cost set of disjoint circuits of total length $|F|-1$ in the clique C', obtained from C by adding a dummy vertex D connected to all the other vertices $v \in V_C$ by an edge with zero length. D has been added to work on a closed *TSP*-like problem instead of an open one. To describe in detail this formulation, which we call TSP_{IP}^{AL}, we first need to introduce the meaning of the variables.

- u_{Dv}: {0,1} variable defined for every $v \in V$. It is 1 when the vertex of C associated with transmitter v is connected to the dummy vertex D in the set of active circuits; 0 otherwise;

- u_{vwl}: {0,1} variable defined for every $\{v, w\} \in E$ and for every $l \in \{0, 1, \ldots, |F|-1\}$. It is 1 when transmitter v is adjacent to transmitter w $(v < w)$ in one of the active circuits and their frequency separation is l; 0 otherwise.

$$(TSP^{AL}_{IP}) \quad \text{Min} \sum_{\{v,w\}\in E_C} \sum_{i=0}^{d_{vw}} p_{vw} u_{vwi} \tag{4.1}$$

$$\text{s.t.} \sum_{i=0}^{|F|-1} u_{vwi} \leq 1 \qquad \forall v \in V_C \tag{4.2}$$

$$u_{Dv} + \sum_{\substack{w\in V_C;\\ v<w}} \sum_{i=0}^{|F|-1} u_{vwi} + \sum_{\substack{w\in V_C;\\ v>w}} \sum_{i=0}^{|F|-1} u_{wvi} = 2 \quad \forall v \in V_C \tag{4.3}$$

$$\sum_{v\in V_C} u_{Dv} = 2 \tag{4.4}$$

$$\sum_{\{v,w\}\in E_C} \sum_{i=1}^{|F|-1} i\, u_{vwi} = |F| - 1 \tag{4.5}$$

$$u_{Dv} \in \{0,1\} \qquad \forall v \in V_C \tag{4.6}$$

$$u_{vwi} \in \{0,1\} \qquad \begin{array}{l} \forall \{v,w\} \in E_C; \\ i = 0, \ldots, |F|-1; \end{array} \tag{4.7}$$

The target, minimising the penalty on disjoint circuits, is expressed by (4.1). Inequalities (4.2) express that at most one of the u_{vwi}'s can be active between every pair of transmitters $\{v, w\}$. Equations (4.3) and (4.4) force every element of the set $V_C \cup \{D\}$ to be connected with exactly two other elements of the same set. Equation (4.5) expresses the fact that the sum of the length of the circuits has to be equal to the fixed span, while equations (4.6) and (4.7) specify the domains of the variables.

For computational reasons it would be difficult to deal with the integer program TSP^{AL}_{IP} directly, so integrality constraints have been relaxed into continuous constraints, obtaining what we will refer to as TSP^{AL}_{LR}. Formally in TSP^{AL}_{LR} the following constraints substitute (4.6) and (4.7) respectively:

$$0 \leq u_{Dv} \leq 1 \qquad \forall v \in V_C \tag{4.8}$$

$$0 \leq u_{vwi} \leq 1 \qquad \forall \{v,w\} \in E_C;\ i = 0, \ldots, |F|-1 \tag{4.9}$$

4.2.1.2 Improved method

Soon we realised that the formulation described in the previous section can be improved by considering a new linear program, which we will refer to as TSP_{LR}. TSP_{LR} is a simplification of TSP_{LR}^{AL} because it provides the same bounds but has many fewer variables. The variables of TSP_{LR} have the following meanings:

- u_{Dv}: continuous variable relaxed from a {0,1} variable. It is defined for every $v \in V$. A value of 1 means that the vertex associated with transmitter v is connected to the dummy vertex D in the set of active circuits;

- u_{vw}^{V}: continuous variable relaxed from a {0,1} variable. It is defined for every $\{v, w\} \in E$. A value of 1 means that transmitter v is adjacent to transmitter w $(v < w)$ in one of the active circuits and the constraint on the edge $\{v, w\}$ is violated;

- u_{vw}^{N}: continuous variable relaxed from a {0,1} variable. It is defined for every $\{v, w\} \in E$. A value of 1 means that transmitter v is adjacent to transmitter w $(v < w)$ in one of the active circuits and the constraint on the edge $\{v, w\}$ is not violated.

$$(TSP_{LR}) \quad \text{Min} \sum_{\{v,w\}\in E_C} p_{vw} u^V_{vw} \tag{4.10}$$

$$\text{s.t. } u^V_{vw} + u^N_{vw} \leq 1 \qquad \forall \{v,w\} \in E_C \tag{4.11}$$

$$u_{Dv} + \sum_{\substack{w\in V_C;\\ v<w}} (u^V_{vw} + u^N_{vw}) + \sum_{\substack{w\in V_C;\\ v>w}} (u^V_{wv} + u^N_{wv}) = 2 \quad \forall v \in V_C \tag{4.12}$$

$$\sum_{v\in V_C} u_{Dv} = 2 \tag{4.13}$$

$$\sum_{\{v,w\}\in E_C} (d_{vw}+1)u^N_{vw} \leq |F| - 1 \tag{4.14}$$

$$0 \leq u_{Dv} \leq 1 \qquad \forall v \in V_C \tag{4.15}$$

$$0 \leq u^V_{vw} \leq 1 \qquad \forall \{v,w\} \in E_C \tag{4.16}$$

$$0 \leq u^N_{vw} \leq 1 \qquad \forall \{v,w\} \in E_C \tag{4.17}$$

TSP_{LR} can be seen as a modification of TSP^{AL}_{LR} where there are just two possible variables (instead of $|F|$) for each active edge $\{v,w\}$: u^V_{vw} (violated, representing separation 0) and u^N_{vw} (non-violated, representing separation $d_{vw}+1$). A consequence of this simplification is the difference between constraint (4.5) and constraint (4.14), where the sign that was previously "=", becomes "≤".

We will prove that, given a clique C, the cost of an optimal solution of TSP_{LR} is always equal to the cost of an optimal solution of TSP^{AL}_{LR}. First we need to give the following definition.

Definition 3. Opt*(G, LP) is the cost of an optimal solution of the linear program LP, when it is applied to graph G.*

Theorem 1. *Opt(C, TSP_{LR}) = Opt(C, TSP^{AL}_{LR}).*

Proof. We will prove the inequalities *Opt(C, TSP_{LR})* $\leq$ *Opt(C, TSP^{AL}_{LR})* and *Opt(C, TSP_{LR})* $\geq$ *Opt(C, TSP^{AL}_{LR})* separately. To make the exposition clearer we will refer to the u variables of TSP_{LR} as $u^{(LR)}$ and to the u variables of TSP_{LR}^{AL} as $u^{(AL)}$.

$Opt(C, TSP_{LR}) \leq Opt(C, TSP_{LR}^{AL})$:

We construct a feasible solution of TSP_{LR} from a given solution of TSP_{LR}^{AL}. $\forall v \in V_C\ u_{Dv}^{(LR)} = u_{Dv}^{(AL)}$; $\forall \{v, w\} \in E_C\ u_{vw}^{V(LR)} = \sum_{i=0}^{d_{vw}} u_{vwi}^{(AL)}$ and $u_{vw}^{N(LR)} = \sum_{i=d_{vw}+1}^{|F|-1} u_{vwi}^{(AL)}$. The solution of TSP_{LR} so obtained is trivially feasible and has the same cost as the original solution of TSP_{LR}^{AL}. The procedure, when applied to an optimal solution of TSP_{LR}^{AL}, proves the inequality.

$Opt(C, TSP_{LR}) \geq Opt(C, TSP_{LR}^{AL})$:

We construct a feasible solution of TSP_{LR}^{AL} from a given solution of TSP_{LR}. $\forall v \in V_C\ u_{Dv}^{(AL)} = u_{Dv}^{(LR)}$; $\forall \{v, w\} \in E_C\ u_{vw0}^{(AL)} = u_{vw}^{V(LR)}$, $u_{vw(d_{vw}+1)}^{(AL)} = u_{vw}^{N(LR)}$; $\forall i \in \{1, \ldots, d_{vw}, d_{vw}+2, \ldots, |F|-1\}\ \forall \{v, w\} \in E_C\ u_{vwi}^{(AL)} = 0$. The solution of TSP_{LR}^{AL} so obtained could violate constraint (4.5). It is possible to select a set of edges $\{v, w\}$ for which $u_{vw0}^{AL} + u_{vw(d_{vw}+1)}^{AL} > 0$, to reduce the value of u_{vw0}^{AL} and $u_{vw(d_{vw}+1)}^{AL}$ variables and to increase other u_{vwi}^{AL} variables, with $i > d_{vw}+1$, by the same quantity. With this operation it is possible to satisfy constraint (4.5) without increasing the cost of the solution, which is consequently less than or equal to the one of the initial solution of TSP_{LR}. The procedure, when applied to an optimal solution of TSP_{LR}, proves the inequality. □

The integrality of the objective of the solutions of TSP_{LR} is not guaranteed, but in the original problem TSP_{IP}^{AL} every solution must have an integer cost, so we can round the cost of a non-integer solution to the integer above. Consequently, in the remainder of this thesis, when we will refer to the result returned by the TSP bound, we will refer to

$$\lceil \text{ optimal solution of } TSP_{LR} \rceil \tag{4.18}$$

Formulation	No. of Constraints	No. of Variables
TSP_{LR}^{AL}	$\|E\| + \|V\| + 2$	$\|V\| + \|F\|\|E\|$
TSP_{LR}	$\|E\| + \|V\| + 2$	$\|V\| + 2\|E\|$

Table 4.1: Dimensions of formulations TSP_{LR}^{AL} and TSP_{LR}.

4.2.1.3 Dimensions of TSP_{LR}^{AL} and TSP_{LR}

In Table 4.1 we compare the dimensions of formulations TSP_{LR}^{AL} and TSP_{LR}. The table clearly shows how the number of variables of the formulation which produces the lower bound is reduced when TSP_{LR} is used instead of TSP_{LR}^{AL}. The columns of the table have the following meanings:

- Formulation: names of the two formulations we compare;
- No. of Constraints: expressions for the number of constraints of the formulations (number of rows of the problem matrix). Constraints defining variable domains are not counted here;
- No. of Variables: expressions for the number of variables of the formulations (number of columns of the problem matrix).

The greatest benefit of the use of TSP_{LR} instead of TSP_{LR}^{AL} should be faster solution times, because a smaller linear program (the number of variables is strongly reduced, especially for problems with $|F|$ large) is in general easier to solve. In our global lower bounding strategies (see Section 4.3), TSP_{LR} will be solved many times, and for this reason any improvement in its solution time translates into a major improvement in the computation time for the entire algorithm.

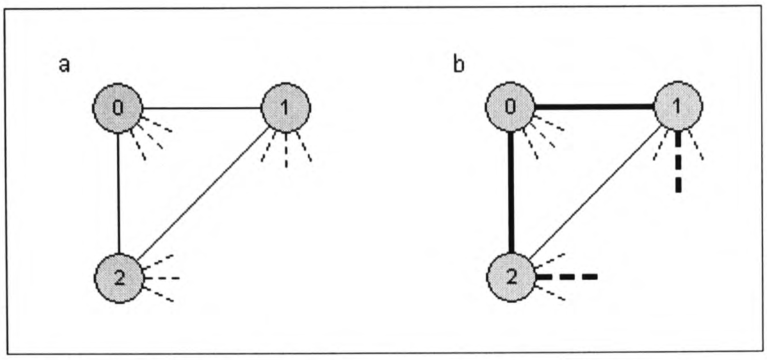

Figure 4.1: Limitation of the *TSP* lower bound.

4.2.1.4 Limitation of the method

The weakest point of the lower bound obtained solving TSP_{LR} (or even TSP_{IP}^{AL}) is that it ignores every (possible) penalty between transmitters that are not consecutive in the circuits of a solution. An example of such a situation is presented in Figure 4.1.

In Figure 4.1 part *a* we show three vertices $(0, 1, 2)$ of a larger clique C on which we calculate the *TSP* bound, and the induced edges. All the d's in the subproblem are zero ($d_{01} = d_{02} = d_{12} = 0$) and $|F|$ frequencies are available ($F = \{0, 1, \ldots, |F| - 1\}$). In part *b* we depict in bold the edges of our subproblem which are active in a solution of the linear program *TSP*$_{LR}$ calculated on the clique C. We imagine the edges $\{0, 1\}$ and $\{0, 2\}$ to be violated ($u_{01}^{V} = u_{02}^{V} = 1$). Switching to the original *FS-FAP* we can observe how the solution that arises from the bold subcircuit would have the vertices 0, 1 and 2 assigned to the same frequency, and consequently a violated constraint between vertices 1 and 2, which is actually ignored by the *TSP* bound (since $u_{12}^{V} = 0$).

We can conclude that we expect this approach not to give very good bounds on problems where optimal solutions have a high density of violated constraints, that would lead to many consecutive violated edges in the solution of TSP_{LR} (TSP_{IP}^{AL}). These problems are anyway quite uncommon because, if they are modelled in a proper way, too many constraints violated would generally imply high interference, and this would probably mean that the available spectrum is simply not adequate to establish the desired network.

4.2.2 *Formula* lower bound

The methods described in this section provide a lower bound for the number of constraint violations that will occur in the level-(k-1)-h (non-maximal) clique $C = \{V_C, E_C\}$.

4.2.2.1 Original method

A first way to obtain a lower bound for the number of constraint violations is through the formula described in Smith et al. [86], briefly introduced in Section 4.1.2:

$$|V_C| - 1 - \left\lfloor \frac{|F| - 1}{k} \right\rfloor \tag{4.19}$$

This formula counts the number of vertices which it is impossible to assign in a violation-free assignment using the available span. This is trivially a lower bound for the number of violations in C.

Unfortunately the bound given by (4.19) in usually poor.

4.2.2.2 Improved method

To generate better lower bounds than those returned by (4.19), we have introduced a new formula, which is based on the following ideas. A lower bound for a level-(k-1)-h clique C is obtained by partitioning the set of available frequencies into $\left\lceil \frac{|F|}{k} \right\rceil$ sets of consecutive frequencies. Interference between transmitters assigned to frequencies of different sets is ignored. The number of violations is minimised if the transmitters are spread between sets as evenly as possible.

In order to formalise this approach, we need the following definitions:

$$\alpha = \left\lfloor \frac{|V_C|}{\left\lceil \frac{|F|}{k} \right\rceil} \right\rfloor \tag{4.20}$$

$$\beta = |V_C| \bmod \left\lceil \frac{|F|}{k} \right\rceil \tag{4.21}$$

α represents the lowest cardinality of the sets into which V_C is partitioned. β is the number of sets with cardinality $\alpha + 1$.

Theorem 2. *The integer value*

$$\frac{\left(\beta\alpha(\alpha+1) + \left(\left\lceil \frac{|F|}{k} \right\rceil - \beta\right)\alpha(\alpha-1)\right)}{2} \tag{4.22}$$

is a lower bound for the number of constraints violated in a level-(k-1)-h clique $C = \{V_C, E_C\}$, where there are $|F|$ consecutive available frequencies.

Proof. We will prove that (4.22) expresses the minimal solution, in terms of constraint violations, for a clique $C' = \{V_{C'}, E_{C'}\}$ where $|V_{C'}| = |V_C|$ and $\forall\{v,w\} \in E_{C'}$ $d_{vw} = k-1$. As $\forall\{v,w\} \in E_C$ $d_{vw} \geq k-1$, the minimal solution for C' is a lower bound for the minimal solution of C.

An assignment S of frequencies to the vertices of $V_{C'}$ is considered and the frequency assigned to vertex v in assignment S is denoted by $f_S(v)$. $\forall p \in \left\{0, 1, \ldots, \left\lceil \frac{|F|}{k} \right\rceil k - 1\right\}$ we define $B_p^S = \{v \in V_{C'} \mid f_S(v) = p\}$.

A new assignment S' with no more constraint violations than S is constructed as follows. We define $F' = \{mk \mid m \in \mathcal{N}_0^+\} \cap F$. S' is obtained, starting from S, by setting $\forall p \in F'$ $B_p^{S'} = \cup_{j=p}^{p+k-1} B_j^S$ and $\forall p \notin F'$ $B_p^{S'} = \emptyset$. There are no more constraint violations in S' than in S because if $|f_{S'}(v) - f_{S'}(w)| \leq k - 1$ with $v, w \in V_{C'}$ then $|f_S(v) - f_S(w)| \leq k - 1$.

Next we show that for a minimal assignment S'' with the properties of S', we have $|B_p^{S''}| - |B_{p'}^{S''}| \leq 1$ for any pair p, p' of frequencies in F', i.e the sizes of the sets B_p^S are either all the same or take two distinct values differing by 1. Suppose that $|B_p^{S''}| = a$ and $|B_{p'}^{S''}| = b$ with $b = a - l$ and $l \geq 2$. If we move a vertex from $|B_p^{S''}|$ to $|B_{p'}^{S''}|$ the change in the number of constraint violations is:

$$\frac{(a-1)(a-2) + (b+1)b - a(a-1) - b(b-1)}{2}$$

$$= -a + b - 1 = -l + 1 < 0 \quad \text{for} \quad l \geq 2.$$

Thus we can repeat this operation for pairs of vertices, decreasing the number of constraint violations until we obtain an assignment S''' with $|B_p^{S'''}| - |B_{p'}^{S'''}| \leq 1$ for every pair p, p' of frequencies in F'. Then there will be β sets of size $\alpha + 1$ and $\left(\left\lceil \frac{|F|}{k} \right\rceil - \beta\right)$ sets of size α. The number of constraint violations of S''' is then $\frac{\left(\beta\alpha(\alpha+1) + \left(\left\lceil \frac{|F|}{k} \right\rceil - \beta\right)\alpha(\alpha-1)\right)}{2}$ □

4.2.2.3 Limitation of the method

The main limitation of the *formula* lower bound is that it reduces all the separations d_{vw} involved to $k - 1$, and this does not appear to be a very promising approach for problems with a wide range of separation values.

4.3 Global lower bounds

In this section we present some techniques which produce lower bounds for the cost paid in a general *FS-FAP*. Having defined the lower bounds working on subproblems as *local*, by contrast we will call these new lower bounds *global*, as they work on complete problems.

The global techniques described in this section are heavily based on the local bounds described in Section 4.2.

4.3.1 Lower bound *LB1*

The idea of this bound, which we call *LB1*, is the same as that described theoretically, but not implemented, in Smith et al. [86] and Allen et al. [7]. A set of disjoint clique-like subproblems of the graph G, representing a *FS-FAP*, is selected and the local lower bounds described in Section 4.2 are calculated on these subproblems. The sum of these local bounds gives a global lower bound.

4.3.1.1 Selection of disjoint clique-like subproblems

The main question arising is how to select the disjoint clique-like subproblems on which to calculate the local lower bounds. Our strategy is to start with an empty set and to insert iteratively new disjoint cliques into it. To do this we fix a set K of separation values and a set H of penalty levels and at each iteration, $\forall k \in K$ and $\forall h \in H$, the level-k-h cliques of the graph not yet covered by the already selected clique subproblems are generated, and the most promising of them is selected.

Considering at each iteration such a large number of clique subproblems would appear to generate a very time consuming algorithm. Fortunately this is not true. At each iteration, for a fixed pair (k,h), all the level-k-h cliques can be retrieved by running the algorithm described in Bron and Kerbosch [25] on the problem obtained from the original one by ignoring the vertices covered by clique subproblems selected in the previous iterations (and the

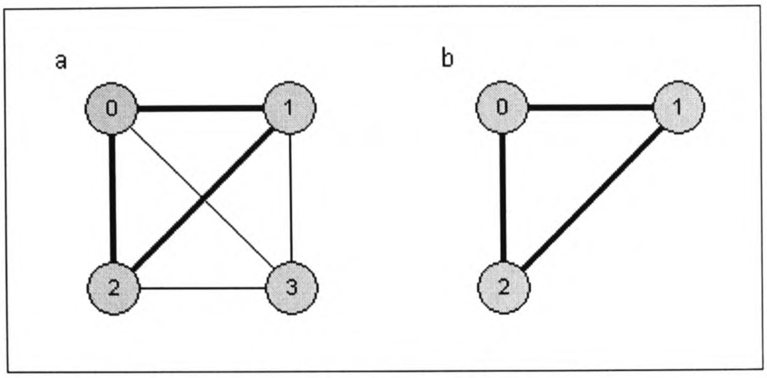

Figure 4.2: Role of level-k-h cliques.

edges involving them) and all the edges with separation less than k or penalty less than h. The key point of this strategy is that the algorithm of Bron and Kerbosch appears to be extremely fast on the graphs associated with realistic frequency assignment problems.

The choice of considering level-k-h for some different values of k instead of level-0-h cliques only (for a given value of h), can be justified with the help of the example in Figure 4.2.

In part a we have a level-0-1 clique with four vertices $(0, 1, 2, 3)$ with $d_{03} = d_{13} = d_{23} = 0$ (thin edges) and $d_{01} = d_{02} = d_{12} = 2$ (bold edges). There are six available frequencies ($F = \{0, 1, 2, 3, 4, 5\}$) and $\forall v, w = 0, 1, 2, 3, v < w$ $p_{vw} = 1$. Both the *TSP* bound and the *formula* bound produce a lower bound of 0 for this problem. In part b of the figure we consider a level-2-1 clique which is a subclique of the level-0-1 clique in part a. On this problem the lower bound obtained by both the methods is 1. This happens because the *TSP* bound and the *formula* bound work better on smaller problems with more homogeneous values of the d's.

Generalising the example above we can conclude that consideration of level-k-h cliques for some different values of k can lead to a better lower bound.

Referring again to Figure 4.2, we can justify the choice of considering level-k-h cliques for some different values of h, instead of treating level-k-1 only (for a given value of k). This time we read the figure as the pictorial representation of a level-0-1 clique with four vertices $(0, 1, 2, 3)$ with $p_{03} = p_{13} = p_{23} = 1$ (thin edges) and $p_{01} = p_{02} = p_{12} = 10$ (bold edges). There are two available frequencies ($F = \{0, 1\}$) and $\forall v, w = 0, 1, 2, 3, v < w\ d_{vw} = 0$ (part a). A lower bound of 2 for the penalty paid in this problem is obtained by the *TSP* bound and, indirectly, by the *formula* bound[1]. In part b of the figure we consider a level-0-10 clique which is a subclique of the level-0-1 clique in part a. On this problem the lower bound obtained by both our methods is 10. This happens because the *TSP* bound and the *formula* bound work better on smaller problems with more homogeneous values of the p's.

Generalising the example above we can conclude that to consider level-k-h cliques for different values of h can lead to a better lower bound.

The sets K and H, which contain respectively the values of k and h for which level-k-h cliques are considered, are parameters of the method.

4.3.1.2 Algorithm to produce *LB1*

The heuristic algorithm we adopt to select the set of disjoint clique subproblems used to generate *LB1* can be summarised as follows. At every iteration, given the set K of separation levels considered and the set H of penalty levels considered, $\forall k \in K$ and $\forall h \in H$ we calculate the *TSP* bound and the *formula* bound on each level-k-h clique C of the not yet covered graph

[1] The *formula* bound produces a lower bound of 2 for the number of constraint violations. This means a lower bound of 2 for the penalty if we consider two of the edges with the smallest penalty as violated, accordingly to the solution of the respective problem *U2W*, which will be described in Section 4.3.1.2.1.

$R = \{V_R, E_R\}$. The result of the *formula* bound, which expresses a number of constraint violations, is converted into a penalty by solving the linear program described in the Section 4.3.1.2.1.

The clique C with the greatest local lower bound is selected (ties are broken by choosing the smallest clique) and its vertices are considered as covered in the following iterations. The algorithm stops when the last selected clique produces a lower bound of zero.

In Figure 4.3 we present pseudocode for the algorithm to calculate *LB1*.

4.3.1.2.1 Conversion of the number of constraint violations into a penalty

To convert the lower bound τ for the number of constraint violations returned by the *formula* bound into a lower bound for the penalty, we solve the linear program described below, where we refer to a clique $C = \{V_C, E_C\}$ and variables have the following meaning:

- z_{vw}: continuous variable relaxed from a {0,1} variable. It is defined for every $\{v, w\} \in E$. A value of 1 means that the constraint between vertex v and vertex w is violated; a value of 0 means that the constraint is not violated.

$$(U2W) \quad \text{Min} \sum_{\{v,w\} \in E_C} p_{vw} z_{vw} \tag{4.23}$$

$$\text{s.t.} \sum_{\{v,w\} \in E_C} z_{vw} \geq \tau \tag{4.24}$$

$$0 \leq z_{vw} \leq 1 \qquad \forall \{v, w\} \in E_C \tag{4.25}$$

In practice, $U2W$ returns the sum of the τ smallest penalties of clique C.

```
*LB1 (Pr)*

INPUT:
$Pr$ = problem of type FS-FAP.
OUTPUT:
a lower bound of the cost of an optimal solution of $Pr$.

$R$ := graph representing $Pr$;
$DC := \emptyset$;
While($V_R \neq \emptyset$)
        $MC := \emptyset$;
        For $k \in K$
                For $h \in H$
                        $MC := MC \cup$ set of all level-$k$-$h$ cliques of $R$;
                EndFor
        EndFor
        $C$ := element of $MC$ with the greatest lower bound;
        $DC := DC \cup \{C\}$;
        If(local lower bound on $C = 0$)
                $V_R := \emptyset$;
        Else
                $V_R := V_R \backslash V_C$;
                $E_R := E_R \backslash \{\{v, w\} | v \in V_C$ or $w \in V_C\}$;
        EndIf
EndWhile
Return $\sum_{C \in DC}$ (local lower bound on $C$);
```

Figure 4.3: Algorithm for lower bound *LB1*.

4.3.2 Lower bound *LB2*

The global lower bound described in this section, which we call *LB2*, is based, like *LB1*, on the application of the local lower bounds described in Section 4.2 to level-k-h clique subgraphs of the original problem. In this case the local bounds are used to produce valid inequalities with which we reinforce the linear relaxation of the integer program $FAP3$ (described in Section 2.2.3). Formally we start by presenting the linear relaxation of $FAP3$ and then describing the inequalities derived from the local lower bounds. Finally we propose the algorithm that produces the global lower bound.

4.3.2.1 $FAP3_{LR}$: the linear relaxation of $FAP3$

A common way to obtain a lower bound for the solution of an integer program is to relax the integrality constraints into continuous constraints and to solve the linear program so obtained. The linear programming relaxation of formulation $FAP3$, which we will refer as $FAP3_{LR}$, is as follows:

$$(FAP3_{LR})\ \text{Min} \sum_{\{v,w\}\in E} p_{vw}x^1_{vw} \tag{4.26}$$

$$\text{s.t. } x^0_{vw} + x^1_{vw} + x^2_{vw} = 1 \qquad \forall\{v,w\}\in E \tag{4.27}$$

$$y_v - y_w \geq (d_{vw}+1)x^0_{vw} - d_{vw}x^1_{vw} - (|F|-1)x^2_{vw} \quad \forall\{v,w\}\in E \tag{4.28}$$

$$y_v - y_w \leq (|F|-1)x^0_{vw} + d_{vw}x^1_{vw} - (d_{vw}+1)x^2_{vw} \quad \forall\{v,w\}\in E \tag{4.29}$$

$$0 \leq y_v \leq |F|-1 \qquad \forall v\in V \tag{4.30}$$

$$0 \leq x^0_{vw} \leq 1 \qquad \forall\{v,w\}\in E \tag{4.31}$$

$$0 \leq x^1_{vw} \leq 1 \qquad \forall\{v,w\}\in E \tag{4.32}$$

$$0 \leq x^2_{vw} \leq 1 \qquad \forall\{v,w\}\in E \tag{4.33}$$

It is the same as $FAP3$ with the exception of constraints (4.30), (4.31), (4.32) and (4.33), which substitute (2.16), (2.17), (2.18) and (2.19).

It is interesting to observe that, as the penalties p_{vw} are defined as integers (see Section 2.1), every solution of $FAP3$ is integer. Consequently if we round the cost of a solution of $FAP3_{LR}$ to the integer above, we still have a lower bound of the cost of the solutions of $FAP3$. This rounding process will be implicitly applied to each modification of $FAP3_{LR}$ in the rest of this thesis.

It is well known (see Aardal et al. [1], Koster [60] and Montemanni [76]) that linear relaxation techniques provide very poor bounds (practically always 0) for all of the mathematical formulations described in Chapter 2. Our target is to define some valid inequalities to reinforce these linear relaxations. We have decided to work on $FAP3$ in this thesis because, as observed in Section 2.2.4, it is the most tractable formulation in terms of dimensions.

4.3.2.2 Reinforcing $FAP3_{LR}$

The inequalities we present are defined on a level-k-h (non-maximal) clique subproblem $C = \{V_C, E_C\}$ of G. They have the following forms:

$$\sum_{\{v,w\}\in E_C} x^1_{vw} \geq \tau \qquad (4.34)$$

$$\sum_{\{v,w\}\in E_C} p_{vw} x^1_{vw} \geq \delta \qquad (4.35)$$

Formally, inequality (4.34) forces the number of constraints violated in C to be at least τ, while (4.35) forces the penalty paid in C to be at least δ. δ can be obtained using the *TSP* bound on the subproblem with the original penalties, while τ can be obtained by applying the *formula* bound or by applying the *TSP* bound on a problem where all the penalties have been changed to 1. In the remainder of the thesis we will refer to the *TSP* bound applied to the problem obtained by reducing the original penalties to

1 as the *unweighted TSP* bound. Analogously we will sometimes refer to the *TSP* bound applied to the original problem as the *weighted TSP* bound.

4.3.2.2.1 Selection of relevant inequalities

The question of which inequalities, among all the possible ones, we should add to $FAP3_{LR}$ arises. After some tests it has become clear that, because of the relative small dimensions of $FAP3_{LR}$, it is possible to handle the linear program reinforced with a great number of relevant inequalities. We then consider all the level-k-h cliques which arise from a suitable definition of the separations set K and of the penalties set H. All the relevant inequalities calculated on them will be added to $FAP3_{LR}$.

Given a clique subproblem on which either the bound on the number of constraint violations or the bound on the penalty is non-zero, an important question is about which constraints should be added to $FAP3_{LR}$: constraint (4.34) only, constraint (4.35) only or both? The answer, together with the consequent choice criterion, is given in the remainder of this section.

We consider the example of Figure 4.4, where a clique C (subproblem of a bigger problem G) with five vertices $(0, 1, 2, 3, 4)$ is depicted. Each thin edge has $d = 0$ and $p = 1$ ($d_{12} = d_{13} = d_{14} = d_{23} = d_{24} = d_{34} = 0$ and $p_{12} = p_{13} = p_{14} = p_{23} = p_{24} = p_{34} = 1$), while each bold edge has $d = 1$ and $p = 3$ ($d_{01} = d_{02} = d_{03} = d_{04} = 1$ and $p_{01} = p_{02} = p_{03} = p_{04} = 3$). There are three available frequencies ($F = \{0, 1, 2\}$).

If we calculate the local lower bound on the number of constraint violations on C, we obtain a bound of 2, which would generate a constraint of type (4.34) with $\tau = 2$. Calculating the weighted *TSP* bound, we obtain a lower bound of 3 for the penalty paid in C, which can be used as δ for the respective constraint of type (4.35).

We now study how the linear program $FAP3_{LR}$ reacts when the two possible inequalities arising from C are added to it. In the following paragraphs some configurations of the x^1 variables corresponding to the edges of C are considered. These configurations do not represent feasible frequency assignments for C, but they can exist within a solution of $FAP3_{LR}$ because $FAP3_{LR}$ is a simplification (i.e. the linear relaxation) of the representation of G in terms of integer programming.

If we add to $FAP3_{LR}$ only the constraint (4.34) with $\tau = 2$, a solution where the only violated edges of C are $\{1,4\}$ and $\{2,3\}$, which produces a penalty of 2, would be feasible[2]. This solution would violate the constraint (4.35) which would force the solution to have a penalty at least equal to 3.

By contrast, adding the constraint (4.35) on the penalty paid on C (with $\delta = 3$) only, a solution where the only violated edge of C is $\{0,1\}$ would be feasible for $FAP3_{LR}$ strengthened by (4.35)[2]. This solution would produce just one constraint violation, and this would violate the constraint (4.34), where $\tau = 2$.

By extension from the example above we can conclude that there is not a dominance between the two types of constraints, and that we should work with both of them, selecting each time whether to add just one of them or both.

The formal criterion we adopt to choose which constraints to add for a given clique $C = \{V_C, E_C\}$ derives from the theorems which follow. To simplify the description of these, we give the following definition:

$$g_i = \left(\begin{array}{c} (i+1)\text{-th greatest penalty in} \\ \text{the problem represented by } C \end{array} \right) \tag{4.36}$$

Thus $g_0 \geq g_1 \geq \cdots \geq g_{|E_C|-1}$, where each g_i corresponds to p_{vw} for some

[2] In this solution of the strengthened $FAP3_{LR}$ some x^0 and x^2 variables will be fractional.

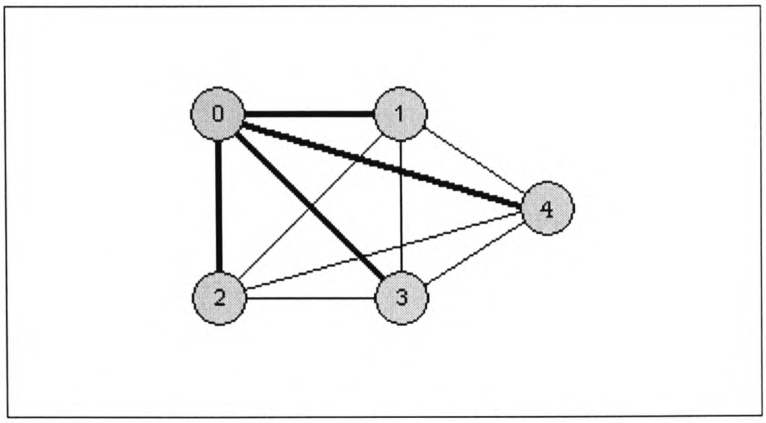

Figure 4.4: Constraints selection criterion.

edge $\{v, w\} \in E_C$.

We refer to τ as the lower bound for the number of constraint violations in C and to δ as the lower bound for the penalty paid in C.

Theorem 3. *If $\tau > 0$ and the condition*

$$\sum_{i=0}^{\tau-2} g_i \geq \delta \tag{4.37}$$

is satisfied on a clique C, then the constraint of type (4.34) generated from C is not dominated by the constraint (4.35) generated from C.

Proof. If the $\tau - 1$ edges with highest penalties are violated, then the constraint (4.34) generated from C will not be satisfied, although the constraint (4.35) generated from C is satisfied. □

Theorem 4. *If $\delta > 0$ and the condition*

$$\sum_{i=|E_C|-\tau}^{|E_C|-1} g_i < \delta \tag{4.38}$$

is satisfied on a clique C, then the constraint of type (4.35) generated from C is not dominated by the constraint (4.34) generated from C.

```
AddCtrs($C$, $LP$, $\tau$, $\delta$)

INPUT:
$C$ = clique subproblem.
$LP$ = (reinforced) linear program of type $FAP3_{LR}$.
$\tau$ = lower bound for the no. of ctr. violations in $C$;
$\delta$ = lower bound for the penalty paid in $C$;

If (($\tau > 0$) or ($\delta > 0$))
    If(($\tau > 0$) and (condition (4.37) is satisfied))
        add the constraint (4.34) generated from $C$;
    EndIf
    If(($\delta > 0$) and (condition (4.38) is satisfied))
        add the constraint (4.35) generated from $C$ to $LP$;
    EndIf
    If(conditions (4.37) and (4.38) are not satisfied)
        add the constraint (4.34) generated from $C$ to $LP$;
    EndIf
EndIf
```

Figure 4.5: Selection of the constraints to generate from a clique-like subproblem.

Proof. If the τ edges with lowest penalties are violated, then the constraint (4.35) generated from C will not be satisfied, although the constraint (4.34) generated from C is satisfied. □

The criterion arising from these theorems is described in the procedure whose pseudocode is presented in Figure 4.5. We adopt the convention that, if both condition (4.37) and condition (4.38) do not apply for a clique C (i.e. the constraints (4.34) and (4.35) calculated on C give the same information), then we add only constraint (4.34).

```
LB2(Pr)

INPUT:
$Pr$ = problem of type FS-FAP.
OUTPUT:
a lower bound of the cost of an optimal solution of $Pr$.

$LP$ := linear program $FAP3_{LR}$ representing $Pr$;
For $k \in K$
    For $h \in H$
        $CS_{kh}$ := set of all level-$k$-$h$ cliques of $G$;
        $\forall\, C \in CS_{kh}$
            $\tau$ := lower bound for the number of constraint violations in $C$;
            $\delta$ := lower bound for the penalty paid in $C$;
            AddCtrs($C$, $LP$, $\tau$, $\delta$);
        End$\forall$
    EndFor
EndFor
Return $\lceil$ Cost(optimal solution of $LP$) $\rceil$;
```

Figure 4.6: Algorithm for lower bound *LB2*.

4.3.2.3 Algorithm to produce *LB2*

In Figure 4.6 we present pseudocode for the algorithm which produces the global lower bound *LB2*. It is the result of the considerations previously illustrated.

4.3.3 Lower bound *LB3*

This lower bound is an extension of the lower bound *LB2* obtained by considering not only maximal level-k-h cliques, but also non-maximal cliques.

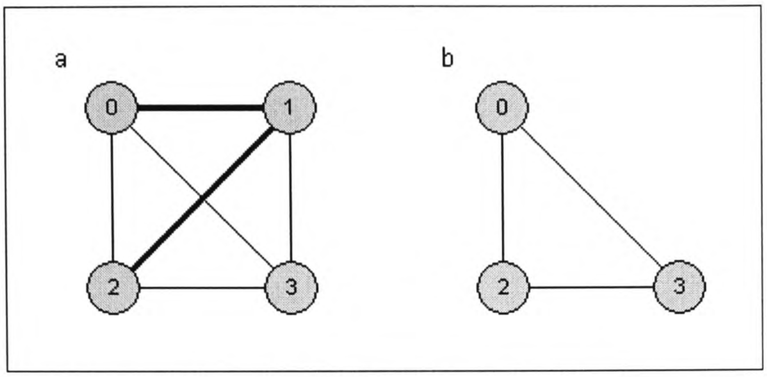

Figure 4.7: Role of non-maximal cliques.

4.3.3.1 Role of non-maximal cliques

A question that arises is whether adding inequalities calculated on non-maximal cliques to $FAP3_{LR}$ can improve the results obtained by adding to $FAP3_{LR}$ only inequalities derived from (maximal) level-k-h cliques.

The answer is easy: non-maximal cliques can give more information than level-k-h cliques only, and this is clear from the example described in Figure 4.7.

In Figure 4.7 part *a* we consider a clique on which we calculate the *TSP* bound and the *formula* bound. The clique has four vertices $(0,1,2,3)$, two available frequencies for each vertex ($F = \{0,1\}$), $\forall v,w \in \{0,1,2,3\}$, $v < w$ $d_{vw} = 0$ and $p_{vw} = 1$. Both our approaches produce a lower bound of 2, generating the following reinforcing inequality for $FAP3_{LR}$:

$$x^1_{01} + x^1_{03} + x^1_{04} + x^1_{12} + x^1_{13} + x^1_{23} \geq 2 \tag{4.39}$$

Now we consider the part *b* of Figure 4.7, in which we have a non-maximal clique which is a subproblem of the clique in part *a*. The reinforcing inequality derived by the *TSP* bound and the *formula* bound for this subproblem is:

$$x_{02}^1 + x_{03}^1 + x_{23}^1 \geq 1 \tag{4.40}$$

In the figure a solution of $FAP3_{LR}$ with the following values for x^1 variables is highlighted in bold:

$$x_{01}^1 = x_{12}^1 = 1 \qquad x_{02}^1 = x_{03}^1 = x_{13}^1 = x_{23}^1 = 0$$

This is a valid solution[3] for constraint (4.39), but it does not respect constraint (4.40). Adding the inequality calculated on the non-maximal clique in part *b* of the figure, can improve the result obtained by adding only the inequality derived from the level-0-1 clique in part *a*, and consequently, by extension from this particular case, we can conclude that adding non maximal cliques can help to obtain better estimates. Our results will also support this conclusion.

In particular, in the example of Figure 4.7 the main limitation of our inequalities is clear: they specify the penalty (number of violations) that must be present in a subproblem, but they are unable to describe relations among violated constraints. As also the global formulation $FAP3_{LR}$ is generally not able to specify these relations, the violated constraints often assume infeasible configurations.

4.3.3.2 Selection of non-maximal cliques

It is clearly impossible to consider all of the non-maximal cliques of a graph. For this reason we have adopted a heuristic criterion to select the most promising of them. The algorithm which implements this criterion takes into account the last available solution of the reinforced linear program $FAP3_{LR}$.

[3] Notice that in such a solution we will have $x_{02}^0 = x_{02}^2 = 0.5$.

Given a level-k-h (non maximal) clique $C = \{V_C, E_C\}$, a chain of non-maximal cliques which are subsets of C is created by the algorithm. The chain starts from the empty set and stops when the maximal clique C is reached. At each iteration, given the current non-maximal clique $S = \{V_S, E_S\}$, we select probabilistically a vertex to be added to S. We need the following definition to specify how the probability is calculated:

$$q(v,w) = \left(\begin{array}{c} \text{number of non-maximal cliques involving the edge} \\ \{v,w\} \text{ already considered by the algorithm} \end{array} \right) \tag{4.41}$$

The probability of selecting a vertex v (not yet in V_S) is then given by:

$$\rho_1(v) = \frac{\displaystyle\sum_{\substack{w \in V_S; \\ v<w}} \frac{2-\overline{x}^1_{vw}}{q(v,w)} + \sum_{\substack{w \in V_S; \\ w<v}} \frac{2-\overline{x}^1_{wv}}{q(w,v)}}{\displaystyle\sum_{\substack{u \in V_C \setminus V_S; \\ u \neq v}} \left(\sum_{\substack{w \in V_S; \\ u<w}} \frac{2-\overline{x}^1_{uw}}{q(u,w)} + \sum_{\substack{w \in V_S; \\ w<u}} \frac{2-\overline{x}^1_{wu}}{q(w,u)} \right)} \tag{4.42}$$

where $\overline{x}^1_{vw}$ indicates the value of variable x^1_{vw} in the last available solution of the (reinforced) formulation $FAP3_{LR}$.

The strategy described gives priority to those vertices which are connected to S with edges that are not violated in the last solution available and that have not been considered too many times in the previous selections of non-maximal cliques.

The pseudocode for the heuristic criterion described in this section is incorporated into the pseudocode for the lower bounding technique $LB3$, which is presented in Figure 4.8.

4.3.3.3 Algorithm to produce *LB3*

In this section we describe the algorithm to produce the global lower bound *LB3*, which incorporates inequalities arising from non-maximal clique subproblems. The pseudocode is given in Figure 4.8.

```
LB3(Pr)

INPUT:
$Pr$ = problem of type FS-FAP.
OUTPUT:
a lower bound of the cost of an optimal solution of $Pr$.

$LP$ := linear program $FAP3_{LR}$ representing $Pr$;
For $k \in K$
  For $h \in H$
    $CS_{kh}$ := set of all level-$k$-$h$ cliques of $G$;
    $\forall\ C \in CS_{kh}$
      $\tau$ := lower bound for the number of constraint violations in $C$;
      $\delta$ := lower bound for the penalty paid in $C$;
      AddCtrs(C, LP, $\tau$, $\delta$);
    End$\forall$
  EndFor
EndFor
While(time < $T_{max}$)
  $\forall\ C \in$ {set of all level-0-1 cliques of $G$}
    If (random number in [0,1) < $\rho_2$)
      $S := \{\emptyset, \emptyset\}$;
      While($S \neq C$)
        $v$ := vertex in $V_C \backslash V_S$ chosen probabilistically (eqn. (4.42));
        $E_S := E_S \cup$ {edges involving $v$ and the vertices already in $V_S$};
        $V_S := V_S \cup \{v\}$;
        $\tau$ := lower bound for the number of constraint violations in $S$;
        $\delta$ := lower bound for the penalty paid in $S$;
        If($\tau \leq$ no. of ctr. violations in $S$ in the last solution of $LP$)
          $\tau := 0$;
        EndIf
        If($\delta \leq$ penalty in $S$ in the last solution of $LP$)
          $\delta := 0$;
        EndIf
        AddCtrs(C, LP, $\tau$, $\delta$);
      EndWhile
    EndIf
  End$\forall$
  update the solution of $LP$;
EndWhile
Return $\lceil$ Cost(optimal solution of LP) $\rceil$;
```

Figure 4.8: Algorithm for lower bound *LB3*.

The first phase of the algorithm is the same as the algorithm to produce $LB2$, described in Figure 4.6.

In the second phase the algorithm enters into an iterative statement. Here at every iteration, each level-0-1 clique C is selected with a given probability ρ_2. If C is selected, we generate a chain of non-maximal cliques, where each one of them differs from the previous one by a new vertex v, added with probability given by (4.42). We calculate all the constraints obtained from these non-maximal cliques and we add to LP the ones which are violated in the last solution of LP. LP is the representation through $FAP3_{LR}$ of the *FS-FAP* Pr, reinforced with some constraints of type (4.34) and (4.35). The choice of the type of constraint to add is done following the criterion described in the pseudocode of Figure 4.5.

A new solution of LP is calculated after consideration of all the selected level-0-1 cliques.

The algorithm exits from the iterative statement and stops when a maximum computation time T_{max} has been reached.

It is important to notice that we may add redundant constraints to LP, because we do not update the solution of LP too often. We choose to do so because to prevent this would be more expensive, in computational terms, than dealing with redundant constraints. In particular, the probability ρ_2 has been inserted to give a tradeoff between the frequency of solution of LP and the probability of having redundant inequalities.

4.4 Computational results

In this section we group some computational results obtained by the methods described in this chapter. We also present a brief analysis of those situations in which our methods do not obtain tight estimates.

All of the linear programs arising within this thesis have been solved using ILOG CPLEX[4] callable libraries version 6.6.

4.4.1 Local lower bounds

In this section we study the results obtained by the two local lower bounding techniques described in Section 4.2. We present the results achieved by the two methods on some of the benchmarks described in Section 3.4.1.

A study of the computation times of the two local lower bounds is omitted as they are always negligible.

We present charts where in the x axis some of the level-0-1 cliques (with a non-null local upper bound) of a given problem are represented. By a *local upper bound* we mean an upper bound for the penalty paid in a given clique subproblem, ignoring the rest of the problem. Each local upper bound is calculated by running the tabu search algorithm described in Section 3.3.2 with appropriate parameter settings. We cannot guarantee the optimality of the upper bounds, and on the contrary we suspect that it is not always reached.

Each chart of this section contains the following values:

- *UB*: upper bounds of the cost paid in the clique when the rest of the problem is ignored (local upper bounds);

- *Formula LB*: local lower bounds obtained by the *formula* bound. As the *formula* bound returns an estimate of the number of constraint violations, in the case of weighted problems we convert the results into

[4] http://www.cplex.com.

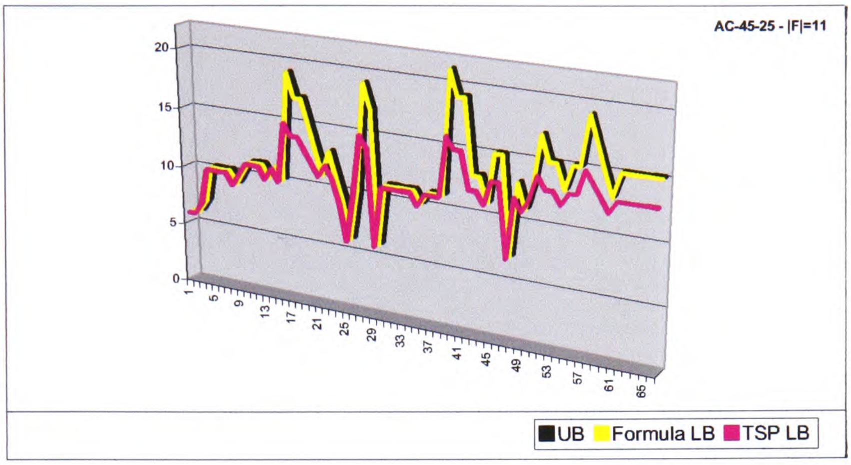

Figure 4.9: Local lower bounds on *AC-45-25* with $|F| = 11$.

a bound on the penalty by solving the respective linear programs $U2W$, introduced in Section 4.3.1.2.1;

- *TSP LB*: local lower bounds obtained by the *TSP* bound;

We have subdivided the results accordingly to the benchmark families.

The target of the problems of the first family is to minimise the number of constraint violations (i.e. $\forall\{v,w\} \in E\ p_{vw} = 1$). In Figure 4.9 we present a chart summarising the results obtained on the level-0-1 cliques of the problem *AC-45-25* with $|F| = 11$ (this information is reported in the top right corner).

In Figure 4.9 the *formula* lower bound dominates the *TSP* bound and matches the upper bound for all the subproblems. The good results of the *formula* bound, which works by reducing all the separations to the smallest of them, suggest that the problem examined is quite regular in terms of separations. It is also interesting to observe how the *TSP* bound tends to underestimate the number of constraint violations when there are many,

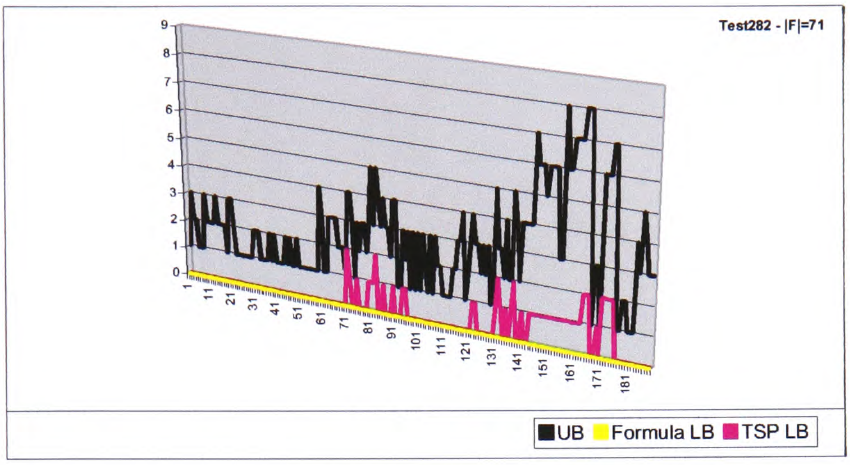

Figure 4.10: Local lower bounds on *Test282* with $|F| = 71$.

confirming that the method works better when there are few violations.

A different situation is depicted in Figure 4.10, where the results obtained for a subset of the level-0-1 cliques of the problem *Test282* with $|F| = 71$ are presented.

The gap between the upper bound and the *TSP* bound is not closed and the *formula* bound always produces a bound of zero. The bad results of the *formula* bound may suggest a very scattered distribution of the separation values.

The two examples above suggest that the characteristics of the problems can heavily influence the performance of our local lower bounds.

In Figure 4.11 we present the results obtained on a weighted problem from the second family of benchmarks. The problem considered is *GSM2-184*, and $|F| = 39$. We can see how the *TSP* bound dominates the *formula* bound, suggesting that the TSP bound has better performance than the (converted) *formula* bound on weighted problems.

The last tests presented concern the problems of the third set of bench-

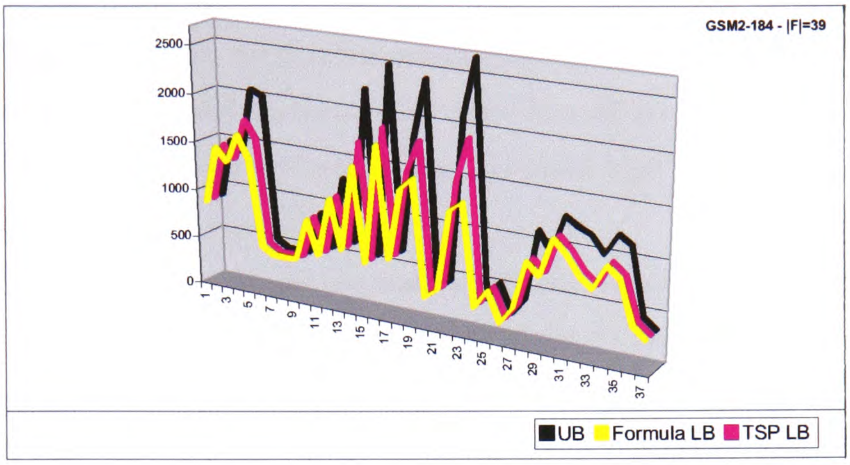

Figure 4.11: Local lower bounds on *GSM2-184* with $|F| = 39$.

marks. We want to study how the quality of the local lower bounds changes when some of the characteristics of the problems change. In particular we study how the bounds are affected by changes in the domain dimension, by changes in the range of the separation values and by changes in the range of the penalty values. We ignore the modification of the edge density because it would not be significant in terms of local problems.

In Figure 4.12 we group the results obtained on the same problem, when the number of available channels changes. In the first graph a domain of 15 frequencies is considered. Only a small fraction of the level-0-1 cliques have a non-zero upper bound and in these cases both the local bounds find the optimal solutions. Considering $|F| = 10$ (second graph) a gap between the upper bound and the lower bounds is present and occurs particularly when the upper bound is high. The *TSP* bound dominates the *formula* bound. In the graph in the bottom of the figure we consider just 5 available frequencies, and the lower bounds obtain poor results, particularly the *TSP* bound. This does not work properly when, as in this case, the density of violations is very high (the reason for this has been explained in Section 4.2.1.4). We can conclude that the performances of our lower bounds are heavily influenced by the domain size: underestimated domains mean high density of violations, which implies poor results.

In Figure 4.13 we group the results obtained on a set of graphs which have the same structure but differ in the range of the d's (separation values). The maximum possible value of separation (the minimum is always 0) is reported in the top left corner of each of the three graphs. The gap between the upper and the lower bounds increases as the range increases. It is important to observe how the lower bounds do not change too much from one graph to another, while the upper bounds become higher and higher: this means that our methods tend not to capture the increase of the size of the maximum d value. Another observation is about the fact that increasing the separation values range without augmenting the available frequencies, makes the density of the violations become higher, producing a result similar to the one obtained by reducing the number of frequencies available on a fixed problem. This could suggest that the degradation of the performance with the increase of the range of the d's could depend also on the augmentation of the density of violated edges, which, we have seen, makes the quality of our lower bounds decrease.

Finally, in Figure 4.14 we present the results obtained when the range of the possible values for the p's (penalty values) is modified on a graph which maintains the other characteristics unchanged. The degradation in the performance of our local lower bounds is clear, especially for the *formula* bound. Also in this case it is interesting to observe how the lower bounds do not capture too much of the increase of the upper bounds when the range of the penalty values becomes bigger, producing an effect similar to the one

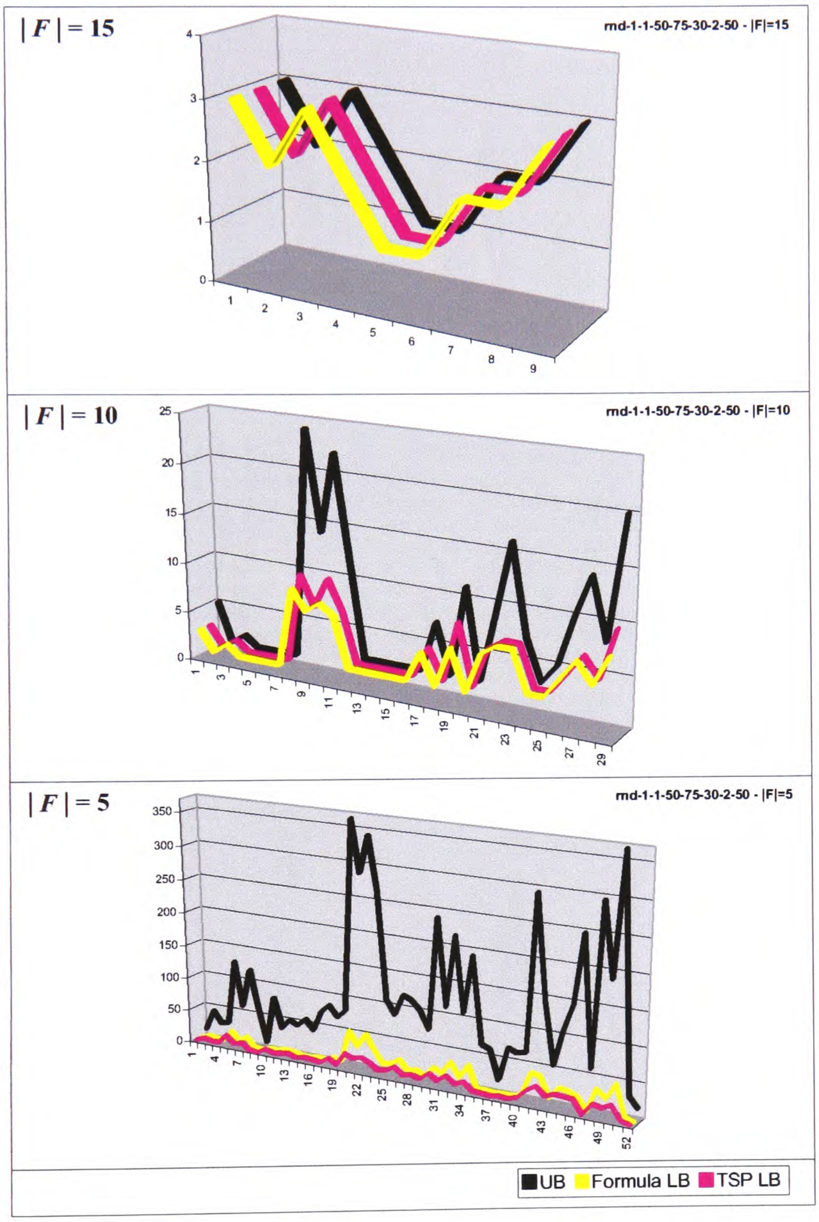

Figure 4.12: Performance of the local lower bounds when the frequency domain is changed.

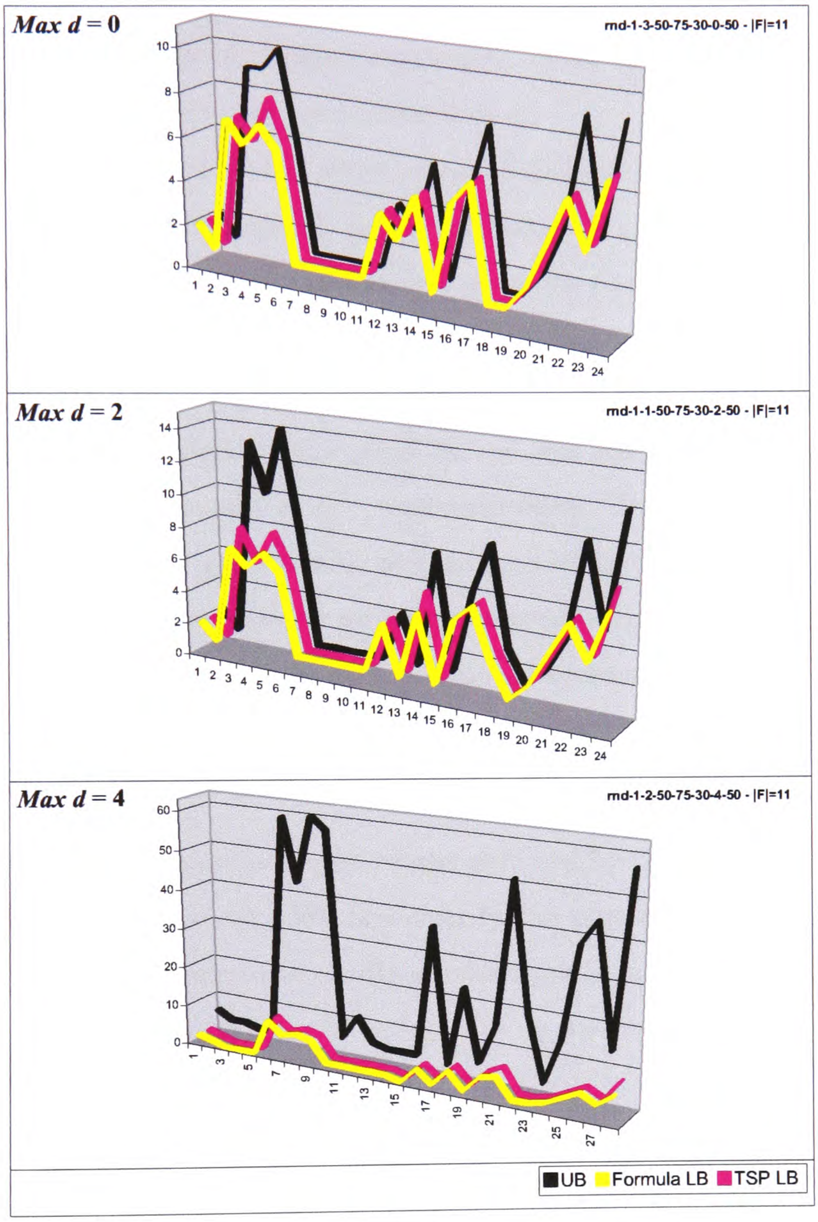

Figure 4.13: Performance of the local lower bounds when the range of the separations is changed.

seen in Figure 4.13. In this case bigger ranges for the p's do not directly imply higher density of violated constraints, so we can conclude that bigger ranges for p's imply worse local lower bounds.

We can conclude that the performance of the local lower bounds described in Section 4.2 heavily depends on the characteristics of the problems. They seem to work very well on the simplest problems (i.e. the average of the d's is not too far from 0, the average of the p's is not too far from 1 and there is an adequate frequency domain), while they have more difficulty when the separations, and especially the penalties, are very scattered or the dimension of the frequency domain is very underestimated. The quality of our lower bound seems to be particularly affected by this last factor. This is not a dramatic problem because, as we have observed in Section 4.2.1.4, a heavily underestimated frequency domain is quite uncommon.

4.4.2 Global lower bounds

In this section we summarise the results obtained by the global lower bounds described in Section 4.3. We first specify the parameter settings we have chosen, then we present the results obtained and a brief study of them.

4.4.2.1 Parameter settings

In this section we specify the values we have chosen for the parameters of the methods $LB1$, $LB2$ and $LB3$.

The sets K and H, which contain respectively the values of k (separations) and h (penalties) we will consider for level-k-h cliques have been set as described below based on a series of tests. These suggested that K and H are not crucial parameters, as the lower bounds (especially $LB3$) are not too much affected by different settings of them. However a good tradeoff

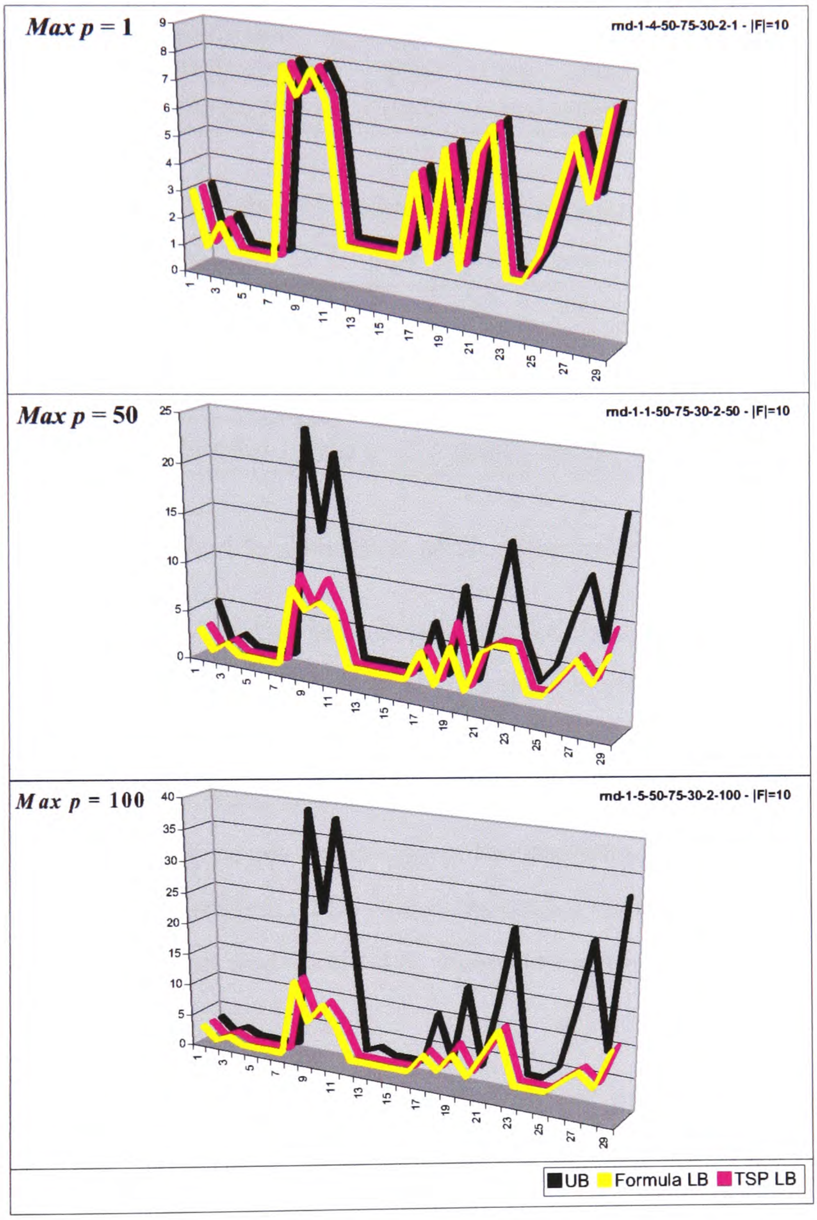

Figure 4.14: Performance of the local lower bounds when the range of the penalties is changed.

Graph	1st	2nd	3rd	4th	5th
GSM2-184	1	2000	4000	6000	8000
GSM2-227	1	2000	4000	6000	8000
GSM2-272	1	2000	4000	6000	8000

Table 4.2: Definition of H. Benchmark set 2.

Graph	1st	2nd	3rd	4th	5th
1-1-50-75-30-2-50	1	10	20	30	40
1-2-50-75-30-4-50	1	10	20	30	40
1-3-50-75-30-0-50	1	10	20	30	40
1-5-50-75-30-2-100	1	20	40	60	80
1-6-50-75-30-0-10000	1	2000	4000	6000	8000

Table 4.3: Definition of H. Benchmark set 3.

between the quality of the bounds and the computational times seems to be reached when K includes all of the levels of separation which appear in the problem (for a frequency assignment problem typically there are no more than five levels) and when H includes five levels of penalty[5]. Consequently K is given by all of the values appearing in the problem, while the definition of H depends on the problem. The values chosen for each weighted problem are specified in Table 4.2 and Table 4.3, where the columns have the following meaning:

- Graph: names of the scenarios. For all the problems derived from a scenario (by defining different frequency domains), the set H contains the same values;

- i^{th}: i^{th} value contained in the set H for each scenario.

It is important to observe that the quality of the lower bounds seems to depend more on the choice of the set K than on the choice of the set H.

[5] This is valid for the weighted problems only. For unweighted problems $H = \{1\}$.

The lower bound $LB3$ has two other parameters: the probability ρ_2, which is used in the second phase of the algorithm and the maximum computation time T_{max}. T_{max} has been fixed at 24 hours and ρ_2 at 0.75. It must be observed that some preliminary tests suggested that ρ_2 is not a crucial parameter for the quality of the lower bounds.

4.4.2.2 Results

In Table 4.4, Table 4.5 and Table 4.6 we summarise the results obtained by the global lower bounding techniques $LB1$, $LB2$ and $LB3$. The columns of the tables have the following meaning:

- Problem: names of the problems. Each name is composed of the following two elements:
  - Graph: name of the graph on which the problem is based;
  - $|F|$: number of channels available;
- UB: best upper bound available for each problem. The bounds are obtained using the heuristic algorithms described in Chapter 3;
- LBi: lower bounds obtained by the method LBi. The two subcolumns have the following meaning:
  - Val: values of the lower bounds;
  - Sec: computation times in seconds.

The results summarised in Table 4.4, Table 4.5 and Table 4.6 can be considered very satisfactory, because the lower bounds are on average good and the methods presented are, as far as we know, the first to work on general

Problem		UB	$LB1$		$LB2$		$LB3$	
Graph	$\|F\|$		Val	Sec	Val	Sec	Val	Sec
AC-45-17	7	32	16	1	16	1	20	31
AC-45-17	9	15	9	1	9	1	10	4
AC-45-25	11	33	21	1	21	2	26	986
AC-45-25	19	8	8	2	8	3	8	3
AC-95-9	6	31	20	1	24	1	27	17
AC-95-9	10	3	3	1	3	1	3	1
AC-95-17	15	33	21	19	25	68	28	738
AC-95-17	21	10	9	19	9	22	9	22
GSM-93	9	32	14	3	15	2	17	32
GSM-93	13	7	4	2	4	2	4	2
GSM-246	21	79	47	14	47	28	50	5657
GSM-246	31	25	16	12	16	22	16	22
Test95	31	12	11	2	12	2	12	2
Test95	36	8	7	2	7	2	7	2
Test282	61	51	19	273	20	190	20	190
Test282	71	27	5	229	5	124	6	1030
P06-5	11	133	121	4	121	7	121	7
P06-5	41	15	15	5	15	7	15	7
P06-3	31	115	109	16	109	41	109	41
P06-3	71	26	26	17	26	31	26	31
P06b-5	21	52	49	5	49	9	49	9
P06b-5	31	25	25	5	25	10	25	10
P06b-3	31	112	106	15	106	59	106	59
P06b-3	71	26	26	17	26	47	26	47

Table 4.4: Lower bounds results. Benchmark set 1.

Problem		UB	$LB1$		$LB2$		$LB3$	
Graph	$\|F\|$		Val	Sec	Val	Sec	Val	Sec
GSM2-184	39	5521	1923	9	4816	88	4856	5035
GSM2-184	49	874	500	9	874	60	874	60
GSM2-227	39	10979	2708	17	7147	179	7328	16064
GSM2-227	49	2459	1125	16	1998	123	1998	123
GSM2-272	39	27416	5206	53	12792	265	12909	85131
GSM2-272	49	7785	2394	38	5168	270	6258	80951

Table 4.5: Lower bounds results. Benchmark set 2.

Problem Graph	$\|F\|$	UB	$LB1$ Val	 Sec	$LB2$ Val	 Sec	$LB3$ Val	 Sec
1-1-50-75-30-2-50	5	1242	247	2	519	4	802	65296
1-1-50-75-30-2-50	10	101	14	1	36	3	52	1879
1-1-50-75-30-2-50	11	59	10	1	28	2	36	616
1-1-50-75-30-2-50	15	11	4	2	11	3	11	3
1-2-50-75-30-4-50	11	323	25	3	54	3	68	5655
1-3-50-75-30-0-50	11	36	12	1	28	2	35	218
1-4-50-75-30-2-1	10	19	14	1	14	1	17	29
1-5-50-75-30-2-100	10	186	24	2	49	3	90	2259
1-6-50-75-30-0-10000	10	6942	1274	2	3083	3	6315	8891

Table 4.6: Lower bounds results. Benchmark set 3.

problems, i.e. without using particular characteristics of a given problem (set of problems).

Lower bound $LB3$ is the best one in terms of results, but has longer computation times. $LB1$ does not seem to be very promising: it is always dominated by $LB2$ (and consequently $LB3$), and on average the difference between the computation times of $LB1$ and $LB2$ is minimal. For this reason in the following we will not consider $LB1$ further. $LB2$ can be seen as a truncated version of $LB3$, and can be used as a first approximation of the lower bound in a short time.

The quality of the lower bounds is not constant over all of the examples and in particular the ratio $LB3/UB$ varies between 0.211 (*1-2-50-75-30-4-50* with $|F| = 11$) and 1 (many problems). We suspect that in some cases the upper bounds do not match the optimum (as observed in Section 3.4.2.2), and consequently we think that the quality of the lower bounds may be sometimes underestimated. It is anyway clear that, given a scenario, our three methods work better when the number of frequencies available is greater and the values of the separations (d's) are defined in small ranges (see Table 3.1, Table 3.2

and Table 3.3).

Observing that the global lower bounds seem to work better in the same conditions in which the local lower bounds obtain their best results (see Section 4.4.1), we can conclude that when the local lower bounds are not tight, as a consequence the global lower bounds do not seem to work effectively. The main question arising now is whether this is the only factor which affects the quality of the global lower bounds, or whether there are other reasons. We will discuss this in the following section, where we will try also to delineate some strategies to improve our global lower bounds.

There is another important observation about the results of Table 4.4, Table 4.5 and Table 4.6. For some problems the lower bounds match the upper bounds and this may suggest that the variables of the reinforced $FAP3_{LR}$ represent a feasible (optimal) assignment for the original *FS-FAP*. Unfortunately this is not true, and this certainly depends on the weakness of constraints (4.28) and (4.29) of formulation $FAP3_{LR}$, which are not strong enough to guarantee a feasible assignment, even when the global lower bounds are optimal. This property, which is intrinsic in our methods, is unfortunate because it makes it non trivial to use the solutions of the reinforced $FAP3_{LR}$ as starting points to generate good upper bounds.

4.4.2.3 Study of the results

In this section we investigate the results presented in the previous section. In particular we focus our attention on those situations where our methods have not obtained tight estimates. The most obvious explanation for these situations is given by the loss of information arising from the use of $FAP3_{LR}$ instead of $FAP3$. In the following of this section we will look for other, less obvious, explanations.

As anticipated in the previous section, sometimes the poor quality of the global lower bounds could be strictly connected with the poor quality of the local lower bounds. To examine this possibility, we have considered a modified version of $LB2$ and $LB3$, where values for the τ's and the δ's are obtained by running upper bounding techniques on clique like subproblems. The comparison of the results obtained by these modified methods with the original ones is presented in Table 4.7, where the columns have the following meaning:

- Problem: names of the problems. Each name is composed of the following two elements:
  - Graph: name of the graph on which the problem is based;
  - $|F|$: number of channels available;
- UB: best upper bound available for each problem, according to the results obtained by the heuristic algorithms described in Chapter 3;
- LBi: results obtained by the two versions of the method LBi. The two subcolumns have the following meaning:
  - LLB: results obtained by the original algorithm LBi;
  - LUB: results obtained by the modified version of the algorithm LBi. For each clique-like subproblem considered, local upper bounds are used instead of the *formula* and the *TSP* local lower bounds for the creation of the reinforcing constraints (4.34) and (4.35). The local upper bounds are calculated by running the tabu search algorithm described in Section 3.3.2 with appropriate parameter settings. Notice that the resulting global estimates cannot be read as lower bounds.

Problem Graph	$\|F\|$	UB	$LB2$ LLB	 LUB	$LB3$ LLB	 LUB
Test282	71	27	5	14	6	>15
1-2-50-75-30-4-50	11	323	54	138	68	>161
GSM-93	13	7	4	4	4	4

Table 4.7: Sensitivity of the global lower bounds to the quality of the local lower bounds.

Some values of the last column of the table are of type " $> x$ ". This means that when the computation has been truncated after 24 hours, the algorithm was still improving and the lower bound was x. These time consuming runs are due to the slowness of the method when the calculation of the local upper bounds is included.

In studying Table 4.7 it should be noted that two factors could perturb the results of the table. Both are connected with the quality of the upper bounds, local and global respectively, which cannot be guaranteed. All the conjectures we will make in the following are consequently based on the hypothesis that all the global and local upper bounds are not greatly overestimated. We remain uncertain about this hypothesis, but we must assume it for the considerations described in the following.

The results presented in Table 4.7 are very heterogeneous. For the problems *Test282* with $|F| = 71$ and *1-2-50-75-30-4-50* with $|F| = 11$, the poor quality of the global lower bound appears to be a direct consequence of the poor quality of the local lower bounds. The improvement of the result when local upper bounds are considered instead of local lower bounds is consistent, notwithstanding that a gap between the global lower bound and the global upper bound remains.

The results commented above can be read as an indication that, for the problems considered, the poor quality of the local lower bounds is an important factor in the quality of our global lower bounds, but it may not be the only one.

The last row of the table, which concerns problem *GSM-93* with $|F| = 13$, confirms this last conjecture. In this case the use of local upper bounds instead of local lower bounds does not affect the global result. This indicates that the quality of the local lower bounds is already very good, but this is not enough to guarantee a tight global lower bound. A second factor, independent of the quality of the local estimates, must consequently affect the quality of the global lower bounds. It seems to depend on something that cannot be captured by the single clique-like subproblems, probably on constraint violations which are caused by the structure of the whole problem (or of a subproblem of it bigger than each clique-like subproblem). For this reason we will refer to this factor as the *context problem* to indicate that it depends on the context in which the clique-like subproblems are inserted. This context is ignored by our techniques, which are based on clique-like subproblems only.

The chart presented in Figure 4.15 should help to understand the context problem. It refers to problem *GSM-93* with $|F| = 13$, and also for this example the unproved hypothesis that the global upper bound is not greatly overestimated must be assumed. In the x axis of the chart all of the level-k-h cliques considered by $LB2$ (see Section 4.4.2.1) with a non-null local upper bound are listed. For each one of these cliques we present the following values (we remind the reader that *GSM-93* is an unweighted problem, i.e. $\forall\{v, w\} \in E \; p_{vw} = 1$):

- Local penalty in the global UB: penalty paid within the subproblem in the solution of the whole problem which produces the best global upper bound available (obtained with the methods described in Chapter 3);

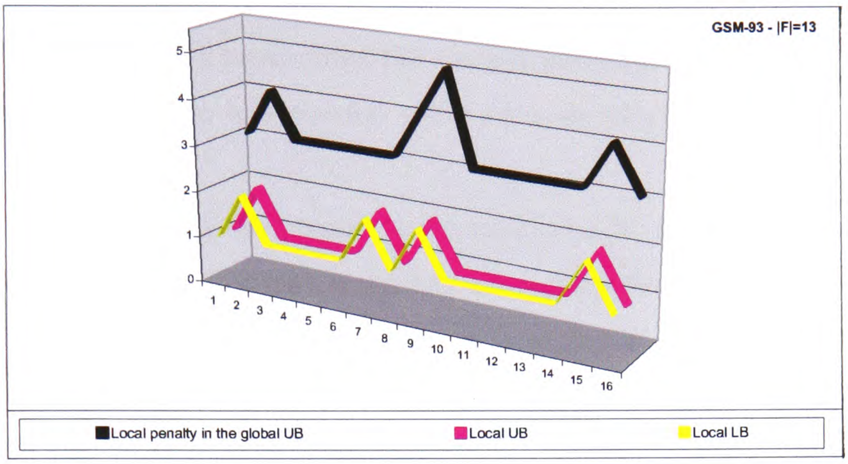

Figure 4.15: Context problem in *GSM-93* with $|F| = 13$.

- Local UB: local upper bound of the penalty paid in the subproblem, without considering the rest of the problem (obtained with the methods described in Chapter 3);
- Local LB: local lower bound of the penalty paid in the subproblem, without considering the rest of the problem. The highest value between those returned by the TSP bound and the *formula* bound is considered.

Analysing Figure 4.15 we can see how, for the problem represented, "Local LB" always matches "Local UB" (i.e. the local lower bounds are optimal), but the values of these series are always below the respective values of "Local penalty in the global UB" (i.e. there is a context problem).

In conclusion we think that when our lower bounds do not work satisfactorally, this is a result of at least three main factors. The first, which is the most obvious one because it is part of our approach to the problem, is the loss of information arising from the use of $FAP3_{LR}$ instead of $FAP3$. The second factor is the unsatisfactory quality of the local lower bounds, while the third is what we have called the context problem.

In Chapter 5 we will develop some methods which aim to improve the local lower bounds.

In Chapter 6 we will propose a simplification for the global formulation $FAP3$, together with some reinforcing inequalities for it. These inequalities represent structural information of the original frequency assignment problem which are lost when the linear relaxation of $FAP3$ is considered. In Chapter 6 we will also briefly describe some approaches we have developed to overcome the context problem, but which have proved to be completely ineffective.

Chapter 5

Improving the local lower bounding techniques

In Chapter 4 we have described some global lower bounding techniques, together with the local lower bounds on which they are based. In Section 4.4.2.3 we have observed that three main reasons can be identified to explain those situations where our global lower bounds are not very tight. One of these reasons is the poor quality of the local lower bounds in these situations. The aim of this chapter is to improve the quality of the local lower bounds.

The *formula* bound (Section 4.2.2) does not seem to have great margin for improvement. More potential seems to be offered by the *TSP* bound (Section 4.2.1). For this reason in this chapter we will work mainly on this method, describing two ideas which should reinforce the linear program TSP_{LR}, on which the TSP bound is based.

We also propose a non-trivial method to convert a lower bound for the number of constraint violations present in a problem into a lower bound for the penalty paid in the same problem.

The chapter is concluded by some computational results.

5.1 Improving the TSP lower bound

In this section we propose two families of constraints, *2-path* inequalities and *subtour elimination* inequalities, which should reinforce TSP_{LR}, the linear program solved to compute the TSP local lower bound.

5.1.1 *2-path* inequalities

The method described in this section is based on a reformulation of TSP_{LR} which includes new variables and new constraints. For simplicity we will refer to the approach as *2-path inequalities*, notwithstanding it is more than a simple new set of inequalities.

The aim of the method is to avoid the undesirable phenomenon described in the example of Figure 4.1, where a constraint violation, implicitly present in the assignment arising from the solution of TSP_{LR}, is ignored by the lower bound.

As stated in Section 4.2.1, the *TSP* lower bound for the *FS-FAP* has been created by converting the well-known TSP lower bound method for the *MS-FAP*, described in Allen et al. [8]. In that paper a set of inequalities developed to improve the *TSP* lower bound for the *MS-FAP* (called *frequency assignment inequalities*) is described (see also Allen et al. [9], Dunkin and Allen [36] and Smith et al. [86], [88] and [91]). In this section we present a family of inequalities for the *FS-FAP* version of the *TSP* bound which are based on the same idea. However, our inequalities look very different from those described in [8] because of the strong differences existing between the *FS-FAP* and the *MS-FAP*.

In the following description we will refer to a clique problem $C = \{V_C, E_C\}$.

Formulation TSP_{LR} is modified into TSP_{LR}^{2P}, which uses the variables defined in Section 4.2.1.2 for formulation TSP_{LR} plus the new set e_{vw}, defined as follows:

- e_{vw}: continuous variable relaxed from a {0,1} variable. It is defined for every $\{v, w\} \in E$. A value of 1 means that the constraint between vertex v and vertex w $(v < w)$ is violated.

$$(TSP_{LR}^{2P}) \quad \text{Min} \sum_{\{v,w\}\in E_C} p_{vw} e_{vw} \tag{5.1}$$

$$\text{s.t. } u_{vw}^V + u_{vw}^N \leq 1 \qquad \forall \{v,w\} \in E_C \tag{5.2}$$

$$u_{Dv} + \left(\begin{array}{l} \displaystyle\sum_{\substack{w\in V_C;\\ v<w}} (u_{vw}^V + u_{vw}^N) + \\ \displaystyle + \sum_{\substack{w\in V_C;\\ w<v}} (u_{wv}^V + u_{wv}^N) \end{array} \right) = 2 \quad \forall v \in V_C \tag{5.3}$$

$$\sum_{v\in V_C} u_{Dv} = 2 \tag{5.4}$$

$$\sum_{\{v,w\}\in E_C} (d_{vw}+1) u_{vw}^N \leq |F| - 1 \tag{5.5}$$

$$e_{vw} - u_{vw}^V \geq 0 \qquad \forall \{v,w\} \in E_C \tag{5.6}$$

$$e_{vz} + e_{wz} - e_{vw} \leq 1 \qquad \begin{array}{l} \forall v,w,z \in V_C;\ v<w<z; \\ d_{vw} \geq d_{vz} + d_{wz} \end{array} \tag{5.7}$$

$$e_{vw} + e_{wz} - e_{vz} \leq 1 \qquad \begin{array}{l} \forall v,w,z \in V_C;\ v<w<z; \\ d_{vz} \geq d_{vw} + d_{wz} \end{array} \tag{5.8}$$

$$e_{vw} + e_{vz} - e_{wz} \leq 1 \qquad \begin{array}{l} \forall v,w,z \in V_C;\ v<w<z; \\ d_{wz} \geq d_{vw} + d_{vz} \end{array} \tag{5.9}$$

$$0 \leq u_{Dv} \leq 1 \qquad \forall v \in V_C \tag{5.10}$$

$$0 \leq u_{vw}^V \leq 1 \qquad \forall \{v,w\} \in E_C \tag{5.11}$$

$$0 \leq u_{vw}^N \leq 1 \qquad \forall \{v,w\} \in E_C \tag{5.12}$$

$$0 \leq e_{vw} \leq 1 \qquad \forall \{v,w\} \in E_C \tag{5.13}$$

Formulation TSP_{LR}^{2P} differs from formulation TSP_{LR} mainly for the following three reasons (apart from the new set of variables):

- each e_{vw} has replaced the respective u_{vw}^V in the objective function;

- there is a new set of inequalities, (5.6), which specifies that each e_{vw}

must be greater than or equal to the respective u^V_{vw};

- there are three new sets of inequalities, (5.7), (5.8) and (5.9), which are the *2-path* inequalities. These inequalities cover all the subproblems with three vertices where if two of its edges are violated then the third one must also be violated. They force the violation of the third edge to be counted when the other two are.

It is interesting to notice that a problem of the same nature as that treated by *2-path* inequalities (i.e. a violation, which is implicitly present in the assignment derived from the solution of TSP_{LR}, is ignored by the bound), can arise on paths of length three of more. For this reason in Allen et al. [8] the equivalent of our *2-path* inequalities is generalised, from a theoretical point of view, for longer paths. The generalisation was not implemented in [8] for computational reasons. For analogous reasons we limit our study to *2-path* inequalities. However, violated paths longer than three would exist only when a great number of constraints are violated and, as pointed out in Section 4.2.1.4, this is quite uncommon in a *FS-FAP*. Thus we do not expect that inequalities for longer paths would be very helpful.

The strategy we adopt to derive *2-path* inequalities is described in the following section.

5.1.1.1 Selection of the inequalities

2-path inequalities are selected by running a complete search, possibly truncated, which examines sets of three variables and tests whether they could generate any inequalities. The algorithm stops when all of the possible sets have been examined or when MI_{tsp} inequalities have already been added to the original formulation. MI_{tsp} is a user defined parameter.

Before starting the search, $\forall v \in V_C$ we calculate the sum of the penalty of the constraints in which v is involved. Formally we define:

$$r(v) = \sum_{\substack{w \in V_C; \\ v<w}} p_{vw} + \sum_{\substack{w \in V_C; \\ v>w}} p_{wv} \tag{5.14}$$

The vertices are examined by the algorithm described above in non-increasing order of r's. Examining the transmitters in this order should help the algorithm to add the most significant inequalities if the search is truncated.

Pseudocode of the algorithm is presented in Figure 5.1.

5.1.2 *Subtour elimination* inequalities

In the *TSP* lower bound for the *MS-FAP* (and in the classical formulation of the *Travelling Salesman Problem*, as well) there is a set of inequalities called *subtour elimination inequalities* (see, for example, Allen et al. [8]). These constraints force the active variables to form a single circuit (*Hamiltonian circuit*) instead of a set of disjoint circuits. In this section we propose a transposition of the *subtour elimination* inequalities for the *FS-FAP* version of the *TSP* bound. In [8] it is stated that *subtour elimination* inequalities have little effects on the tightness of the lower bound for the *MS-FAP*. We hope to obtain a different result for the *FS-FAP*.

Referring to a clique problem $C = \{V_C, E_C\}$ and to the formulation TSP_{LR} (see Section 4.2.1.2), the *subtour elimination* constraints can be writ-

```
*Add2Path(C)*

INPUT:
$C$ = clique problem of type $FS\text{-}FAP$.

$LP$ := formulation $TSP_{LR}^{2P}$ (without (5.7)'s, (5.8)'s and (5.9)'s) representing $C$;
$na$ := 0;
$o$ := vector of the transmitters of $V_C$ ordered with
                    $r$ (eqn. (5.14)) non-increasing;
For $i$:= 0 to $|V_C| - 1$
    For $j$:= $i + 1$ to $|V_C| - 1$
        For $k$:= $j + 1$ to $|V_C| - 1$
            $v$ := smallest index among $o[i]$, $o[j]$ and $o[k]$;
            $w$ := intermediate index among $o[i]$, $o[j]$ and $o[k]$;
            $z$ := greatest index among $o[i]$, $o[j]$ and $o[k]$;
            If($d_{vw} \geq d_{vz} + d_{wz}$)
                add the respective constraint (5.7) to $LP$;
                $na$ := $na$ + 1;
            EndIf
            If($d_{vz} \geq d_{vw} + d_{wz}$)
                add the respective constraint (5.8) to $LP$;
                $na$ := $na$ + 1;
            EndIf
            If($d_{wz} \geq d_{vw} + d_{vz}$)
                add the respective constraint (5.9) to $LP$;
                $na$ := $na$ + 1;
            EndIf
            If($na > MI_{tsp}$)
                stop;
            EndIf
        EndFor
    EndFor
EndFor
```

Figure 5.1: Selection of *2-path* inequalities.

ten as follows:

$$\left(\begin{array}{l} \displaystyle\sum_{\substack{v \in S\setminus\{D\};\\ w \in V_C\setminus S;\\ v<w}} \left(u^V_{vw} + u^N_{vw}\right) + \\ \displaystyle + \sum_{\substack{v \in S\setminus\{D\};\\ w \in V_C\setminus S;\\ w<v}} \left(u^V_{wv} + u^N_{wv}\right) \end{array} \right) + \sum_{w \in V_C \setminus S} u_{Dw} \geq 2 \quad \begin{array}{l} \forall S \subset (V_C \cup \{D\}); \\ D \in S \end{array} \tag{5.15}$$

Inequalities (5.15) specify that for each $S \subset (V_C \cup \{D\})$ which includes the dummy vertex D[1], at least two variables involving a vertex in S and a vertex in $V_C\setminus S$ must be active. This forces the active variables to form a Hamiltonian circuit.

It is important to notice that in the description of the *subtour elimination* inequalities we have referred to TSP_{LR}. The inequalities can also be added without any modification to TSP^{2P}_{LR} (the problem arising when *2-path* inequalities are considered).

The strategy we adopt for the selection of the *subtour elimination* inequalities is described in the following section.

5.1.2.1 Selection of the inequalities

It would be extremely impractical to derive and add to TSP_{LR} all of the possible inequalities (5.15), and also if we do this, the reinforced TSP_{LR} formulation would contain too many useless constraints because most of the *subtour elimination* inequalities would be redundant in the optimal solutions. For this reason, instead of adding all of the possible inequalities, we adopt an iterative algorithm to retrieve significant *subtour elimination* inequalities only.

[1] D has been arbitrarily included into S to simplify the notation. This does not limit the power of the inequalities.

The algorithm, given a solution of TSP_{LR} (possibly already reinforced with some *subtour elimination* inequalities), selects at each iteration a new inequality to be added to TSP_{LR} or decides to stop. In particular, at each iteration i the method constructs a complete graph C^i_{MC} such that $V_{C^i_{MC}} = V_C \cup \{D\}$, where a cost c_{vw} is associated with each edge $\{v, w\}$ of the clique. We define the costs as follows:

$$c_{vw} = \begin{cases} \overline{u}^V_{vw} + \overline{u}^N_{vw} & \text{if } v \neq D \\ \overline{u}_{Dw} & \text{otherwise} \end{cases} \tag{5.16}$$

where $\overline{u}^V_{vw}$, $\overline{u}^N_{vw}$ and $\overline{u}_{Dw}$ contain the value of the respective variables in the last available solution of TSP_{LR}.

To describe exactly how the algorithm works at each iteration i, we need the following definitions.

Definition 4. *Given $S \subset V_{C^i_{MC}}$, $S \neq \emptyset$, the* cut *$\phi(S)$ induced by S is defined as the set of edges $\{v, w\} \in E_{C^i_{MC}}$ which have exactly one endpoint in S.*

Definition 5. *Given a cut $\phi(S)$, we define:*

$$\text{CutCost}(\phi(S)) = \sum_{\{v,w\} \in \phi(S)} c_{vw} \tag{5.17}$$

Our strategy is based on solving, at each iteration i, a *Minimum Cut Problem* on the graph C^i_{MC}, constructed as described above. The minimum cut problem is to find a minimum-cost cut $\phi(S)$. To solve this problem we adopt the algorithm described in Goemans [45] (where the reader can find also a more detailed description of the problem).

Given an optimal solution of the minimum cut problem, we then have the following result:

Theorem 5. *If $\phi(S)$ is a minimum cut of the graph C^i_{MC} with $D \in S$ then:*

```
AddSubtourElimination(C)

INPUT:
$C$ = clique problem of type FS-FAP.

$LP := $ representation of $C$ through formulation $TSP_{LR}$;
stop := FALSE;
$i := 0$;
While(stop = FALSE)
    stop := TRUE;
    solve $LP$;
    construct $C^i_{MC}$ from the last solution of $LP$;
    $\phi(S) :=$ minimum cut in $C^i_{MC}$;
    If($CutCost(\phi(S)) < 2$)
        add the constraint (5.15) induced by $S$ to $LP$;
        stop := FALSE;
    EndIf
    $i := i + 1$;
EndWhile
```

Figure 5.2: Selection of *subtour elimination* inequalities.

- *If* CutCost($\phi(S)$) < 2 *then the* subtour elimination inequality *defined by the vertex set S is violated in the last solution of TSP_{LR};*

- *If* CutCost($\phi(S)$) ≥ 2 *then the last solution of TSP_{LR} respects* all *of the possible* subtour elimination inequalities.

Proof. Follows from the construction of the graph C^i_{MC}. □

The iterative algorithm we adopt to select violated *subtour elimination* inequalities is summarised in Figure 5.2.

The following observation is important. We can add all of the significant *subtour elimination* inequalities because there are generally just a few in our benchmarks. For other types of problems it may be necessary to consider criteria to stop the search process before its natural end. Examples of such criteria could be a maximum number of constraints added or a limit on the time spent searching for them. We have not investigated these alternatives.

5.2 Reinforcing $U2W$

In this section we describe a non-trivial method to obtain a lower bound for the penalty paid in a problem when a lower bound for the number of constraint violations present in the same problem is available.

In Section 4.3.1.2.1 we have proposed a technique to convert a lower bound for the number of constraint violations into a lower bound for the penalty paid. It was based on the solution of the linear program $U2W$. In this section we present a set of inequalities which should improve the quality of the conversion by reinforcing $U2W$.

In the remainder of this section we will adopt the notation previously introduced in Section 4.3.1.2.1.

The weakest point of the original linear program $U2W$ is that, given a clique, it ignores the relations among its edges. Practically, it simply selects the τ edges with the smallest penalties. For this reason the solution of $U2W$ often does not correspond to a feasible assignment of the vertices of the clique. An example of this situation is given in Figure 5.3, where a clique C with four vertices ($V_C = \{0, 1, 2, 3\}$) is depicted. We define $\forall \{v, w\} \in E_C \; d_{vw} = 0$, $p_{02} = p_{03} = p_{23} = 1$ (thin edges) and $p_{01} = p_{12} = p_{13} = 3$ (bold edges). Two frequencies are available ($F = \{0, 1\}$). Both the *formula* bound and the unweighted TSP bound return a lower bound of 2 for the number of constraint violations present in the clique. An optimal solution of the linear program $U2W$ (conversion from the number of constraint violations to a

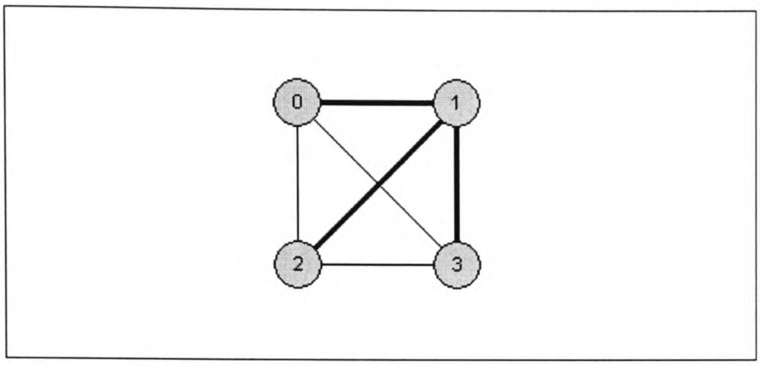

Figure 5.3: Limitation of the conversion from number of constraint violations to penalty based on the linear program $U2W$.

penalty) with $\tau = 2$ is trivially the following one:

$$z_{02} = z_{03} = 1 \qquad z_{01} = z_{12} = z_{13} = z_{23} = 0$$

This solution generates a lower bound of 2 for the penalty paid in the clique. Such a solution does not generate a feasible assignment with the same cost because, as edges $\{0, 2\}$ and $\{0, 3\}$ are active, transmitters 0, 2 and 3 would be assigned to the same frequency. This should cause the edge $\{2, 3\}$ to be violated despite not being counted in the lower bounds.

The conversion can be improved by adding to $U2W$ the analogue of the *2-path* inequalities for the TSP lower bound (see Section 5.1.1). Formally, given a clique-like subproblem $C = \{V_C, E_C\}$, we define the following set of inequalities (in this case we do not need to add new variables to the formulation):

$$z_{vz} + z_{wz} - z_{vw} \leq 1 \quad \forall v, w, z \in V_C;\ v < w < z;\ d_{vw} \geq d_{vz} + d_{wz} \tag{5.18}$$

$$z_{vw} + z_{wz} - z_{vz} \leq 1 \quad \forall v, w, z \in V_C;\ v < w < z;\ d_{vz} \geq d_{vw} + d_{wz} \tag{5.19}$$

$$z_{vw} + z_{vz} - z_{wz} \leq 1 \quad \forall v, w, z \in V_C;\ v < w < z;\ d_{wz} \geq d_{vw} + d_{vz} \tag{5.20}$$

If we consider again the example of Figure 5.3, we can notice how the version of $U2W$ reinforced with inequalities (5.18), (5.19) and (5.20) returns a solution of cost 3 (when $\tau = 2$), and the edge $\{2,3\}$ is now counted as violated by the lower bound.

Inequalities (5.18), (5.19) and (5.20) look the same as the *2-path* inequalities introduced in Section 5.1.1 for the TSP bound. However, the ideas on which the two approaches are based are different. The TSP bound version of the inequalities works *within* the method to improve the estimate, while the version described in this section is to improve the conversion of an *already* calculated lower bound for the number of constraint violations into a lower bound for the penalty paid. In practise this last version is used *after* a lower bound has been calculated, while the TSP version of the inequalities is used *during* the calculation of the lower bound. For this reason, the lower bound obtained by solving $U2W$ reinforced by *2-path* inequalities (5.18), (5.19) and (5.20), where τ is the solution of the unweighted TSP lower bound, is dominated by the lower bound obtained by the weighted TSP bound reinforced with the *2-path* inequalities described in Section 5.1.1 (TSP^{2P}_{LR}). On the other hand, the method described in this section may produce better estimates for the penalty than those based on the TSP bound only, by converting the lower bounds for the number of constraint violations returned by the *formula* bound.

5.2.1 Selection of the inequalities

The strategy we follow to retrieve the inequalities (5.18), (5.19) and (5.20) is the same as that described in Section 5.1.1.1 for the selection of the TSP bound version of the *2-path* inequalities.

The maximum number of constraints added, which was MI_{tsp} in the description of Section 5.1.1.1, is in this case called MI_{u2w}, and it is again a user defined parameter.

5.3 Computational results

In this section we present the results of the methods described in the present chapter. We first define the parameter settings we have adopted, then we analyse how much the local lower bounds result improved when the new techniques are in use, and finally we study how the global lower bounds are affected by these local lower bound improvements.

5.3.1 Parameter settings

The only parameters appearing in the methods described in this chapter are those relative to the maximum number of *2-path* inequalities which are considered. There is a parameter for each of the two versions of these inequalities. MI_{tsp} is the one for the TSP bound version (Section 5.1.1.1), while MI_{u2w} is for the $U2W$ version (Section 5.2.1). Both the parameters have been fixed at 5000. This limit is only reached for a few problems, and it never affects the quality of the estimates.

The parameters of the methods described in Chapter 4 assume the values specified in Section 4.4.2.1.

5.3.2 Local lower bounds

The aim of this section is to understand how much the techniques discussed in this chapter are able to improve local lower bounds. For this analysis we use charts similar to those adopted in Section 4.4.1 to study the original local lower bounds. The logic of the charts in the same, but the sequences of values depicted are different. In this section we have the following values:

- UB: upper bounds of the penalty paid in the subproblems (ignoring each time the rest of the problem). It is calculated by running the tabu search algorithm described in Section 3.3.2 with appropriate parameter settings. Optimality is not guaranteed;

- TSP: lower bounds of the penalty paid in the subproblems obtained by the original TSP bound;

- $TSP + 2P$: lower bounds of the penalty paid in the subproblems obtained by the TSP bound reinforced with *2-path* inequalities (described in Section 5.1.1);

- $TSP + SE$: lower bounds of the penalty paid in the subproblems obtained by the TSP bound reinforced with *subtour elimination* inequalities (described in Section 5.1.2);

- $TSP + 2P + SE$: lower bounds of the penalty paid in the subproblems obtained by applying the TSP bound reinforced with both *2-path* and *subtour elimination* inequalities;

- $U2W$: lower bounds of the penalty paid in the subproblems obtained by solving the linear program $U2W$ (described in Section 4.3.1), with τ as the highest lower bound for the number of constraint violations available (*formula* bound and unweighted TSP bound are considered);

- $U2W + 2P$: lower bounds of the penalty paid in the subproblems obtained by solving the linear program $U2W$ reinforced with *2-path* inequalities (described in Section 5.2), with τ as the highest lower bound for the number of constraint violations available (*formula* bound and unweighted TSP bound are considered).

We aim to study, through three different pairs of charts, how the improvement techniques react when the characteristics of the problems change.

The first two charts are grouped in Figure 5.4. They show how the quality of the estimates changes when the dimension of the frequency domain is changed. We can observe how "$TSP + SE$" never improves the result obtained by "TSP", and this indicates that *subtour elimination inequalities* have no effect when applied to formulation TSP_{LR}. The situation is different for *2-path* inequalities, which are able to improve the results obtained by "TSP" ("$TSP + 2P$"). It is interesting to observe that *subtour elimination* inequalities are sometimes effective (see the first chart) when applied to TSP^{2P}_{LR} ("$TSP + 2P + SE$"). "$U2W + 2P$", which improves substantially the estimate produced by "$U2W$", always obtains at least the same results as "$TSP + 2P$".

Comparing the two charts of the figure, "$U2W + 2P$" gives the most robust method: it is always close to the best estimates in the first graph and it obtains the best results in the second, where the TSP-based methods do not seem to be particularly effective.

It must anyway be observed that the gap between the upper bounds and the lower bounds is not closed.

A second set of charts appears in Figure 5.5. They are presented to study how the quality of the estimates is affected by modifications in the range of the separation values. The observations about the charts of Figure 5.4 are mainly valid also in this case. Figure 5.5 suggests that the improvement techniques work better when all of the d's are 0 (first chart). When the values of the d's are scattered (second chart) the quality of the estimates decreases, both in term of gain on the original methods described in Chapter 4 and in terms of absolute quality. In the second chart the gap between the

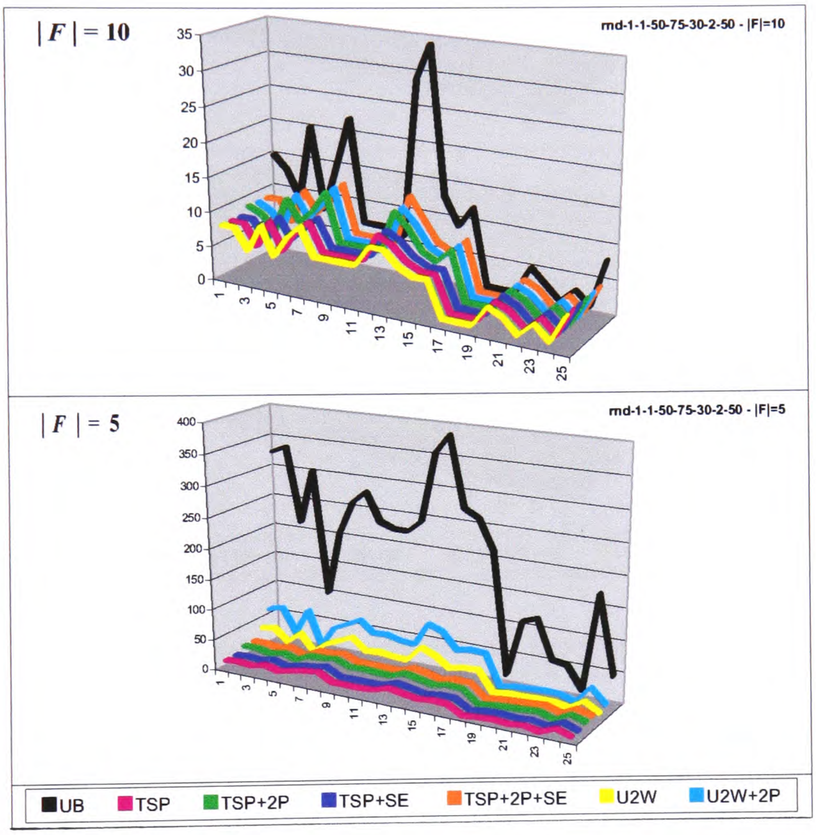

Figure 5.4: Performance of the improved local lower bounds when the frequency domain is changed.

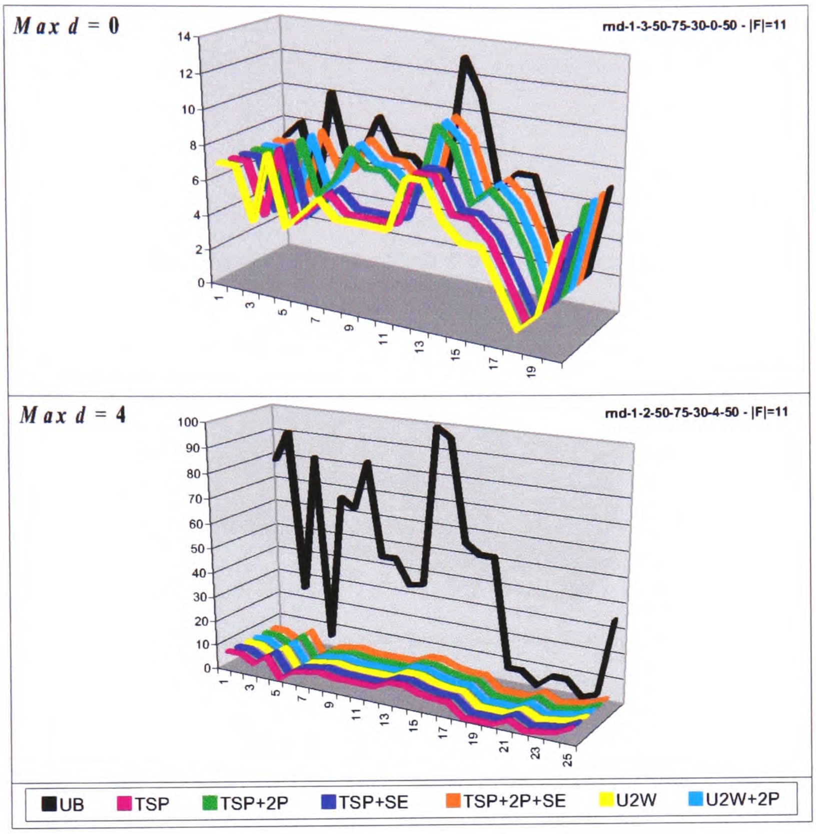

Figure 5.5: Performance of the improved local lower bounds when the range of the separations is changed.

upper and the lower bounds remains wide.

In Figure 5.6 we present the results obtained by the local lower bound techniques on two problems which differ only for the range of the penalty values. Analysing the charts it appears clear that the improvement techniques described in this chapter obtain better results when the penalty values are scattered (second graph). In this case the gap between the upper and the lower bounds is relatively small, notwithstanding that it remains probably too large.

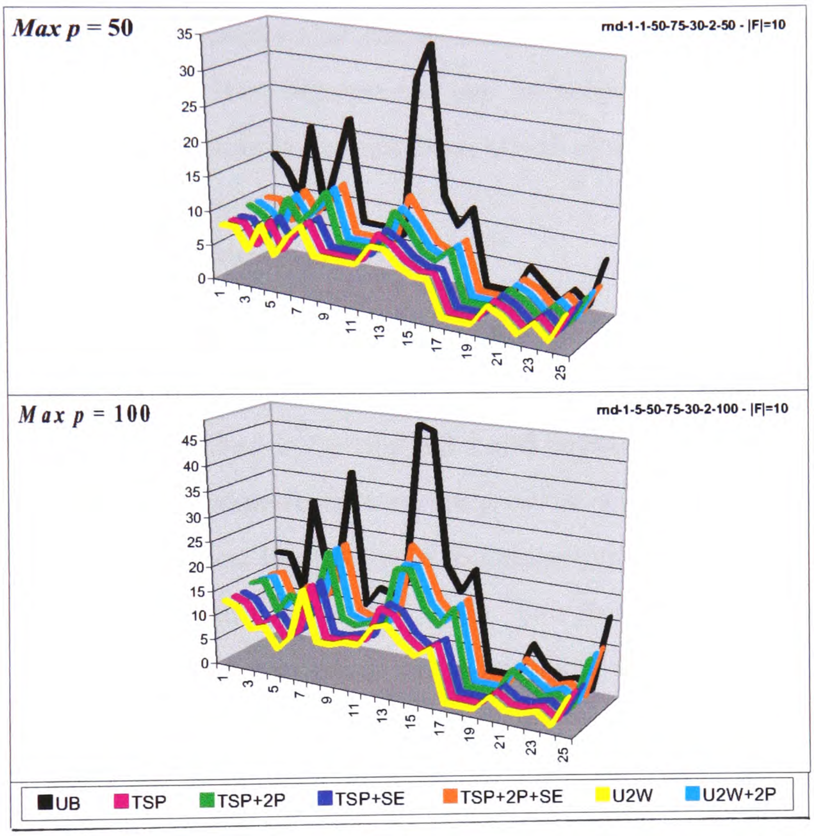

Figure 5.6: Performance of the improved local lower bounds when the range of the penalties is changed.

We can conclude that the best method in terms of robustness is "$U2W + 2P$", whose estimates are always very close to the best available. "$TSP + 2P + SE$" sometimes obtains better results, but it does not work effectively in particular circumstances (see the second chart of Figure 5.4).

We must finally observe that the techniques described in this chapter are not particularly effective, because the gap between the local lower bounds and the local upper bounds has not been closed at a satisfactory level.

5.3.3 Global lower bounds

We have carried out some tests to evaluate the effects of the local lower bound improvement techniques described in this chapter on the global lower bounds $LB2$ and $LB3$, described in Section 4.3.2 and Section 4.3.3 respectively.

The results of the experiments are not positive. $LB2$ is sometimes slightly improved when the local improvement techniques are in use, but no improvement is achieved for our best method, $LB3$. This suggests that when inequalities generated from a larger number of clique-like subproblem are considered ($LB3$ considers non-maximal cliques, which are ignored by $LB2$), the effects of using local lower bounds which are not tight are mitigated. A different justification for the lack of global improvements may be that the techniques described in this chapter are not very effective, and consequently the local estimates are not sufficiently improved by them.

It is interesting to observe that the results of Table 4.7, which suggest that better local lower bounds could sometimes lead to better global lower bounds, are not contradicted. From the second chart of Figure 4.13 and from some tests which are not reported, we know that for the benchmarks *Test282* with $|F| = 71$ and *1-2-50-75-30-4-50* with $|F| = 11$, i.e. the problems of Table 4.7 whose results were improved when local upper bounds were used instead of local lower bounds, the techniques described in this chapter do not produce any significant improvement in the local lower bounds.

Chapter 6

Improving the global lower bounding techniques

In this chapter we aim to improve the global lower bounding techniques $LB2$ and $LB3$, described in Chapter 4.

In the first section we discuss the role of the structure of formulation $FAP3_{LR}$ (presented in Section 4.3.2.1) within the lower bounds and we propose a simplified formulation, $FAP3^S_{LR}$, which achieves the same results, but with shorter computation times and smaller memory requirements.

In the second part of the chapter some reinforcing inequalities for $FAP3^S_{LR}$, which are called *global 2-path* inequalities, are introduced. These inequalities express some structural characteristics of the problems, characteristics which are represented by the integer program $FAP3$, but which are lost when its linear relaxation $FAP3_{LR}$ (or $FAP3^S_{LR}$) is considered.

A section is dedicated to a brief description of some techniques we have developed aiming to overcome the context problem, but which have proved to be completely ineffective.

The last section is dedicated to computational results.

Some of the methods described in this chapter have been presented, together with part of the theory developed in Chapter 4, in Montemanni et al. [80].

6.1 Simplified linear relaxation $FAP3_{LR}^{S}$

A question concerning the results obtained by the global lower bounding techniques $LB2$ and $LB3$ arises. It is about the contribution of formulation $FAP3_{LR}$ to these results, and in particular on the role of the structure of the formulation in the estimates. We suspect that $FAP3_{LR}$ works just as a container for reinforcing inequalities (4.34) and (4.35), which in this case would autonomously supply all of the useful information.

We define a new formulation, that is simply a container for inequalities (4.34) and (4.35), and we study the results obtained when this new formulation is used instead of $FAP3_{LR}$ within the global lower bounds $LB2$ and $LB3$. The formulation, which is called $FAP3_{LR}^{S}$, is a simplification of $FAP3_{LR}$. Variables x^0, x^2 and y have been deleted (together with constraints (4.27), (4.28) and (4.29), that involve these variables). The formulation which results is as follows.

$$(FAP3_{LR}^{S}) \quad \text{Min} \sum_{\{v,w\}\in E} p_{vw}x_{vw}^{1} \tag{6.1}$$

$$\text{s.t.} \quad 0 \leq x_{vw}^{1} \leq 1 \qquad \forall\{v,w\}\in E \tag{6.2}$$

It must be observed that when no constraint (4.34) or (4.35) is added to $FAP3_{LR}^{S}$, the optimal solution trivially has cost 0 for all of the problems, and consequently the formulation makes sense only when some reinforcing inequalities are added to it.

In the remainder of this chapter we will refer to formulation $FAP3^S_{LR}$ reinforced with a set of inequalities $\mathcal{I}$ as $FAP3^S_{LR} + \mathcal{I}$.

There is an important result which connects the solutions of $FAP3_{LR}+\mathcal{I}$ with the solutions of $FAP3^S_{LR} + \mathcal{I}$ when the inequalities of $\mathcal{I}$ involve x^1 variables only. To describe it we refer to Definition 3 and to a *FS-FAP* represented through a graph G. We also assume[1] that $\forall \{v, w\} \in E \ d_{vw} \leq |F| - 2$.

Theorem 6. *If the inequalities contained in $\mathcal{I}$ involve x^1 variables only, then Opt(G, $FAP3_{LR}+\mathcal{I}$) = Opt(G, $FAP3^S_{LR}+\mathcal{I}$).*

Proof. We will prove the inequalities *Opt*(G, $FAP3_{LR} + \mathcal{I}$) $\leq$ *Opt*(G, $FAP3^S_{LR} + \mathcal{I}$) and *Opt*($G$, $FAP3_{LR} + \mathcal{I}$) $\geq$ *Opt*(G, $FAP3^S_{LR} + \mathcal{I}$) separately. To make the exposition clearer we will refer to the x^1 variables of $FAP3^S_{LR}$ as $x^{1(S)}$.

Opt(G, $FAP3_{LR} + \mathcal{I}$) $\leq$ *Opt*(G, $FAP3^S_{LR} + \mathcal{I}$):

Starting from a feasible solution Sol^S of $FAP3^S_{LR} + \mathcal{I}$, we construct a feasible solution Sol of $FAP3_{LR} + \mathcal{I}$. We define the values of the variables of Sol as follows: $\forall \{v, w\} \in E \ x^1_{vw} = x^{1(S)}_{vw}$, $x^0_{vw} = x^2_{vw} = \left(1 - x^{1(S)}_{vw}\right)/2$; $\forall v \in V \ y_v = 0$. Sol is feasible for $FAP3_{LR}+\mathcal{I}$. It also has the same cost as Sol^S. The procedure, when applied to an optimal solution of $FAP3^S_{LR} + \mathcal{I}$, proves the inequality.

Opt(G, $FAP3_{LR} + \mathcal{I}$) $\geq$ *Opt*(G, $FAP3^S_{LR} + \mathcal{I}$):

As $FAP3^S_{LR}$ is a simplification of $FAP3_{LR}$, this inequality is automatically true. □

As inequalities (4.34) and (4.35) involve x^1 variables only, we can conclude that $FAP3^S_{LR}$ can substitute $FAP3_{LR}$ inside the global lower bounds $LB2$

[1] The assumption is not restrictive because if $\exists \{v, w\} \in E$ such that $d_{vw} > |F| - 2$ then the edge $\{v, w\}$ can be eliminated from E, and a fixed cost of p_{vw} can be added instead.

and $LB3$ without any loss of quality in the results. This confirms that most of the information represented in the integer program $FAP3$ is lost when its linear relaxation, $FAP3_{LR}$, is considered.

A direct consequence of Theorem 6 is the result which we present in the remainder of this section. We first need the following definition.

Definition 6. *$I = \{C_I, t_I, \xi_I\}$ is the concise definition for the following reinforcing inequality:*

$$\sum_{\{v,w\}\in E_{C_I}} c^I_{vw} x^1_{vw} \geq \xi_I \tag{6.3}$$

where the elements of the inequality have the following meaning:

- *$C_I = \{V_{C_I}, E_{C_I}\}$: clique-like subproblem on which inequality I is defined;*
- *t_I: type of inequality I. It can be (4.34) or (4.35);*
- *ξ_I: right hand side of inequality I;*
- $c^I_{vw} = \begin{cases} 1 & \textit{if } t_I = (4.34) \\ p_{vw} & \textit{if } t_I = (4.35) \end{cases}$

Proposition 1. *Given a problem G, if $\overline{X} = \{\overline{x}^1_{vw} | \{v,w\} \in E\}$ is a non-zero cost, optimal solution of the linear program $FAP3^S_{LR}$ representing G reinforced with the set of inequalities $\mathcal{I}$, of type (4.34) and (4.35), then $\exists\, I \in \mathcal{I}$ which satisfies the following inequality:*

$$\sum_{\{v,w\}\in E_{C_I}} c^I_{vw} \overline{x}^1_{vw} = \xi_I \tag{6.4}$$

Proof. We suppose that there is an optimal solution $\overline{X}$ such that $\forall I \in \mathcal{I}$ the following inequality is satisfied:

$$\sum_{\{v,w\}\in E_{C_I}} c^I_{vw}\overline{x}^1_{vw} > \xi_I \tag{6.5}$$

We will show that $\overline{X}$ cannot be optimal.

We first need the following definition:

$$\zeta = \min_{I\in\mathcal{I}} \left\{ \frac{\sum_{\{v,w\}\in E_{C_I}} c^I_{vw}\overline{x}^1_{vw}}{\xi_I} \right\} \tag{6.6}$$

Now we consider a new solution $\overline{\overline{X}}$ such that $\forall\{v,w\} \in E$ $\overline{\overline{x}}^1_{vw} = \frac{\overline{x}^1_{vw}}{\zeta}$. $\overline{\overline{X}}$ respects all of the inequalities in $\mathcal{I}$ because $\forall I \in \mathcal{I}$ we have:

$$\sum_{\{v,w\}\in E_{C_I}} c^I_{vw}\overline{\overline{x}}^1_{vw} = \frac{1}{\zeta}\sum_{\{v,w\}\in E_{C_I}} c^I_{vw}\overline{x}^1_{vw}$$

$$\geq \frac{\xi_I}{\sum\limits_{\{v,w\}\in E_{C_I}} c^I_{vw}\overline{x}^1_{vw}} \sum_{\{v,w\}\in E_{C_I}} c^I_{vw}\overline{x}^1_{vw} = \xi_I$$

The cost of $\overline{\overline{X}}$ is less than that of $\overline{X}$ ($\zeta > 1$ because of (6.5)), which consequently was not optimal. □

The proposition above confirms our conjectures about the role of the local lower bounds within the global estimates. It implies that if there is a problem for which every local lower bound for the penalty paid (number of constraint violations present) is much smaller than the penalty paid (number of constraint violations present) within the respective subproblems in an optimal frequency assignment, then the global lower bound cannot be tight.

It is interesting to observe that the gap between each local lower bound and the penalty (number of constraint violations) present in the respective

Formulation	No. of Constraints	No. of Variables
$FAP3_{LR}$	$3\|E\|$	$\|V\| + 3\|E\|$
$FAP3^S_{LR}$	0	$\|E\|$

Table 6.1: Dimensions of formulations $FAP3_{LR}$ and $FAP3^S_{LR}$.

edges in an optimal frequency assignment can be due to a poor performance of the local lower bounding techniques or to the presence of a context problem.

It must also be noticed that the above proposition confirms that the consideration of non-maximal cliques can be useful, because the addition of new constraints can lead to a redistribution of constraint violations among edges, and eventually to a higher-cost solution of $FAP3^S_{LR}$.

6.1.1 Dimensions of $FAP3_{LR}$ and $FAP3^S_{LR}$

In Table 6.1 we compare the dimensions, in terms of number of constraints and number of variables, of formulation $FAP3_{LR}$ and formulation $FAP3^S_{LR}$. The columns of the table have the following meanings:

- Formulation: names of the two formulations we compare;
- No. of Constraints: expressions for the number of constraints of the formulations (number of rows of the problem matrix). Constraints defining variable domains are not counted here;
- No. of Variables: expressions for the number of variables of the formulations (number of columns of the problem matrix).

From Table 6.1 the importance of the simplification appears clear. When the simplified formulation is used there is a much smaller memory requirement because of the reduction in the number of variables and the reduction in the number of constraints. As it is generally easier to solve a smaller problem, the simplification should also guarantee an improvement in the speed of the algorithm.

6.2 *Global 2-path* inequalities

A simple way to reinforce $FAP3^S_{LR}$ is to add to it some inequalities, which are the analogue of the *2-path* inequalities for the linear program $U2W$, described in Section 5.2 (and also of the *2-path* inequalities for the linear program TSP_{LR}, described in Section 5.1.1). Formally the inequalities are as follows:

$$x^1_{vz} + x^1_{wz} - x^1_{vw} \leq 1 \qquad \begin{array}{l} \forall v, w, z \in V : \\ \{v,w\},\{v,z\},\{w,z\} \in E;\ d_{vw} \geq d_{vz} + d_{wz} \end{array} \tag{6.7}$$

$$x^1_{vw} + x^1_{wz} - x^1_{vz} \leq 1 \qquad \begin{array}{l} \forall v, w, z \in V : \\ \{v,w\},\{v,z\},\{w,z\} \in E;\ d_{vz} \geq d_{vw} + d_{wz} \end{array} \tag{6.8}$$

$$x^1_{vw} + x^1_{vz} - x^1_{wz} \leq 1 \qquad \begin{array}{l} \forall v, w, z \in V : \\ \{v,w\},\{v,z\},\{w,z\} \in E;\ d_{wz} \geq d_{vw} + d_{vz} \end{array} \tag{6.9}$$

The presence of these inequalities should help to avoid many situations where the constraint violations (which are forced to exist by inequalities (4.34) and (4.35)) assume infeasible configurations. This should increase the quality of the global lower bounds.

It is interesting to notice that, as inequalities (6.7), (6.8) and (6.9) involve only x^1 variables, Theorem 6 still applies. Also Theorem 1 still applies because if a solution $\overline{X} = \{\overline{x}^1_{vw} | \{v,w\} \in E\}$ respects a set of *global 2-path* inequalities, then $\forall \zeta > 0$ the solution $\overline{\overline{X}}$ (where $\forall \{v,w\} \in E\ \overline{\overline{x}}^1_{vw} = \frac{\overline{x}^1_{vw}}{\zeta}$) respects the same set of *global 2-path* inequalities, and consequently the proof of Theorem 1 will still be valid.

It would be extremely impractical to deal with all of the possible *global 2-path* inequalities of a problem. For this reason we have developed a technique which, at each iteration of the global lower bounding technique $LB3$, adds only those inequalities which are violated in the last solution available. The technique is described in detail in the following section.

6.2.1 Selection of the inequalities

The strategy we adopt to select violated *global 2-path* inequalities is based on a complete search similar to the one presented in Figure 5.1, which was used to retrieve *2-path* inequalities for the linear program TSP_{LR}.

The main difference between the algorithm which is described in Figure 6.1 and the one of Figure 5.1, is that in Figure 6.1 an inequality is added only if it is violated in the last available solution of the reinforced $FAP3^S_{LR}$, while in Figure 5.1 all of the potentially violated inequalities are added at once.

The difference highlighted above is required since the consideration of all of the possible *2-path* inequalities is possible for a subproblem, but not for a complete problem.

The procedure of Figure 6.1 is applied each time a new solution of the reinforced $FAP3^S_{LR}$ is available (at each iteration of $LB3$).

6.3 Methods to overcome the context problem

We have developed some methods aiming to overcome the context problem (see Section 4.4.2.3), but unfortunately they have proved to be completely

```
AddGlobal2Path(Pr, LP)

INPUT:
$Pr$ = problem of type FS-FAP;
$LP$ = (reinforced) representation of $Pr$ through $FAP3^{S}_{LR}$.

For $v$:= 0 to $|V| - 1$
    For $w$:= $v + 1$ to $|V| - 1$
        For $z$:= $w + 1$ to $|V| - 1$
            If(($\{v, w\} \in E$) and ($\{v, z\} \in E$) and ($\{w, z\} \in E$))
                If(($d_{vw} \geq d_{vz} + d_{wz}$) and ($\overline{x}^1_{vw} < \overline{x}^1_{vz} + \overline{x}^1_{wz} - 1$))
                    add the respective constraint (6.7) to $LP$;
                EndIf
                If(($d_{vz} \geq d_{vw} + d_{wz}$) and ($\overline{x}^1_{vz} < \overline{x}^1_{vw} + \overline{x}^1_{wz} - 1$))
                    add the respective constraint (6.8) to $LP$;
                EndIf
                If(($d_{wz} \geq d_{vw} + d_{vz}$) and ($\overline{x}^1_{wz} < \overline{x}^1_{vw} + \overline{x}^1_{vz} - 1$))
                    add the respective constraint (6.9) to $LP$;
                EndIf
            EndIf
        EndFor
    EndFor
EndFor

where $\overline{x}^1_{vw}$ is the value of variable $x^1_{vw}$ in the current solution of $LP$.
```

Figure 6.1: Selection of *global 2-path* inequalities.

ineffective. For this reason we present only a brief description of them.

The idea on which the approaches are based is to insert the calculation of the local lower bounds into the context of the entire problem, and not simply into the context of the clique-like subproblem to which each local lower bound refers.

The main conceptual difference from the global lower bounding technique $LB2$, described in Section 4.3.2, is then that the local lower bounds are not calculated *before* the global lower bound, but they are calculated *together* with the global lower bound. The only way to do this is to embed the methods which produce the local estimates within the global formulation $FAP3_{LR}^{S}$, and consequently the calculation of the local lower bounds within the solving process of the modified $FAP3_{LR}^{S}$. Unfortunately such a strategy implies a new set of constraints (and a new set of variables) to be added to $FAP3_{LR}^{S}$ for each clique-like subproblem considered.

We have developed a version of the TSP local lower bound (see Section 4.2.1) in which the formulation TSP_{LR} is embedded into $FAP3_{LR}^{S}$, and the values of the variables x^{V} of TSP_{LR} are connected to those of the variables x^{1} of $FAP3_{LR}^{S}$.

Another method we have developed is an embedded version of the *formula* local lower bound, in which a new set of constraints and a new set of variables are added to $FAP3_{LR}^{S}$ to connect the value of each local estimate to the number of different frequencies used within the respective clique-like subproblem in the global solution of the modified $FAP3_{LR}^{S}$.

Unfortunately the connections between the (embedded) local lower bounds and the global problem are, in the techniques described above, extremely weak, and for this reason the approaches are ineffective.

6.4 Computational results

In this section we summarise the results obtained by the global lower bounds $LB2$ and $LB3$ when the improving techniques described in this chapter are in use.

6.4.1 Parameter settings

The parameters used within the improved lower bounding techniques $LB2$ and $LB3$ have the same values specified in Section 4.4.2.1 for the original methods.

6.4.2 Lower bound $LB2$

The improvements for the global lower bound $LB2$ are only in the computation times, and they derive from the use of the simplified formulation $FAP3^S_{LR}$ instead of $FAP3_{LR}$. No improvement in the quality of the estimates is expected as we do not apply *global 2-path* inequalities to this bound.

The results achieved by $LB2$ when $FAP3^S_{LR}$ is used are compared, in Table 6.2, Table 6.3 and Table 6.4, with those obtained in Chapter 4. The columns of the tables have the following meaning:

- Problem: names of the problems. Each name is composed of the following two elements:
  - Graph: name of the graph on which the problem is based;
  - $|F|$: number of channels available;
- UB: best upper bounds available. The bounds are obtained using the heuristic algorithms described in Chapter 3;

- $LB2$: lower bounds obtained by $LB2$. The subcolumns have the following meaning:
  - Val: values of the lower bounds;
  - Sec: computation times in seconds. The subcolumns are defined as follows:
    - Old: results achieved in Chapter 4 ($FAP3_{LR}$ is in use);
    - New: results achieved when $FAP3^S_{LR}$ is used instead of $FAP3_{LR}$.

From Table 6.2, Table 6.3 and Table 6.4 it appears quite clear how $LB2$ saves computation time when formulation $FAP3^S_{LR}$ is used instead of $FAP3_{LR}$. The phenomenon is particularly evident for problems with longer computation times.

6.4.3 Lower bound $LB3$

For the global lower bound $LB3$, the improvements derive from the adoption of the simplified formulation $FAP3^S_{LR}$ and from the use of the *global 2-path* inequalities.

The results achieved by the new version of $LB3$ are compared, in Table 6.5, Table 6.6 and Table 6.7, with those obtained in Chapter 4. The columns of the tables have the following meaning:

- Problem: names of the problems. Each name is composed of the following two elements:
  - Graph: name of the graph on which the problem is based;
  - $|F|$: number of channels available;

Problem			$LB2$		
Graph	$\|F\|$	UB	Val	Sec	
				Old	New
AC-45-17	7	32	16	1	1
AC-45-17	9	15	9	1	1
AC-45-25	11	33	21	2	2
AC-45-25	19	8	8	3	1
AC-95-9	6	31	24	1	1
AC-95-9	10	3	3	1	1
AC-95-17	15	33	25	68	26
AC-95-17	21	10	9	22	17
GSM-93	9	32	15	2	2
GSM-93	13	7	4	2	1
GSM-246	21	79	47	28	7
GSM-246	31	25	16	22	7
Test95	31	12	12	2	1
Test95	36	8	7	2	1
Test282	61	51	20	190	117
Test282	71	27	5	124	95
P06-5	11	133	121	7	5
P06-5	41	15	15	7	5
P06-3	31	115	109	41	17
P06-3	71	26	26	31	18
P06b-5	21	52	49	9	5
P06b-5	31	25	25	10	5
P06b-3	31	112	106	59	16
P06b-3	71	26	26	47	17

Table 6.2: Improved lower bound $LB2$ results. Benchmark set 1

Problem			$LB2$		
Graph	$\vert F\vert$	UB	Val	Sec	
				Old	New
GSM2-184	39	5521	4816	88	28
GSM2-184	49	874	874	60	23
GSM2-227	39	10979	7147	179	52
GSM2-227	49	2459	1998	123	46
GSM2-272	39	27416	12792	265	119
GSM2-272	49	7785	5168	270	63

Table 6.3: Improved lower bound $LB2$ results. Benchmark set 2.

Problem			$LB2$		
Graph	$\vert F\vert$	UB	Val	Sec	
				Old	New
1-1-50-75-30-2-50	5	1242	519	4	2
1-1-50-75-30-2-50	10	101	36	3	1
1-1-50-75-30-2-50	11	59	28	2	1
1-1-50-75-30-2-50	15	11	11	3	1
1-2-50-75-30-4-50	11	323	54	3	2
1-3-50-75-30-0-50	11	36	28	2	1
1-4-50-75-30-2-1	10	19	14	1	1
1-5-50-75-30-2-100	10	186	49	3	1
1-6-50-75-30-0-10000	10	6942	3083	3	1

Table 6.4: Improved lower bound $LB2$ results. Benchmark set 3

- UB: best upper bounds available. The bounds are obtained using the heuristic algorithms described in Chapter 3;
- $LB3$: lower bound obtained by $LB3$. The subcolumns have the following meaning:
  - Old: results presented in Chapter 4. The subcolumns are defined as follows:
    - Val: values of the lower bounds;
    - Sec: computation times in seconds;
  - New: results achieved when the improving techniques described in this chapter are in use. The subcolumns have the same meaning as in column "Old".

As expected, from Table 6.5, Table 6.6 and Table 6.7 we can see how the techniques described in this chapter allow better results to be obtained. This is true especially for the weighted problems, for which the results presented in Chapter 4 were poorer. In particular 10 problems of the 30 for which optimality was not reached in Chapter 4, have the estimate improved.

When there is no improvement in the estimate, the computation time of the revised method is practically always shorter. This is not true just for 3 of the 29 problems for which a better estimate has not been found. For these 3 problems the use of *global 2-path* inequalities has produced a longer computation time (notwithstanding the simplified formulation $FAP3^S_{LR}$ was in use).

It is interesting to observe that some experiments we have carried out indicate that when the local improvement techniques described in Chapter 5 are used in conjunction with the global techniques presented in this chapter,

Problem			$LB3$			
Graph	$\lvert F\rvert$	UB	Old		New	
			Val	Sec	Val	Sec
AC-45-17	7	32	20	31	20	27
AC-45-17	9	15	10	4	10	2
AC-45-25	11	33	26	986	26	1114
AC-45-25	19	8	8	3	8	1
AC-95-9	6	31	27	17	27	13
AC-95-9	10	3	3	1	3	1
AC-95-17	15	33	28	738	28	756
AC-95-17	21	10	9	22	9	17
GSM-93	9	32	17	32	17	26
GSM-93	13	7	4	2	4	1
GSM-246	21	79	50	5657	50	5679
GSM-246	31	25	16	22	16	7
Test95	31	12	12	2	12	1
Test95	36	8	7	2	7	1
Test282	61	51	20	190	21	18779
Test282	71	27	6	1030	6	993
P06-5	11	133	121	7	121	5
P06-5	41	15	15	7	15	5
P06-3	31	115	109	41	109	17
P06-3	71	26	26	31	26	18
P06b-5	21	52	49	9	49	5
P06b-5	31	25	25	10	25	5
P06b-3	31	112	106	59	106	16
P06b-3	71	26	26	47	26	17

Table 6.5: Improved lower bound $LB3$ results. Benchmark set 1

Problem			$LB3$			
Graph	$\|F\|$	UB	Old		New	
			Val	Sec	Val	Sec
GSM2-184	39	5521	4856	5035	4856	5032
GSM2-184	49	874	874	60	874	23
GSM2-227	39	10979	7328	16064	7445	70677
GSM2-227	49	2459	1998	123	1998	46
GSM2-272	39	27416	12909	85131	16144	85677
GSM2-272	49	7785	6258	80951	6310	66705

Table 6.6: Improved lower bound $LB3$ results. Benchmark set 2.

Problem			$LB3$			
Graph	$\|F\|$	UB	Old		New	
			Val	Sec	Val	Sec
1-1-50-75-30-2-50	5	1242	802	65296	806	76057
1-1-50-75-30-2-50	10	101	52	1879	53	345
1-1-50-75-30-2-50	11	59	36	616	36	230
1-1-50-75-30-2-50	15	11	11	3	11	1
1-2-50-75-30-4-50	11	323	68	5655	71	18794
1-3-50-75-30-0-50	11	36	35	218	36	31
1-4-50-75-30-2-1	10	19	17	29	17	24
1-5-50-75-30-2-100	10	186	90	2259	94	11138
1-6-50-75-30-0-10000	10	6942	6315	8891	6586	2060

Table 6.7: Improved lower bound $LB3$ results. Benchmark set 3.

no improvement of the results presented in Table 6.5, Table 6.6 and Table 6.7 is achieved.

Chapter 7

Conclusion and future work

7.1 Conclusion

In this thesis we have treated the fixed spectrum frequency assignment problem, modelled through binary constraints.

We have presented an efficient implementation of two well-known meta-heuristic algorithms. In particular we have proposed a version of the tabu search algorithm characterised by a variable length tabu list which seems to be extremely effective when compared with other algorithms presented in the literature.

The main contribution of the thesis is represented by some novel lower bounding techniques we have developed. The most promising of these techniques obtains the estimates (*global lower bounds*) by solving a linear program reinforced with some inequalities, each one derived from the lower bound of the penalty paid (number of constraint violations present) in a clique-like subproblem of the original problem (*local lower bounds*). The method is based on the consideration that it is "easy" to calculate lower bounds for clique-like subproblems, and that these (local) lower bounds can be used to obtain a (global) lower bound for the penalty paid in the whole problem.

An analysis of the results obtained by the global lower bounding techniques has been carried out, and two main factors which may lead to estimates which are not tight (when linear relaxation is used) have been identified. These factors are the quality of the local lower bounds, which sometimes is not high enough, and a structural problem of our methods, which we have called the context problem.

Some techniques aiming to improve the quality of the local lower bounds have been proposed, but they have proved not to be strong enough.

Finally, some simplifications and improvements for the global lower bounding techniques have been proposed. They have often led to better results, both in terms of computation time and quality of the estimates.

In Table 7.1, Table 7.2 and Table 7.3 we summarise the results obtained by our upper and lower bounding techniques on the benchmarks adopted in this thesis. The columns of the tables have the following meanings:

- Problem: names of the problems. Each name is composed of the following two elements:
  - Graph: name of the graph on which the problem is based;
  - $|F|$: number of channels available;
- LB: best lower bounds available. The bounds are obtained using the global lower bound technique $LB3$, described in Chapter 4 and improved in Chapter 6;
- UB: best upper bounds available. The bounds are obtained using the heuristic algorithms described in Chapter 3;
- LB/UB: ratios of the data contained in the previous two columns.

Problem		LB	UB	LB/UB
Graph	$\|F\|$			
AC-45-17	7	20	32	0.625
AC-45-17	9	10	15	0.667
AC-45-25	11	26	33	0.788
AC-45-25	19	8	8	1.000
AC-95-9	6	27	31	0.871
AC-95-9	10	3	3	1.000
AC-95-17	15	28	33	0.848
AC-95-17	21	9	10	0.900
GSM-93	9	17	32	0.531
GSM-93	13	4	7	0.571
GSM-246	21	50	79	0.633
GSM-246	31	16	25	0.640
Test95	31	12	12	1.000
Test95	36	7	8	0.875
Test282	61	21	51	0.412
Test282	71	6	27	0.222
P06-5	11	121	133	0.910
P06-5	41	15	15	1.000
P06-3	31	109	115	0.948
P06-3	71	26	26	1.000
P06b-5	21	49	52	0.942
P06b-5	31	25	25	1.000
P06b-3	31	106	112	0.946
P06b-3	71	26	26	1.000

Table 7.1: Benchmark set 1. Summary of the results.

Problem		LB	UB	LB/UB
Graph	$\|F\|$			
GSM2-184	39	4856	5521	0.880
GSM2-184	49	874	874	1.000
GSM2-227	39	7445	10979	0.678
GSM2-227	49	1998	2459	0.813
GSM2-272	39	16144	27416	0.589
GSM2-272	49	6310	7785	0.811

Table 7.2: Benchmark set 2. Summary of the results

Problem Graph	$\lvert F \rvert$	LB	UB	LB/UB
1-1-50-75-30-2-50	5	806	1242	0.649
1-1-50-75-30-2-50	10	53	101	0.525
1-1-50-75-30-2-50	11	36	59	0.610
1-1-50-75-30-2-50	15	11	11	1.000
1-2-50-75-30-4-50	11	71	323	0.220
1-3-50-75-30-0-50	11	36	36	1.000
1-4-50-75-30-2-1	10	17	19	0.895
1-5-50-75-30-2-100	10	94	186	0.505
1-6-50-75-30-0-10000	10	6586	6942	0.949

Table 7.3: Benchmark set 3. Summary of the results.

The results summarised in Table 7.1, Table 7.2 and Table 7.3 are extremely satisfactory, especially because, as far as we know, there is no other general purpose lower bounding technique available in the literature. All the methods presented so far work by exploiting some specific peculiarity of a given benchmark set, and consequently are effective only for the problems contained in that set.

The largest benchmarks adopted in this thesis are close to the maximum size that can be handled directly by the current implementation of the lower bounding techniques. Larger problems could be handled by an implementation with a more compact memory structure (and possibly by using a special purpose linear program solver instead of CPLEX). This was considered beyond the scope of this thesis.

7.2 Future work

The work presented in this thesis can be extended, in our opinion, mainly in four different directions.

The first direction, which is the most obvious one, concerns the development of effective techniques to improve the local lower bounds and to overcome the context problem.

The second research direction concerns a theoretical study of the graphs representing the problems, aiming to identify those characteristics which help our methods to produce good estimates (this study would be important especially for the lower bounds). This research should lead to a problem classification based on the expectation of the quality of the estimates produced by our methods.

In this thesis we have already identified some of the problem features which seem to help our lower bounding techniques, but a formal classification, with the related graph theoretical study, has not been considered because it was beyond the scope of this work.

The third research direction is about the study of an effective heuristic technique to transform a solution of $FAP3_{LR}^{S}$ (or $FAP3_{LR}$) into a low cost assignment. As anticipated in Section 4.4.2.2, this seems to be a non trivial task.

The fourth research direction concerns the extension of the methods we have proposed to permit them to deal with more complex models of frequency assignment problems. In particular it should be quite easy to modify our methods to make them able to treat problems represented through multi-graphs, i.e. problems in which for a pair of transmitters there are different penalty levels, each one associated with a different separation level (see Eisenblätter [41] for a more detailed description of such a model).

Bibliography

[1] K. Aardal, C.A.J. Hurkens, J.K. Lenstra, and S.R. Tiourine. Algorithms for frequency assignment problems (extended abstract). *CWI Quarterly*, 9:1–8, 1996.

[2] K.I. Aardal, A. Hipolito, C.P.M. van Hoesel, and B. Jansen. A branch-and-cut algorithm for the frequency assignment problem. Technical Report RM 96/011, Maastrict University, 1996.

[3] K.I. Aardal, C.A.J. Hurkens, J.K. Lenstra, and S.R. Tiourine. Algorithms for the radio link frequency assignment problem. Technical Report UU-CS-1999-36, Utrecht University, November 1999.

[4] J. Abril, F. Comellas, A. Cortés, J. Ozón, and M. Vaquer. A multi-agent system for frequency assignment in cellular radio networks. *IEEE Transactions on Vehicular Technology*, 49(5):1558–1565, September 2000.

[5] S.M. Allen, S. Hurley, D.H. Smith, and S.U. Thiel. Using lower bounds in minimum span frequency assignment. In *Meta-Heuristics: Advances and Trends in Local Search Paradigms for Optimization*, pages 191–204. Kluwer Academic Publishers, 1999.

[6] S.M. Allen, S. Hurley, R.K. Taplin, and R.M. Whitaker. Automatic cell planning of broadband fixed wireless networks. 51st IEEE Vehicular Technology Conference, May 2001.

[7] S.M. Allen, D.H. Smith, and S. Hurley. Confidential report. University of Glamorgan, May 1998.

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

[9] S.M. Allen, D.H. Smith, and S. Hurley. Generation of lower bounds for minimum span frequency assignment. *Discrete Applied Mathematics*, to appear.

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

[11] A. Avenali, C. Mannino, and A. Sassano. Minimizing the span of d-walks to compute optimum frequency assignments. *Mathematical Programming*, to appear.

[12] J. Bater. *Non-binary (higher-order) modelling and solution techniques for frequency assignment in mobile communications networks.* PhD thesis, University of London, February 2000.

[13] J. Bater, P. Jeavons, and D. Cohen. Are there optimal reuse distance constraints for FAPs with random Tx placements? Technical Report CSD-TR-98-01, Royal Holloway. University of London, March 1998.

[14] J. Bater, P. Jeavons, and D. Cohen. Non-binary constraints for frequency assignment. In *International Joint Conference on Artificial Intelligence*, 1999.

[15] J. Bater, P. Jeavons, D. Cohen, and N. Dunkin. Are there effective binary frequency separation constraints for frequency assignment coverage problems? In *Proceedings of the NATO RTA SET/ISET Symposium on Frequency Assignment, Sharing and Conservation in Systems (Aerospace),* Aalborg, Denmark, October 1998, NATO RTO-MP13, pages 11.1–11.6, 1999.

[16] R. Battiti, A. Bertossi, and D. Cavallaro. A randomized saturation degree heuristic for channel assignment in cellular radio networks. *IEEE Transactions on Vehicular Technology*, 50(2):364–374, March 2001.

[17] L. Borgne. *Automatic Frequency Assignment for Cellular Telephones Using Local Search Heuristics.* Undergraduate Thesis, Uppsala University, 1994.

[18] R. Borndörfer, A. Einsenblätter, M. Grötschel, and A. Martin. Stable set and other techniques for frequency assignment problems. Preprint, Konrad-Zuse-Zentrum für Informationstechnik Berlin.

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

[20] R. Borndörfer, A. Einsenblätter, M. Grötschel, and A. Martin. The orientation model for frequency assignment problems. Technical Report TR 98-01, Konrad-Zuse-Zentrum für Informationstechnik Berlin, 1998.

[21] A. Bouju, J.F. Boyce, C.H.D. Dimitropoulos, and G. vom Scheidt. Tabu search for the radio links frequency assignment problem. In *Proceedings of ADT'95, Applied Decision Technologies*, pages 233–250, 1995.

[22] A. Bouju, J.F. Boyce, C.H.D. Dimitropoulos, G. vom Scheidt, J.G. Taylor, A. Likas, G. Papageorgious, and A. Stafylopatis. Intelligent search for the radio links frequency assignment problem. In *Proceedings of DSP'95, Digital Signal Processing*, 1995.

[23] J.F. Boyce, C.H.D. Dimitropoulos, G. vom Scheidt, and J.G. Taylor. GENET and tabu search for combinatorial optimization problems. In *WCNN'95, World Congress on Neural Networks*, 1995.

[24] R. Bradbeer. Application of new techniques to military frequency assignment. In *Proceedings of the NATO RTA SET/ISET Symposium on Frequency Assignment, Sharing and Conservation in Systems (Aerospace),* Aalborg, Denmark, October 1998, NATO RTO-MP13, pages 1.1–1.5, 1999.

[25] C. Bron and J. Kerbosch. Algorithm 457: Finding all cliques of an undirected graph [H]. *Commun. ACM*, 16(9):575–577, 1973.

[26] 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(4):1252–1260, July 1999.

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

[28] P. Chardaire and A. Sutter. A decomposition method for quadratic zero-one programming. *Management Science*, 41(4):704–712, April 1995.

[29] D. Costa. On the use of some known methods for T-colorings of graphs. *Annals of Operations Research*, 41:343–358, 1993.

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

[31] M.B. Cozzens and D.-I. Wang. The general channel assignment problem. *Congressus Numerantium*, 41:115–129, 1984.

[32] D. de Werra and Y. Gay. Chromatic scheduling and frequency assignment. *Discrete Applied Mathematics*, 49:165–174, 1994.

[33] R. Diestel. *Graph Theory*. Springer-Verlag, 2000.

[34] R. Dorne and J.-K. Hao. Constraint handling in evolutionary search: A case study of the frequency assignment. *Lecture Notes in Computer Science*, 1141:801–810, 1996.

[35] R. Dorne and J.-K. Hao. Tabu search for graph coloring, T-coloring and set T-coloring. In *Meta-Heuristics: Advances and Trends in Local Search Paradigms for Optimization*, pages 77–92. Kluwer Academic Publishers, 1999.

[36] N. Dunkin and S.M. Allen. Frequency assignment problems: Representations and solutions. Technical Report CSD-TR-97-14, Royal Holloway, University of London, 1997.

[37] N. Dunkin, J. Bater, P. Jaevons, and D. Cohen. Towards high order constraint representations for the frequency assignment problem.

Technical Report CSD-TR-98-05, Royal Holloway, University of London, 1998.

[38] N. Dunkin and P. Jeavons. Expressiveness of binary constraints for the frequency assignment problem. In A. Ferreira and D. Krob, editors, *Proceedings of the IEEE/ACM Workshop, Dial M for Mobility*, 1997.

[39] A. Einsenblätter, M. Grötschel, and A.M.C.A. Koster. Frequency assignment and ramifications of coloring. Technical Report 00-47, Konrad-Zuse-Zentrum für Informationstechnik Berlin, December 2000.

[40] A. Eisenblätter. A frequency assignment problem in cellular phone networks (extended abstract). Technical Report SC 97-27, Konrad-Zuse-Zentrum für Informationstechnik Berlin, July 1997.

[41] A. Eisenblätter, T. Kürner, and R. Fauß. Radio planning algorithms for interference reduction in cellular networks. In *Communications for the millenium, Proceedings of COST252/259 Joint Workshop*, pages 87–92, April 1998.

[42] M. Fischetti, C. Lepschy, G. Minerva, G.R. Jacur, and E. Toto. Frequency assignment in mobile radio system using branch-and-cut techniques. *European Journal of Operational Research*, 123:241–255, 2000.

[43] F. Glover. Heuristics for integer programming using surrogate constraints. *Decision Sci.*, 8:156–166, 1977.

[44] F. Glover, E. Taillard, and D. de Werra. A user's guide to tabu search. *Annals of operations research*, 41:3–28, 1993.

[45] M.X. Goemans. Minimum cuts. In: *Advanced algorithms*, Massachusetts Institute of Technology, December 1994.

[46] M. Grötschel. Frequency assignment in mobile phone systems. *Lecture Notes in Computer Science*, 1974:81–86, 2000.

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

[48] M.P.M. Hall, L.W. Barclay, and M.T. Hewitt, editors. *Propagation of radiowaves.* The Institution of Electrical Engineers, 1996.

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

[50] C. Helmberg. Semidefinite programming for combinatorial optimization. Technical Report 00-34, Konrad-Zuse-Zentrum für Informationstechnik Berlin, October 2000.

[51] C.A.J. Hurkens and S.R. Tiourine. Upper and lower bounding techniques for frequency assignment problems. Technical Report COSOR 95-34, Eindhoven University of Technology, 1995.

[52] S. Hurley, D.H. Smith, and S.U. Thiel. FASoft: A system for discrete channel frequency assignment. *Radio Science*, 32(5):1921–1939, 1997.

[53] S. Hurley, R.M. Whitaker, and D.H. Smith. Channel loading in private mobile radio networks. *Annals of Operations Research*, to appear.

[54] J.C.M. Janssen and T.E. Wentzell. Lower bounds from tile covers for the channel assignment problem. Preprint, Dalhousie University.

[55] B. Jaumard, O. Marcotte, and C. Meyer. Mathematical models and exact methods for channel assignment in cellular networks. In *Telecommunications Network Planning*, pages 239–255, 1999.

[56] P. Jeavons, N. Dunkin, and J. Bater. Why higher order contraints are necessary to model frequency assignment problems. In *Proceedings of ECAI 98. 13th European Conference on Artificial Intelligence.* John Wiley & Sons, 1998.

[57] T.R. Jensen and B. Toft. *Graph Coloring Problems.* Wiley Interscience, 1995.

[58] S. Kirkpatrick, C.D. Gelatt, and M.D. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.

[59] A.W.J. Kolen. A genetic algorithm for frequency assignment. Technical report, Maastricht University, 1999.

[60] A.M.C.A. Koster. *Frequency Assignment. Models and Algorithms.* PhD thesis, Maastricht University, November 1999.

[61] A.M.C.A. Koster, S.P.M. van Hoesel, and A.W.J. Kolen. The partial constraint satisfaction problem: Facets and lifting theorems. *Operations Research Letters*, 23(3-5):89–97, 1998.

[62] A.M.C.A. Koster, S.P.M. van Hoesel, and A.W.J. Kolen. Lower bounds for minimum interference frequency assignment problems. Technical Report RM 99/026, Maastricht University, October 1999.

[63] A.M.C.A. Koster, S.P.M. van Hoesel, and A.W.J. Kolen. Solving frequency assignment problems via tree-decomposition. *Electronic Notes on Discrete Mathematics*, 3, 1999.

[64] R. Kuruppilai, M. Dontamsetti, and F.J. Cosentino. *Tecnologie Wireless.* McGraw-Hill Libri Italia, 1999. In Italian.

[65] T.L. Lau and E.P.K. Tsang. Solving the radio link frequency assignment problem with the guided genetic algorithm. In *Proceedings of the NATO RTA SET/ISET Symposium on Frequency Assignment, Sharing and Conservation in Systems (Aerospace)*, Aalborg, Denmark, October 1998, NATO RTO-MP13, pages 14b.1–14b.27, 1999.

[66] C. Lepschy, R. Menolascino, and G. Minerva. Optimisation problems in mobile networks planning. *Ricerca Operativa*, 28(85/86):95–119, 1999.

[67] D.S.P. Leung. Application of the partial backtracking technique to the frequency assignment problem. In *Proceedings of IEEE Int. Sym. Electromagnetic Compatibility*, pages 70–74, 1981.

[68] J. Loxton. Report on a problem on assignment of radio frequencies. University of New South Wales, September 1985.

[69] V. Maniezzo. Exact and approximate nondeterministic tree-search procedures for the quadratic assignment problem. *INFORMS Journal of Computing*, 11(4):358–369, 1999.

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

[71] V. Maniezzo, A. Carbonaro, and R. Montemanni. An approach to frequency assignment problem based on an ANTS heuristic. In *Proceedings of the 3rd Metaheuristic International Conference (MIC'99)*, pages 311–316, 1999.

[72] V. Maniezzo and R. Montemanni. An exact algorithm for the min-interference frequency assignment problem. Technical Report WP-CO0003, Universitá degli Studi di Bologna, 2000.

[73] A. Manni. *Telefonia Cellulare e Sistema GSM.* Calderini, 1998. In Italian.

[74] C. Mannino and A. Sassano. An enumerative algorithm for the frequency assignment problem. *Discrete Applied Mathematics*, to appear.

[75] R. Montemanni. *Algoritmi Esatti ed Euristici per il Problema dell'Assegnamento di Frequenze.* Tesi di Laurea, Universitá degli Studi di Bologna, March 1999. In Italian.

[76] R. Montemanni. Optimization techniques in radio frequency assignment. Transfer Report, University of Glamorgan, April 2000.

[77] R. Montemanni, J.N.J. Moon, and D.H. Smith. An improved tabu search algorithm for the fixed spectrum frequency assignment problem. University of Glamorgan. In preparation.

[78] R. Montemanni and D.H. Smith. A tabu search algorithm with a dynamic tabu list for the frequency assignment problem. Technical report, University of Glamorgan, November 2001.

[79] R. Montemanni, D.H. Smith, and S.M. Allen. An ANTS algorithm for the minimum span frequency assignment problem with multiple interference. Submitted for publication, April 2001.

[80] R. Montemanni, D.H. Smith, and S.M. Allen. An improved lower bound for the fixed spectrum frequency assignment problem. Submitted for publication, October 2001.

[81] R. Montemanni, D.H. Smith, and S.M. Allen. Lower bounds for fixed spectrum frequency assignment. *Annals of Operations Research*, to appear.

[82] C.A. Morgenstern. *Algorithms for General Graph Coloring.* PhD thesis, University of New Mexico, May 1990.

[83] R.A. Murphey, P.M. Pardalos, and M.G.C. Resende. Frequency assignment problems. In: *Handbook of Combinatorial Optimization* Vol. 4 (ed. D.-Z. Du and P. Pardalos), Kluver Academic Publishers, 1999.

[84] S. Sarkar and K.N. Sivarajan. Hypergraph models for cellular mobile communication systems. *IEEE Transactions on Vehicular Technology,* 47(2):460–471, May 1998.

[85] J. Scourias. Overview of the Global System for Mobile communications. Technical report, University of Waterloo, 1995.

[86] D.H. Smith, S.M. Allen, and S. Hurley. Lower bounds for channel assignment. In: *Methods and Algorithms for Radio Channel Assignment* (ed. R.A. Leese), Oxford University Press, to appear.

[87] D.H. Smith, S.M. Allen, and S. Hurley. Characteristics of good metaheuristic algorithms for the frequency assignment problem. *Annals of Operations Research*, to appear.

[88] D.H. Smith, S.M. Allen, S. Hurley, and W.J. Watkins. Frequency assignment: Methods and algorithms. In *Proceedings of the NATO RTA SET/ISET Symposium on Frequency Assignment, Sharing and Conservation in Systems (Aerospace),* Aalborg, Denmark, October 1998, NATO RTO-MP13, pages K.1–K.18, 1999.

[89] D.H. Smith, S.M. Allen, S. Hurley, and W.J. Watkins. Frequency assignment problems with multiple interference. Submitted for publication, 2000.

[90] D.H. Smith and S. Hurley. Bounds for the frequency assignment problem. *Discrete Mathematics*, 167/168:571–582, 1997.

[91] D.H. Smith, S. Hurley, and S.M. Allen. A new lower bound for the channel assignment problem. *IEEE Transactions on Vehicular Technology*, 49(4):1265–1272, 2000.

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

[93] D.H. Smith, R.K. Taplin, and S. Hurley. Frequency assignment with complex co-site constraints. *IEEE Transactions on Electromagnetic Compatibility*, 43(2):210–218, May 2001.

[94] K.A. Smith, B.K. Kim, and G.F. Sargent. Minimising channel interference in real cellular radio networks. *IEEE Global Communications Conference*, 4:2192–2197, 1998.

[95] D. Tcha, Y. Chung, and T. Choi. A new lower bound for the frequency assignment problem. *IEEE/ACM Transactions on Networking*, 5(1):34–39, February 1997.

[96] S. Tiourine, C. Hurkens, and J.K. Lenstra. An overview of algorithmic approaches to frequency assignment problems. In *CALMA Symposium on Combinatorial ALgorithms for Military Applications*, pages 53–62, 1995.

[97] H.P. van Benthem. GRAPH. Generating Radio link frequency Assignment Problem Heuristically. Master's thesis, Delft University of Technology, May 1995.

[98] C. Voudouris. *Guided Local Search for Combinatorial Optimisation Problems.* PhD thesis, University of Essex, 1997.

[99] C. Voudouris and E. Tsang. Solving the radio link frequency assignment problem using guided local search. In *Proceedings of the NATO RTA SET/ISET Symposium on Frequency Assignment, Sharing and Conservation in Systems (Aerospace),* Aalborg, Denmark, October 1998, NATO RTO-MP13, pages 14a.1–14a.9, 1999.

[100] J.P. Warners. A potential reduction approach to the radio link frequency assignment problem. Master's thesis, Delft University of Technology, April 1995.

[101] J.P. Warners, T. Terlaky, C. Roos, and B. Jansen. A potential reduction approach to the frequency assignment problem. Technical Report 95-98, Technische Universiteit Delft, 1995.

[102] W.J. Watkins, S. Hurley, and D.H. Smith. Evaluation of models for area coverage. Technical Report No. 98003, Cardiff University, December 1998.

[103] R.M. Whitaker, S. Hurley, and D.H. Smith. Frequency assignment heuristics for area coverage problems. Technical report, Cardiff University, September 2000.

[104] H.P. Williams. *Model solving in mathematical programming.* John Wiley & Sons, 1993.

[105] M. Williams. Making the best use of the airways: an important requirement for military communications. *Electronics & Communication Engeneering Journal,* pages 75–83, April 2000.

[106] J. Žerovnik. Experiments with a randomized algorithm for a frequency assignment problem. Technical Report No. 97-27, Ecole Normale Supérieure de Lyon, September 1997.