Université de Toulouse

THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par :
Université Toulouse III Paul Sabatier (UT3 Paul Sabatier)

Discipline ou spécialité :

Informatique et Génie Industriel

Présentée et soutenue par :
Kata Kiatmanaroj

le : mercredi 27 juin 2012

Titre :

Allocation de fréquence dans les systèmes de communication par satellites de type SDMA

Ecole doctorale :
Systèmes (EDSYS)

Unité de recherche :
LAAS-CNRS

Directeur(s) de Thèse :
M. Christian ARTIGUES
M. Laurent HOUSSIN

Rapporteurs :

M. Alexandre CAMINADA
M. Roberto MONTEMANNI

Membre(s) du jury :

M. Jin-Kao HAO
M. Thierry Gayraud

Acknowledgements

This thesis would not have been possible without the guidance and the help of several individuals who in one way or another contributed and extended their valuable assistance in the preparation and completion of my work.

First and foremost I offer my sincerest gratitude to my supervisors, Christian Artigues and Laurent Houssin. They have supported me throughout my study with their patience and knowledge. They allow me to work in my own way and guided my through all difficulties, professionally and personally, I experienced.

I sincerely thank Prof. Alexandre Caminada and Prof. Roberto Montemanni for their examination of this thesis. I am honored by their close insight, their constructive and detailed feedback and their interest in my work. I am also grateful to Prof. Jin-Kao Hao and Prof. Thierry Gayraud for accepting to be members of the jury.

I am grateful to Erwan Corbel from Thales Alenia Space, our industrial partner, who has provided me technical requirements from the industry which is invaluable for this research work.

I would like to acknowledge the French government and GISTDA (Geo-Informatics and Space Technology Development Agency) of Thailand in the award of a Doctoral Research Studentship that provided the necessary financial support for this research and my living expenses.

Many thanks to all members of MOGISA research group who always make my everyday work at the laboratory a pleasure.

Thank Chirstèle Mouclier for all administrative support since the very beginning of my study.

Thanks Romaric Guileerm, Panwadee Tangpattanakul, Dariga and Gaëtan Toulon, Pranee and Pierre Rossi and all of my friends for their help and support.

Last but not the least, I would like to thank my wife and my family for their personal support and great patience at all times.

Abstract

In this thesis, we consider frequency assignment problems arising from an SDMA satellite communication system which consists of a satellite and a number of users distributed inside a fixed sized service area. The objective is to assign a given number of frequency carriers to as many users as possible. This assignment should not violate the incurred interference constraints. Two types of interference are considered i.e. binary and cumulative interference. For each of them, single carrier and multiple carrier frequency assignment models are taken into account. We also propose an Integer Linear Programming (ILP) formulation to deal with 2-dimensional frequency $\times$ time assignments which is more complicated and harder to solve. Single carrier FAPs are solved by greedy algorithms and ILP. A Beam Moving algorithm is devised to further improve the solutions by solving a non-linear optimization problem. Multiple carrier FAPs are modelled as scheduling problem and ILPs. We show that the scheduling model solved through constraint programming methods offers superior performance than the proposed ILP. It is worth noting that, by transforming the cumulative interference into binary interference, scheduling method together with clique-induced constraints yields much better results.

A frequency assignment problem that incorporates the specifications and constraints provided by the industry is also considered. These requirements render the resource allocation problem highly complex. This complexity and the fact that frequency assignment plans must be recomputed frequently in order to cope for user mobility yield classic optimization tool such as ILP impractical. According to this, two greedy algorithms are devised and tested.

Résumé

Le travail présenté dans cette thèse traite des problèmes d'affectation de fréquences (FAP) qui se produisent dans les systèmes de communication par satellite utilisant la technologie SDMA. Ces systèmes se composent d'un satellite et d'une zone de service de taille fixe dans laquelle sont répartis des utilisateurs. L'objectif est alors de servir un maximum d'utilisateur en fréquence dans cette zone de service. Cependant, l'affectation ne doit pas violer les contraintes d'interférence qui apparaissent lorsque deux utilisateurs utilisent une même fréquence ou lorsqu'ils se partagent une même plage de fréquence. Deux types d'interférences sont considérés dans cette étude : les interférences binaire et cumulative. Pour chacune d'elles, les problèmes d'affectation de fréquence de type mono-porteuse (une fréquence par utilisateur) et multi-porteuses (plusieurs fréquences par utilisateur) sont traités. Le problème de l'affectation bidimensionnelle est aussi abordé et nous proposons des modèles de Programmation Linéaire en Nombre Entiers (PLNE) pour le résoudre. Au niveau des méthodes de résolution, nous utilisons des algorithmes gloutons, des modèles de PLNE pour le problème de type mono-porteuse. En outre, un algorithme de déplacement continu de faisceau est conçu pour améliorer les solutions en résolvant un problème d'optimisation continu non linéaire. Concernant le problème de type multi-porteuses, nous le ramenons à un problème d'ordonnancement et celui-ci est résolu à l'aide de la PLNE et la Programmation Par Contraintes (PPC). Il est par ailleurs montré que les résultats issus de la PPC sont meilleurs que ceux de la PLNE. De plus, en transformant les interférences cumulatives en interférences binaires, la méthode d'ordonnancement avec les contraintes induites par les cliques donne de bien meilleurs résultats.

Nous considérons également un problème industriel dans lequel de nombreuses contraintes apparaissent ce qui rend le problème très complexe et insoluble avec des méthodes exactes. Face à ce constat, deux algorithmes gloutons sont réalisés et leurs résultats sont comparés.

Contents

Part I

Introduction

Introduction

With its ubiquitous availability, versatility, and high reliability, satellite communication has revolutionized the world we live in. Fixed and mobile telephone services, television and radio broadcast, internet access, and a large number of applications have changed the way people all over the globe interact. Nonetheless, these wide ranges of applications call for continuing increase in traffic demands and requirements. To cope for these needs, satellite communication technology must be continuously evolved towards greater capacity, higher flexibility, and better services.

Spatial Division Multiple Access (SDMA) appears to be a viable alternative to help achieve these requirements simultaneously [107]. SDMA technology employs antenna arrays and multi-dimensional non-linear signal processing techniques to provide significant increases in capacity and quality of the wireless communication systems [139]. This technology is not restricted to any particular modulation format or air-interface protocol, and is compatible with all currently deployed air-interfaces [138].

SDMA can be applied to a satellite with multi-spot-beam antennas [60] in order to direct communication signals to numerous zones on the Earth's surface. The antennas are highly directional allowing the same frequency carrier to be reused in other surface zones where the frequency separation is sufficiently large. This sufficiently large separation refers to spatial separation in that frequency interference between the reused frequency carriers, if exists, does not harm the communications.

The frequency reuse technique is commonly implemented to yield higher system capacity. To offer services to a large number of users, the system can both utilize several frequency carriers, or frequencies in short, and reuse them as many times as possible. In order to perform frequency reuse efficiently without creating excessive interference, the frequency carriers should be carefully selected. Since frequency is also a limited resource, an efficient method is needed. The method to determine how the frequency carrier is selected and assigned concerns a mathematical discipline called combinatorial optimization which is a subset of mathematical optimization relating to operations research, algorithm theory, and computational complexity theory. Nonetheless, this class of problem itself is known in the literature as Frequency Assignment Problem (FAP) [68], [106].

In this thesis, we consider frequency assignment problems arise from an SDMA satellite communication system which consists of a satellite and a number of users distributed inside a fixed sized service area. The objective is to assign a given number of frequency carriers to as many users as possible. This assignment should not violate the incurred interference constraints.

The thesis is organized as follows: Chapter 1 provides literature review on the FAP. In this chapter, work on the frequency assignment is summarized and classified. The problem classification is based mainly on the optimization objective. Methods and techniques for solving the FAP are also covered. Frequency assignment for satellite communications is also discussed.

Chapter 2 presents our satellite communication problem which is unique in that it considers simultaneously several complex characteristics, namely cumulative interference constraints, multiple carrier models, 2-dimensional time and frequency allocation and interference modulation possibilities linked to the SDMA technology. Interference is considered either binary or cumulative. We also propose problem classification ranging from "easier" problems to "harder" ones.

Based on the problem classification, we firstly deal with problems considering single carrier frequency assignments and interference modification through beam moving in Chapter 3. We then move to harder problems considering multiple carrier frequency assignments (but no beam moving is allowed) in Chapter 4.

Chapter 5 provides treatment to a real world problem based on requirements and constraints provided by the industry. Conclusion and perspective are provided in the last chapter.

Part II

Thesis body

The frequency assignment problem: literature review

1

1 Introduction

In wireless communication, radio frequency is crucial since it is the means for transferring information between two or more connections. Radio frequency usage varies from one application to another. For each wireless application, a continuous block of radio frequency, called a frequency spectrum or a frequency band is required. It is the International Telecommunication Union (ITU) that coordinates the shared global use of the available frequency spectrum.

For a frequency division duplex (FDD) system, two-way communication is achieved by utilizing frequency spectrum in pair, one for transmitter and another for receiver. This pair of frequency spectrum is denoted by $[f_{\min}, f_{\max}]$ and $[f_{\min}+D, f_{\max}+D]$ where D is the duplex distance. This duplex distance is defined to be large enough so that there is no interference between the two ranges. For a time division duplex (TDD) system, two-way communication is achieved by utilizing the same frequency within the frequency spectrum $[f_{\min}, f_{\max}]$ but not at the same time. In both FDD and TDD systems, the frequency spectrum is usually partitioned into a set of frequency channels N, all with the same frequency bandwidth Δ. These frequency channels are indexed consecutively as $F = \{1, \ldots, N\}$ where $N = (f_{\max} - f_{\min})/\Delta)$. In FDD, this index represents a pair of frequency carriers, one on the lower frequency band, another on the upper frequency band.

The available frequency channels N are assigned among the users in the communication system. To be able to support a large number of users, these frequency channels should be reused; and by reusing them, signal interference may occur. Interference of signals is determined by the signal-to-noise ratio at the receiving end of a connection. There, the signal of the transmitting

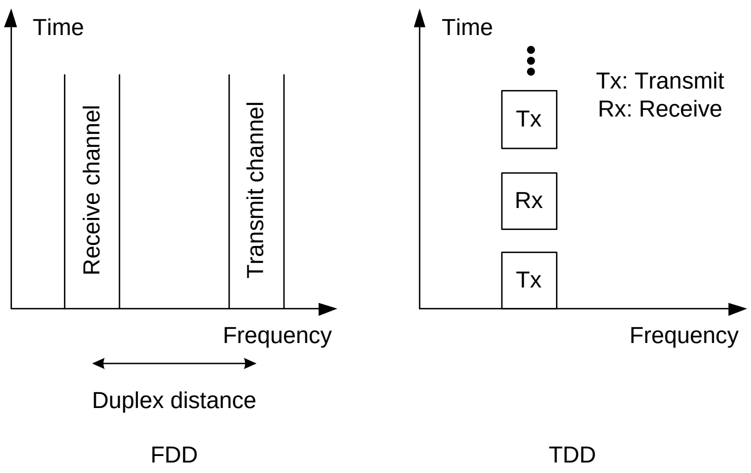


Figure 1.1: FDD and TDD channels.

end should be clearly distinguishable from noise which comes from other signals transmitted at interfering frequencies and from the nature as background noise. The received signal power depends on the transmitted power, the distance between the transmitter and the receiver, the direction it is transmitted to, the direction it is received and the type and shape of environment it traverses. It is impossible to precise these factors in the calculation and get accurate prediction of the signal-to-noise ratio at the receiver. Instead, we simplify the environment variations by treating them as constants and the signal-to-noise ratio now depends largely on the frequency choice. The method to determine how the frequency carrier is selected and assigned in such a way that interference is avoided or at least minimized concerns a mathematical discipline called combinatorial optimization. Combinatorial optimization is a subset of mathematical optimization relating to operations research, algorithm theory, and computational complexity theory. Nonetheless, this class of problem itself is known in the literature as Frequency Assignment Problem (FAP) [68], [106].

According to the latest survey made by Aardal et al. [5], there is no "the" frequency assignment problem. Instead, there are several assignment problems vary from one application to another with various models, objectives and solving methods.

FAP first appeared in the work by Metzger [120]. He is the first that proposed mathematical optimization (a graph coloring technique) as a solution method to a problem on assigning frequencies to mobile phone networks. The assignment concerned both reducing the interference and minimizing the number of frequencies used according to the need at that time when each single frequency was charged for usage. Metzger recognized that the classical vertex-coloring problem in graph theory is analogous to frequency assignment problems where only co-channel constraints are involved. In fact, frequency assignment problems have been associated with graph coloring and its generalization ever since [68], [50]. It is also well known that the graph coloring are NP[1]-hard problem and consequently, the FAP. The time needed to solve this type of problems grows exponentially with the size of the problem. Therefore, it is very unlikely to find any efficient algorithm. We refer to [59] for more information on the NP-hardness and computational complexity.

There are three types of frequency assignment problems. Most of the problems consider static models where the channel assignment remains fixed over time. This is called Fixed Channel Assignment (FCA). Contrarily, if the assignment is based on the demands that change over time, it is then called Dynamic Channel Assignment (DCA). Mix between FCA and DCA creates another type called Hybrid Channel Assignment (HCA). Survey and detailed discussion on these assignment schemes in terms of complexity and performances can be found in Katzela and Naghshineh [88]. Other approaches for the DCA problem have been recently proposed [114], [112], [48] and [108]. We mostly consider FCA in this thesis.

2 Interference

Interference is handled differently from one application to another. Typical interference is called co-channel interference which may occur when same frequency is used (at the same time). If interference occurs when frequency channels differ by one *e.g.* $|f_v - f_w| = 1$ while v and w correspond to two transmitters, it is called adjacent channel interference. In GSM base station, several transmitters can share the same antenna: channel spacing requirement for the assigned channels could be one or two depending on the hardware types. In WLAN, the recommended channel separation is 5 channels in US and 4 channels in much of the world in order to avoid interference [1]. Generally, channel spacing or separation can be treated as a distance d_{vw} so that interference between two transmitters v and w may occur when $|f_v - f_w| < d_{vw}$ ($|f_v - f_w| = 0$ for co-channel interference).

Interference can be conveniently represented by an interference graph $G = (V, E)$. Each antenna is represented by a vertex $v \in V$. Two vertices v and w for which their corresponding signals may interfere for at least one pair of transmitting frequencies, are connected by an edge $\{v, w\} \in E$.

In fact, multiple signals may disturb the communication at the same time and thus interference can be considered cumulative. In the literature, only

[1]Nondeterministic Polynomial-time

a few approaches explicitly take into account this cumulative interference, see [47], [117], [10], [128], [66] and [65]. According to Aardal et al. [5], cumulative interference is ignored in most models where only interference between pairs of connections or antennae is measured. This is considered as binary interference.

We consider both binary interference and cumulative interference in the thesis.

3 Classification and formulations

To model interference, a penalty p_{vwfg} can be given depending on the interference level for each pair of frequencies $f \in F_v$ and $g \in F_w$ of vertices v and w with the associated allowable frequency sets F_v and F_w. In most models this penalty depends only on v, w, and the frequency distance d_{vw}. FAPs with this structure is called "Distance FAPs".

Distance FAP can be represented by a mathematical programming formulation which consists of a set of variables, constraints, and an objective function. Aardal et al. [5] have summarized the work in the literature and suggested a formulation based on variables representing frequency choice for each vertex.

For every vertex v and available frequency $f \in F_v$ they define:

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

Define a demand D_v as the number of required frequencies that should be assigned to a vertex v. The following is considered as the demand constraints:

$$\sum_{f \in F_v} x_{vf} = D_v \quad \forall v \in V.$$

Aardal et al. [5] have also classified the distance FAPs based on their objective function. Common objective functions are Maximum Service FAP, Minimum Order FAP, Minimum Span FAP and Minimum Interference FAP.

3.1 Maximum Service FAP

If the complete assignment in that all vertices are assigned according to their required number of frequencies cannot be found ($\sum_{f \in F_v} x_{vf} \leq D_v$), one would attempt to consider an objective to assign as many frequencies as possible to the vertices. This is called Maximum Service FAP or Max-FAP.

Let n_v denotes the number of frequencies assigned to vertex $v \in V$ and D_v the frequency demand, the maximum service FAP formulation can be given as

$$\begin{aligned}
\max \; & \sum_{v \in V} n_v && && (1.1)\\
\text{s.t.} \; & \sum_{f \in F_v} x_{vf} = n_v && \forall v \in V, && (1.2)\\
& n_v \le D_v && \forall v \in V, && (1.3)\\
& x_{vf} + x_{wg} \le 1 && \forall v, w \in E, f \in F_v, g \in F_w \mid p_{vwfg} > p_{\mathtt{max}}, && (1.4)\\
& x_{vf} \in \{0,1\} && \forall v \in V, f \in F_v, && (1.5)\\
& n_v \in Z_+ && \forall v \in V. && (1.6)
\end{aligned}$$

Equation (1.1) provides the objective which maximizes number of assigned frequencies governed by constraints (1.2) and (1.3). Equation (1.4) provides interference constraints to pair of frequencies if the penalty incurred by choosing frequencies f and g for vertices v and w exceeds the threshold $p_{\mathtt{max}}$. In case that, the frequency assignment is forbidden if the signal-to-noise ratio (SNR) is less than a predefined threshold, this $p_{vwfg} > p_{\mathtt{max}}$ condition can be replaced with $\text{SNR}_{vwfg} < \text{SNR}_{\mathtt{threshold}}$.

While most of the frequency assignment considers assigning a frequency at a time to the vertices, Jaumard et al. [82, 84] consider assigning a block of contiguous frequencies. Later, Fischetti et al. [55] and Jaumard et al. [83] consider minimizing the total number of unsatisfied frequency demands which is in fact a Max-FAP.

Another approach is taken by Mathar and Mattfeldt [118] and Chang and Kim [29] for which frequency assignment in cellular radio networks is considered. Instead of maximizing the number of assigned frequencies, they choose to minimize the the average blocking probability of the whole network. They use Erlang-B formula to calculate blocking probability. Note that the Erlang-B is widely used in telecommunication applications in order to dimension the network according to the given traffic demand. Minimizing the average blocking probability can also be found in [75], [164], [162] and [94].

3.2 Minimum Order FAP

When each single frequency is charged for usage, network operators need to minimize the number of different frequencies in order to minimize their frequency cost. The model to minimize the number of different frequencies is called the Minimum Order FAP (MO-FAP). In order to model this, another binary variable is introduced for each frequency as

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

The MO-FAP model can be formulated as

$$\begin{aligned}
\min \sum_{f \in F} y_f & & (1.7)\\
\text{s.t. } x_{vf} \leq y_f & \quad \forall v \in V, f \in F_v, & (1.8)\\
\sum_{f \in F_v} x_{vf} = D_v & \quad \forall v \in V, & (1.9)\\
x_{vf} + x_{wg} \leq 1 & \quad \forall v, w \in E, f \in F_v, g \in F_w \mid p_{vwfg} > p_{\mathtt{max}}, & (1.4)\\
x_{vf} \in \{0, 1\} & \quad \forall v \in V, f \in F_v, & (1.5)\\
y_f \in \{0, 1\} & \quad \forall f \in F. & (1.10)
\end{aligned}$$

Equation (1.7) provides the objective which minimizes the number of used frequencies. Constraints (1.8) state that if a frequency is assigned to a vertex, this frequency should be counted as used. Constraints (1.9) concern assigning the number of frequencies according to the demand.

MO-FAP can be found in Aardal et al. [4], Dorne and Hao [46, 70] and Hao et al. [71]. Giortzis and Turner [61] also work on MO-FAP but they call the problem as maximum packing-FCA.

3.3 Minimum Span FAP

If one is bounded to pay the frequency cost by the size of frequency spectrum (a block of contiguous frequencies), it is of interest to find the way to assign frequencies using the smallest frequency spectrum as possible. The size of the frequency spectrum is the difference between the maximum frequency and the minimum frequency and is generally denoted as a frequency span. The problem to assign frequencies and minimize the frequency span is called Minimum Span FAP (MS-FAP). MS-FAP can be modelled by introducing two variables representing both ends of the frequency spectrum.

Define variables $z_{\mathtt{max}}$ and $z_{\mathtt{min}}$ for the largest frequency and the smallest frequency. The MS-FAP can be formulated as

$$\min\ z_{\mathtt{max}} - z_{\mathtt{min}} \tag{1.11}$$

$$\text{s.t.} \sum_{f \in F_v} x_{vf} = D_v \qquad \forall v \in V, \tag{1.9}$$

$$x_{vf} + x_{wg} \leq 1 \qquad \forall vw \in E, f \in F_v, g \in F_w \mid p_{vwfg} > p_{\mathtt{max}}, \tag{1.4}$$

$$z_{\mathtt{max}} \geq f y_f \qquad \forall f \in F, \tag{1.12}$$

$$z_{\mathtt{min}} \leq f y_f + f_{\mathtt{max}}(1 - y_f) \quad \forall f \in F, \tag{1.13}$$

$$x_{vf} \leq y_f \qquad \forall v \in V, f \in F_v, \tag{1.8}$$

$$x_{vf} \in \{0, 1\} \qquad \forall v \in V, f \in F_v, \tag{1.5}$$

$$y_f \in \{0, 1\} \qquad \forall f \in F, \tag{1.10}$$

$$z_{\mathtt{min}}, z_{\mathtt{max}} \in Z_+. \tag{1.14}$$

Equation (1.11) provides the objective which minimizes the frequency span. If a frequency is assigned then $y_f = 1$ and this frequency f should be contained in the spectrum. Constraints (1.12) and constraints (1.13) include f in the spectrum and provide the value of the largest and the smallest frequencies. $f_{\mathtt{max}}$ denotes the maximum frequency and acts as an upper bound for the $z_{\mathtt{min}}$.

Minimizing span problem arises with the high demand on frequency resources to cater for the enormous growth in the mobile telephony. Gamst [58] and Sivarajan [145] were among the first who consider this objective. More works on minimum span problem which also concern frequency assignment in cellular networks are found in the late 90's such as in [132], [160], [81], [146], [148], [149], [115], and [12].

Apart from cellular-related work, Robert [137] proposes T-coloring which is based on [68] in order to minimize either the number of channels or the frequency span. Further results can be found in Raychaudhuri [136].

3.4 Minimum Interference FAP

In the past, operators paid for each frequency usage so most problems at that time stressed more on minimizing number of frequencies or the frequency span. Nowadays, it becomes very common that the operators pay for the license fee for a right to use parts of the frequency spectrum. They provide services based on a fixed number of frequencies. Nonetheless, high demand in communications leads to the needs for a lot more frequencies than the licensed ones. According to this, frequency reuse is inevitable and the more the frequencies are reused, the more the need for the operator to reduce the interference. Minimum interference FAP (MI-FAP) thus becomes the topic of high interest. In fact, contribution to the MI-FAP is much higher than those in other FAPs.

Distance FAP may become infeasible since the interference cannot be avoided by any frequency assignment or frequency plan. Instead of using penalties as part of constraints, here, we deal with it directly by trying to minimize the overall penalties values incurred by the frequency choices as:

$$\min \sum_{vw \in E} \sum_{f \in F_v, g \in F_w} p_{vwfg} x_{vf} x_{wg} \tag{1.15}$$

The terms $x_{vf}x_{wg}$ is not linear; nonetheless, the linearization can be achieved by introducing the variables $z_{vwfg} = x_{vf}x_{wg}$ using the following set of constraints:

$$z_{vwfg} \geq x_{vf} + x_{wg} - 1 \qquad \forall vw \in E, f \in F_v, g \in F_w \tag{1.16}$$

$$z_{vwfg} \leq x_{vf} \qquad \forall vw \in E, f \in F_v, g \in F_w \tag{1.17}$$

$$z_{vwfg} \leq x_{wg} \qquad \forall vw \in E, f \in F_v, g \in F_w \tag{1.18}$$

$$z_{vwfg} \in \{0, 1\} \qquad \forall vw \in E, f \in F_v . g \in F_w \tag{1.19}$$

Hard constraints such as the frequency separation constraints can be achieved by setting high penalty values. The models proposed so far consider binary interference. To deal with cumulative interference, interference from multiple sources should be calculated. Generally, this interference should not be greater than a predefined threshold which is mostly set related to the acceptable signal-to-noise ratio value. Together with the penalty values p_{vwfg} derived in the same manner, cumulative interference constraints for a vertex v if frequency f is chosen can be given as

$$\sum_{w \in \text{Interf}} \sum_{g \in F_w} p_{vwfg} x_{vf} x_{wg} \leq I_{vf} x_{vf} \quad \forall v \in V, f \in F_v. \tag{1.20}$$

"Interf" corresponds to a set of interfering vertices to vertex v and I_{vf} corresponds to the interference threshold of vertex v. Equation (1.20) can be further linearized by defining a large upper bound on interference such as by setting it equal to the sum of interference to the vertex v if all of its neighbors are interfering. The upper bound of this type is usually named a "big" M in the literature. With this M, the linear form of Equation (1.20) can be written as

$$\sum_{w \in \text{Interf}} \sum_{g \in F_w} p_{vwfg} x_{wg} \leq I_{vf} + M(1 - x_{vf}) \quad \forall v \in V, f \in F_v. \tag{1.21}$$

A huge interest in frequency minimization in mobile telephony leads to a lot of research. Various solving methods have been used. MI-FAP with penalty minimization can be found as Constraint Satisfaction Problem (CSP) or Partial CSP in [17] and [99]. Zerovnik [163] considers minimizing number of constraint violations using a randomized graph coloring algorithm. Borndörfer et al. [19] propose an ILP formulation instead of graph coloring as it could not model the interference minimization correctly. Koster et al. [100, 101, 102] propose tree decomposition to solve FAPs as ILP, dynamic programming, and partial constraint satisfaction problems.

Search heuristics are also considered. Verfaillie et al. [157] propose Russian doll search which replaces one search by n successive searches on nested subproblems. Tsang and Voudouris [152] consider guided local search by augmenting the objective function with penalties. Tiourine et al. [150] consider several local search algorithms which are tabu search, simulated annealing and variable-depth search. Bouju et al. [24] compare tabu search, GENET and double-update Boltzman machine. Instead of consider the penalties or constraints, these methods take constraints in form of cost function, fitness function, or energy function to be minimized. Cost function can be found in work on simulated annealing as in [49], [96] and [8]. Fitness function can be found in work on evolutionary search or genetic algorithm as in [86], [105], [80], [36], [37] and [126]. Energy function used in a modified hopfield network is proposed by Kim et al. [93].

3.5 Other variations

A different approach is proposed recently in Boche et al. [16]. They consider a certain measure of fairness in the allocation of resource. This involves mathematical disciplines such as social choice theory, social welfare theory and axiomatic theory.

Segredo et al. [144] consider multi-objective optimization. They use multiobjectivisation technique [1] and propose a Multiobjectivised Memetic Algorithm for solving frequency plans in real-world GSM network instances.

Whitley et al. [161] analyze various forms of the Frequency Assignment Problem using the theory of elementary landscapes. They show that three variants of the Frequency Assignment Problem are either directly an Elementary Landscape, or are a superposition of two Elementary Landscapes. More details on elementary landscape can be found in [147].

[1] Multiobjectivisation is a technique to transforms a mono-objective optimization problem into a multi-objective one with the aim to avoid local optima [97].

4 Solving methods

4.1 Exact methods

Implicit enumeration methods such as tree search and dynamic programming can provide an optimal solution to the problem. The goal is to find a set of value assignments to certain variables that will satisfy specific mathematical equations and inequations and maximize or minimize a certain function. Such exact solving scheme can be based on mathematical programming formulations, mainly integer or mixed-integer programming formulations or on constraint satisfaction problem (CSP) formulations. Generally the solution space grows exponentially with the size of the problem. Techniques such as instance reduction and node pruning e.g. through no-good recording, LP relaxation, lower bounding methods, etc. are applied in order to reduce the search space and the search time.

Koster et al. [103] propose a dynamic programming approach based on the tree-decomposition of the constraint graph. Sanchez et al. [143] combine Russian doll search with tree decomposition and closed a hard FAP instance that remained open for 10 years. More recently, Allouche et al. [9] closed the last open instance from the CELAR instance set [53] through a parallelization of tree decomposition-based dynamic programming. Dib et al. [43, 41, 42] proposed no-good recording techniques based on tabu lists for solving MS-FAP problems through constraint programming. Montemanni et al. [124, 125] proposed efficient clique subproblem-based lower bounds for the sum of weighted constraint violation in fixed spectrum frequency assignment problems.

Bosio and Yuan [20] present optimization approaches for WLAN Access Point location and frequency assignment. They propose a two-step approach to deal with access point location and frequency assignment in order to maximize access efficiency. For each of the two steps they derive integer hyperbolic formulations and their linearizations, and propose an enumerative integer formulation.

Recently, Mann and Szajkó [116] presents a systematic study of the complexity of different FAP ILP models proposed by Aardal et al. [5]. They examine different types of constraints, different problem sizes and constraint densities, and varying sets of available frequencies. They then conduct empirical measurements with an ILP solver to assess how problem complexity depends on these factors.

4.2 Heuristics and metaheuristics

In order to solve the assignment problem or obtain good solutions in a reasonable amount of time, a large number of heuristic techniques have been applied in the literature. These techniques produce approximate solutions and there is no guarantee that an optimal solution is found. For a given suboptimal solution, there is generally no information either how far away it is from the optimal solution. Providing that an exact search for the optimal solution is impractical for a large-scale problem due to its exponentially growing computation time, the heuristic method then become a viable alternative. Different approaches have been proposed for solving frequency assignment problems. Each of them are described in the subsections below.

4.2.1 Greedy algorithm

Greedy algorithm is an algorithm that iteratively makes the choice that looks best at the moment. That is, it makes a locally optimal choice in the hope that this choice will lead to a globally optimal solution [34]. Once a decision has been made, it is never reconsidered. The advantage to using a greedy algorithm is that it is simple and fast. The disadvantage is that it is possible that the local optimal choices made iteratively may lead to a bad globally solution.

In frequency assignment, greedy algorithm can be constructed by iteratively selecting a vertex and assign a feasible frequency to it. The vertex selection and frequency assignment follow a given set of rules. Vertices can be ordered statically in that the order remains unchanged over the iterations. They can also be ordered dynamically in which a next vertex is determined at each iteration. Three vertex ordering are generally considered:

- Highest degree first: the vertices are ordered by non-decreasing degree,
- Smallest degree last: the vertices are iteratively ordered so that at each step, the vertex having the least degree is chosen and removed from the vertex set,
- Random order.

The degree of v can be simply the number of vertices adjacent to v. Sivarajan et al. [145] consider the degree by taking into account the demands and frequency distance requirement.

Costa [35] adopts the well known DSATUR [25] by defining the saturation degree of a vertex v as the number of blocked (non-assignable) frequencies. At each iteration, the greedy algorithm selects the vertex having the largest saturation degree and assigns the smallest feasible frequency to it. Borndörfer et al. [18] propose a slight modification in that the frequency assignment should also minimize the cost increase. Valenzuela et al. [153] define a Generalized Saturation Degree (GSD) for a vertex v as the sum of weights of its blocked frequencies. If the frequency f is blocked for vertex v, the weight of f is given by the largest penalty p_{vwfg} for all adjacent vertices w to v.

Sung and Wong [148] propose sequential packing algorithm by determining a family of stable sets $S_1, \ldots, S_n$ such that each vertex v is contained in exactly m_v stable sets in the family. All the vertices in stable set S_i are then assigned frequency i. Sequential packing is achieved by constructing S_{i+1} after completing S_i.

4.2.2 Local search

Local search (LS) is a neighborhood search method for solving computationally hard optimization problems. According to Papadimitriou [130], local search starts with a given feasible solution and iteratively replaces this solution by selecting a better one from a subset of the solution set. If there is no improvement found, the algorithm stops. The solution subset is depended on the current solution (and is therefore called the neighborhood of the solution) and is defined as a set of solutions that can be obtained by a predefined set of small changes or moves to the current solution. Two moves are commonly found: 1-exchange and 2-exchange moves. The 1-exchange move can be performed by selecting a vertex v and change its frequency. Consider a frequency assignment problem with the number of frequency m, the number of possible moves for the vertex v is $m - 1$. The 2-exchange move can be performed by selecting two vertices and swapping their frequencies. These moves can be found, for example, in Park and Lee [132] and in Mishra et al. [121].

There is a trade-off in the neighborhood definition. The larger the neighborhood, the higher the probability of finding a good solution in it, but the larger the computational requirements to fully explore it. Large Neighborhood Search (LNS) considers such large neighborhoods, with typically an exponential number of neighbor solutions. This neighborhood is however fully explored by an efficient algorithm, either because the problem of finding the best neighbor is polynomial (it may resort to a shortest path problem or to a min-cost flow problem for example), or because a (truncated) implicit enumeration technique is used to explore it. Palpant et al. [129] propose a LNS scheme for the FAP.

Guided local search is proposed by Tsang and Voudouris [152]. It allows the algorithm to escape the local minima. The basic idea is to augment the objective function with penalties whose the sum is to be minimized. When trapped, the penalties that are associated with the local optima are increased. By minimising the penalty sum, this can direct the search away from local optima.

4.2.3 Tabu search

Tabu search is a modern heuristic method introduced by Glover [62] as an efficient way of finding high quality solutions to hard combinatorial optimisation problems. In contrast to the standard local search, it allows non-improving moves. At each iteration the best solution in the neighborhood is selected as the new solution. This new solution can be worse than the previous one. The algorithm maintains a tabu list which is a short-term set of the solutions that have been visited in the recent past. Cycling is avoided by not reverting to the recent solutions in this list. Nonetheless, since the tabu list has a fixed definable length, the solution that was in the tabu list can be chosen again once it is removed from the list. The removal is done according to first-in-first-out basis. The algorithm stops after a fixed number of non-improving moves has reached.

Costa [35] solves the MS-FAP by iteratively reducing the frequency span of the spectrum used for assignment. Tabu is used while trying to recolor the conflicting nodes (nodes which are not sufficiently spaced according to the frequency separation requirement). In order to prevent cycling, a tabu list is defined containing nodes and their associated colors that they changed from.

Borgne [17] proposes a sophisticated move which is adapted from the Kempe Chains Interchange. The idea is first select two adjacent vertices u and v that are assigned the same frequency f_1. Select another frequency f_2 which is not assigned to the vertices adjacent to u. Then assign f_1 to all vertices which were assigned f_2 and assign f_2 to all vertices which were assigned f_1.

Bouju et al. [23] apply arc consistency as pre-processing procedure before performing the tabu search. This arc consistency help reducing the size of the search space by trimming the available values in the domain of each variable. They define a move as a change of a frequency. Instead of checking all possible moves at each iteration, they define the neighbors as all the vertices within $N\%$ of the maximum number of constraint violations such that if $N = 1$, only the vertices with the maximum number of violations are considered. If $N = 0$, all vertices are considered.

Adjakplé and Jaumard [7] considers two types of moves to solve MI-FAP with block frequency assignment. The first move consists of replacing exactly one frequency block assigned to a vertex with other available blocks. The neigborhood is restricted by fixing the maximum number of moves. Blocks with the largest local violation are chosen first. The second move is performed periodically. It considers reassigning all frequencies in a vertex having the largest local violation. The reassignment is done by a greedy algorithm.

Hao and Perrier [72] use tabu search for solving the MS-FAP. They apply the standard MI-FAP as a fitness function for the search algorithm and use 1-exchange neigbborhood for the move. The algorithm is improved further in Hao et al. [71].

A dynamic tabu list is considered by Montemanni et al. [122] for assigning frequencies in cellular network. The length of the list reduces with every iteration. In addition, after a number of tabu search iterations, all of the cells are re-optimized. The optimization is performed at each cell at a time by fixing frequencies assigned to all cells except those for the selected cells. The assignment of this selected cell is optimized by an exact method.

Vasquez et al. [154] propose a Consistent Neighbourhood to work with Tabu search in order to overcome the risk for narrowly missing the best solution. The consistent neighbourhood method is maintained by an effective propagation of the instantiations through the constraints. They use the developed algorithm to solve four real-life problems which are the frequency assignment problems with polarisation, the daily photograph scheduling problem, the Agile Earth Observing Satellite (AEOS) management and the antenna positioning problem.

Elhachimi and Guennoun [51] propose a hybrid Tabu search with an adaptive Constraint Satisfaction technique. The algorithm uses a local search method to obtain an initial solution that respects all constraints with minimum cost. A global solution is constructed by permutation of all frequencies of a constraint link in its frequency domain in order to obtain the smallest maximum frequency used.

Mabed et al. [113] propose a new fixed channel assignment (FCA) model for GSM radio networks which takes into account both spatial and temporal variation of traffic. They propose an original and effective hybrid Genetic-Tabu search algorithm to get high quality frequency plans. The hybrid algorithm combines a problem specific crossover and a Tabu search procedure. Based on the generated and real data, the proposed algorithm gives better frequency plans in terms of the three quality criteria which are minimizing total interference, minimize the worst performance of the frequency plan over the time and minimize the maximal amount of interference on one station.

4.2.4 Simulated annealing

Simulated annealing (SA) is a probabilistic metaheuristic proposed by Kirkpatrick et al. [95] and Černý [156]. It can be used to find the global optimum of a cost function or an objective that may possess several local minima by providing a means to escape from each of them. It allows hill-climbing moves in hope to find a global optimum. If the move is better than its current position then simulated annealing will always accept it. If the move is worse then it will be accepted based on some probability. The probability of accepting a worse move is a function of both the temperature T of the system and of the change in the cost function. The choice is almost random when T is large, but increasingly selects the better solution as T goes to zero (frozen).

According to Aardal et al. [5], in most algorithms, the neighborhood of SA is defined by 1-exchange moves. Typically a vertex is selected at random and move into the least costly alternative frequency. In some cases, the new frequency is assigned at random. The algorithm's temperature decreases only after a specified number of iterations have been performed at a constant temperature.

Duque-Antón et al. [49] use the algorithm for solving MI-FAP. The vertex and its frequency are randomly chosen. Occasionally, the frequency is chosen as the mostly assigned one in the nearby non-interfering vertices in order to improve the algorithm performance. The cooling rate is chosen so that the difference between the average solution cost of n iterations at temperature t_1 and that of n iterations at temperature t_2 is no more than the standard deviation of the solution costs at temperature t_1. The system is frozen when the current solution does not change during the last n iterations.

Knälmann and Quellmalz [96] also apply SA for solving MI-FAP. The vertex and its frequency are randomly chosen and changed.

Mathar and Mattfeldt [118] consider a move defined by permutations of the current solution. A solution is represented by $\pi = \pi_1, \ldots, \pi_m$ where π_i is an ordering of vertices assigned with $i_t h$ frequency. The move is done by randomly selecting a frequency f and a permutation from a set of pre-defined feasible permutations then apply this permutation to π_f.

The temperature T remains constant in the work by Zerovnik [163]. At each iteration a vertex v with a large number of violated constraints is selected. A new frequency f is assigned with probability $e^{-S_f/T}$ where S_f is equal to the number of constraints that would be violated if f is assigned to v.

Al-Khaled [8] also apply an adaptive cooling rate which depends on the difference between the average of the accepted solutions of the iterations at temperature t_1 and those at temperature t_2.

Threshold accepting is proposed and applied by Hellebrandt and Hellter [73]. The initial temperature is chosen in a way that the acceptance rate is between 0.8 and 0.9. They use 1-exchange moves with a neighborhood restricted to those for which the move does not violate the hard constraints. Before the temperature is reduced, they apply the one-cell optimization algorithm in order to optimize over the neighborhood of the current solution.

4.2.5 Genetic algorithm

Genetic algorithm (GA) is an metaheuristic proposed by Holland [74] and is inspired by Darwin's theory of evolution. A population of chromosomes represents a solution set. A gene, which is part of a chromosome, represents a variable. Typically, solutions are represented as binary strings (binary encoding), but other encodings are also possible, e.g. permutation encoding.

Solutions (chromosomes) from one population are taken and used to form new solutions (offspring). These chromosomes are selected from their fitness values which reflect how close they are to achieving the objectives or specification. The fitness value is determined by a fitness function which is a particular type of objective function. Offspring are created by performing operations over the selected chromosomes. These operations are crossover and mutation.

The crossover copies the content of the parent's chromosomes in order to create offspring. In single point crossover, a crossover point is selected; the binary string from the beginning of chromosome to the crossover point is copied from one parent, the rest is copied from the other one. In two point crossover, two crossover points are selected; the binary string from the beginning of the chromosome to the first crossover point is copied from one parent, the part from the first to the second crossover point is copied from the second parent, and the rest is copied from the first parent. In case of uniform crossover, bits are randomly copied from both parents. The arithmetic crossover uses an arithmetic operation on the parents' chromosomes to make a new offspring.

The mutation can be performed by bit inversion on randomly selected bits or also by any local search or even metaheuristic operator on the chromosome (in which often called memetic algorithms).

The selection-crossover-mutation steps are repetitively performed until the new population is completed. The new population is then used in the next iteration of the algorithm. Commonly, the algorithm terminates when either a maximum number of generations has been produced, or a fitness function has been satisfied.

The fitness function is defined over the genetic representation and measures the quality of the represented solution. It is always problem dependent.

According to Aardal et al. [5], the most common way to represent a solution is that each chromosome is a vector s of length equal to the number of vertices. s_j is simply the frequency assigned to v_j. To cope with multiple demands, split graph model is used. The split graph is a graph in which the vertices can be partitioned into a clique and an independent set [56].

In a second representation, each chromosome is a partition of vertices in a family of n subsets $S_{f_1}, \ldots, S_{f_n}$ where S_{f_i} is the set of vertices that are assigned the frequency f_i. In a third representation, each chromosome is a permutation of the vertices representing the canonical assignment (smallest frequency first). Similar to the first representation, a split graph model is used for multiple demands. These chromosome representations are denoted by *Rep1*, *Rep2* and *Rep3* respectively.

Asexual crossover is introduced in Cuppini [38]. It consists of choosing two genes G1 and G2 in a chromosome and apply crossover points in both genes. A child is created by completing the first part of G1 with the second part of G2 and the first part of G2 with the second part of G1. A chromosome is represented as *Rep2* and chosen with probability proportional to its fitness value which is the weighed sum of the interference level and the frequency span.

Kapsalis et al. [85] model the fitness function consisting of a weighted sum of the number of distinct frequencies, of violated constraints, and mobility costs. Two chromosome represenions are used. The first is similar to *Rep1*, the second is *Rep2*. Two crossovers are used. The first crossover considers exchanging frequencies assigned to a pair of vertices u, v in one parent to those in another parent if the selected constraint with the corresponding vertices u, v is satisfied in both parents. The second crossover is a single-point one with a crossover point set by considering hard constraints. Two mutations are used. The first mutation consists of changing a pair of vertices and swapping their frequencies. The second mutation consists of choosing a pair of vertices whose frequency assignments violate a hard constraint, then randomly assign new non-violated frequency to each of them.

In Dorne and Hao [45], the *Rep1* chromosome representation is used. The fitness function is the number of unsatisfied constraints. A chromosome is selected from the current population by favoring the elements not yet trapped in local optima. The child is obtained only by mutation, no crossover is applied. The mutation is given by selecting an infeasible assignment (frequency assignment to a vertex that violate one or more constraints) and replacing it with the best alternative. Nonetheless, the child will be accepted either if its fitness function is not worse than its parent's or randomly with a given probability. The algorithm is extended to cope for multiple demands in [46]. Later, by Hao and Dorne [70], three crossovers are introduced; they are single-point, uniform and conflict based.

A three-point crossover is applied in Lai and Coghill [105] while a random mutation to each gene is applied.

Jaimes-Rimero et al. [80] propose a local search algorithm for generating the new population. After a solution with zero blocking probability is found by a genetic algorithm, a local search is used for minimizing the overall interference level.

In Velenzuela et al. [153], the *Rep3* chromosome representation is used. The mutation operator consists of exchanging the position of two vertices. The fitness function is the span of the permutation. The selection of the first parent is made according to the circular ordering. The second parent is chosen with probability proportional to its fitness value.

Genetic-Fix algorithm is proposed by Ngo and Li [126]. A chromosome is represented by *Rep2*. Crossover and mutation are designed to maintain the number of 1's in the chromosome (the number of assigned frequencies to each vertex is unchanged). A two-point crossover is applied by selecting an initial gene and a final gene and then swapping only subsets of genes between this two genes. The mutation is performed by randomly changing the assigned frequency to a different one.

Wang and Gu [158] consider both the objective function and the penalty function. They use stochastic ranking to select the chromosomes.

Kolen [98] proposes a new approach to mutation and crossover. The mutation operator consists of a 1-opt local search that converts the input solution into a 1-optimal solution. This method is applied to every chromosome including the new ones so, at any stage, all of the chromosomes are 1-optimal. The crossover is an optimal operator. Once the two parents are selected, the best possible combination of their genes is calculated to generate a single child.

Luna et al. [110] develop an evolutionary algorithm specifically for automatic frequency planning problem in GSM networks. Later, in [111], Luna et al. use the formulation proposed in [110] to develop and compare four different metaheuristics which are Genetic Algorithms, Scatter Search, Evolutionary Algorithms, and Local Search with Restarts. All of them utilize the same local search method which is customized for GSM network.

Segredo et al. [144] propose a multiobjectivised memetic algorithm which is based on the Non-Dominated Sorting Genetic Algorithm II (NSGA-II) to solve frequency assignment problem. The model provides benefits in terms of solution quality, and in terms of time saving. It has improved that previously known best frequency plans for two real-world network instances.

4.2.6 Ant colony optimization

Ant colony optimization (ACO) is a population-based metaheuristics derived from observation of real ants' behavior. According to Dorigo and Stützle [44], the main idea is that the self-organizing principles which allow the highly coordinated bahavior can be exploited to coordinate populations of artificial ants that collaborate to solve difficult optimization problems. To apply ACO, the optimization problem is transformed into the problem of finding the best path on a weighted graph. The artificial ants (hereafter ants) incrementally build solutions by moving on the graph. A move is controlled by two parameters: the attractiveness and the pheromone trail level. The attractiveness is based on the structure of the problems such as costs and constraints. The pheromone trail level takes into account the times a given move has been successful. A lower bound is required to fix the initial level of pheromones. Over time, the pheromone trail starts to evaporate, thus reducing its attractive strength. Pheromone evaporation also has the advantage of avoiding the convergence to a locally optimal solution. Pheromone trails are updated when all ants have completed the construction of their solution.

Abril et al. [6] use ACO to solve the frequency assignment based on the interference (constraint) graph. Initially, the vertices are randomly assigned frequencies. A given number of ants are then placed randomly at the vertices. At each iteration, each ant moves from the current vertex to the adjacent vertex having the greatest constraints with a probability p_n and replaces its frequency with a best one with a probability p_c. Both probabilities are adjustable parameters.

Maniezzo and Carbonaro [115] use ACO to solve MI-FAP. The initial lower bound is computed by solving the relaxation of the orientation formulation proposed in [19]. Every ant moves to a new vertex and selects a new frequency at each iteration. The solution produced by each ant is locally upgraded by local search.

Montemanni et al. [123] use ACO algorithm to solve MS-FAP. They start by fixing the frequency span to a sufficiently high value and iteratively minimize the interference in the fixed spectrum problem by running the algorithm. Once a zero cost assignment is found, the frequency span is decreased by one and the ACO algorithm is run again. A move is defined as assigning a frequency to a vertex. Instead of defining trail levels on moves from vertex u to v, they define them on vertex-frequency pairs giving an indication of how proficient it has been in the past to assign frequency f to the vertex v.

Lui et al. [109] combine multi-agent systems and evolutionary algorithms to solve the frequency assignment problem in WLAN network. The algorithm is modelled on the basis of the T-coloring problems. The objective is to minimize the frequency spectrum required for a given level of reception quality over the network.

4.2.7 Artificial neural network

Metaheuritics based on artificial neural network (ANN) mimic the natural learning process. An ANN consists of an interconnected group of artificial neurons. Solutions are generated by this network of neurons whose states represent the values of the variables involved in the model. The objective of the problem is represented by an energy function. In order to minimize it, the neurons change their states dynamically as a function of the states of the neighboring neurons.

Using neural networks for combinatorial optimization problems was first proposed by Hopfield and Tank [76] for solving the traveling salesman problem. In the FAP, a neuron V_{if} is usually associated with each vertex-frequency pair (i, f) where $i \in V$ and $f \in F_v$. Two neurons are coupled if the corresponding vertices are adjacent in the interference graph. The energy function generally consists of a weighted sum of interference constraints and frequency demand constraints. The state of a neuron is updated by a local updating rule.

Kunz [104] applies ANN for the MI-FAP. The coupling weight between two neurons V_{uf} and V_{wg} depends on the interference type of the corresponding frequency assignments (frequency f, g at vertices u, w).

In Kim et al. [93], the energy function takes into account several types of interference constraints and the level of the unsatisfied demand. This unsatisfied demand is taken as an additional input to each neuron, forcing the assignment of new frequencies to vertices.

Recently, Elhachimi and Guennoun [52] propose a hybrid approach for solving the FAP based on a neural network whose stimulation is constructed by a partial solution generated by a greedy algorithm. They use searching techniques in conjunction with hierarchical genetic algorithms for the optimization of the parameters and topology of the network.

4.2.8 Hyper-heuristic

Recently, Chaves-González et al. [30] solve frequency assignment problems using a new parallel hyper-heuristic approach. They obtain very high quality solutions that beat any other results published. The parallel hyper-heuristic is based on several complex metaheuristics. It searches within a search space of these heuristics and controls the metaheuristics output to distribute the workload according to the results obtained by each heuristic along the whole execution time of the system. Seven heuristics are used. They are hybridized with the same local search strategy proposed by Luna et al. [111]. These heuristics are Iterated Local Search [135], Variable Neighborhood Search [69], Greedy Randomized Adaptive Search Procedure (GRASP) [54], genetic algorithm, Population Based Incremental Learning [14], Scatter Search [63] and Artificial Bee Colony [87].

5 Frequency assignment in satellite communication system

In literature, frequency interferences in satellite systems are treated in two different ways. The first considers co-channel interference between two or more adjacent satellites (inter-system interference). The second considers a single satellite system where frequency interference comes from frequency reuse, e.g. by multiple spot beams.

Futabiki and Nishikawa [57] propose a Gradual Neural Network (GNN) to minimize co-channel interference between several satellite communication systems. The GNN consists of $N \times M$ binary neurons representing the N-carrier-M-segment system. This is based on the fact that each satellite operate with multiple carriers while each carrier may occupy different frequency bandwidths. To be able to model the interference, the carriers should be segmented into a number M of fixed-size channels. Co-channel interference is considered for each pair of channels with the same frequency. This interference is represented by an $M \times M$ interference matrix. The energy function is defined to represent the remaining constraint of the N-carrier-M-segment frequency assignment. The binary neural network achieves constraint satisfaction using heuristic methods, whereas the gradual expansion scheme seeks cost optimization.

Salcedo-Sanz et al. [141] also consider inter-system interference with the N-carrier-M-segment channel model. They propose a hybrid Hopfield network-simulated annealing algorithm (HopSA) for minimizing the co-channel interference. The HopSA algorithm consists of a fast digital Hopfield neural network which manages the problem constraints hybridized with a simulated annealing which improves the quality of the solutions obtained. Three different objective functions are considered. The first objective is to minimize the maximum peak of the interference between the systems. The second objective is minimizing the total interference of the systems. The third objective function takes into account the first two objectives. Nonetheless, the authors found that the HopSA shows lack of scalability, which leads to poor results when the size of the problem grows. Salcedo-Sanz and Bousoño-Calzón propose an improvement algorithm in [140].

Wang et al. [159] propose a new approach called a noisy chaotic neural network with variable thresholds (NCNN-VT) in order to solve inter-system frequency assignment problem. The NCNN-VT model consists of $N \times M$ noisy chaotic neurons. The NCNN-VT facilitates the interference minimization by mapping the objective to variable thresholds of the neurons. This technique obtains better solutions compared to both techniques proposed in [57, 141].

Salman et al. [142] present a number of algorithms based on differential evolution to order to solve inter-system frequency assignment problem. By rearranging frequencies of one set of carriers while keeping the other set fixed, the objective is to minimize the largest and total interference among carriers. They have investigate several schemes ranging from adaptive differential evolution to hybrid algorithms in which heuristic is embedded. Based on the same set of benchmark problems used in [57, 141, 159], the proposed algorithms outperform the existing ones both in terms of the quality of the solutions and computational time.

Sara et al. [10] presents an algorithm for resource allocation in a multi-spot satellite network to obtain a quasi-optimal time/frequency plan for a set of terminals with a known geometric configuration, under interference constraints. The study is based on spatial distribution of satellite spots and model interference based on geographical zones in that the users within the same zone exhibit the same radio propagation condition.

As a conclusion, the problem considered in this thesis differs from the one considered in the literature on frequency assignment for satellite communication. We consider resource allocation in multi-spot satellite similar to that of Sara et al. [10], but our study is based on dedicated spot-to-user concept and model interference based on each user's radio propagation condition.

The objective of our study is to assign as many users as possible to the system while these users can have different frequency demands. To the problem classification proposed by Aardal et al. [5], our problem is closest to the Max-FAP problem. Nonetheless; instead of assigning as many frequencies as possible to the vertices (users), we look for a solution such that either the frequencies are assigned to the vertices according to their demands or no frequency assignment at all to some vertices.

A large part of the work is motivated by a collaboration between Thales Alenia Space, LAAS and IRIT [77, 11]. In the considered industrial problem, the need for repeated simulations and integration of the FAP solution methods in a dynamic context oriented the choice of fast (greedy) algorithms rather than more time consuming metaheuristics. On the other hand, exact methods are also considered either to measure the distance of the greedy algorithms from optimum or to compare the best performance that can be obtained by different FAP models for satellite communications (cumulative vs. binary interferences, fixed vs. user-dedicated beam).

Problem description

2

1 Introduction

This chapter provides description of the problem studied in this thesis. In section 2, an overview of a satellite communication system is given. System and resource assignment concepts are briefly described. These concepts are covered in more details in the subsequent sections. Section 3 depicts how the satellite and users communicate wirelessly. This wireless communication should follow the quality criteria presented in Section 4. Section 5 provides a resource assignment framework. Constraints on resource assignment are then discussed in two subsequent sections. Problem summary, problem classification and the treatments are proposed in the last section.

2 Satellite communication system

A satellite communication system in the context of our study is based on the model and requirements provided by Corbel and Houssin et al. in [33] and [78]. According to Corbel, this study could provide resource assignment for the pre-defined scenarios supported by various air interfaces such as DVB-S2 [3] and DVB-RCS [2].

The system consists of a satellite, a gateway and a number of user terminals. The satellite acts as a relay point between the users and the gateway providing bi-directional communication links between the two parties. The gateway, which is excluded in our study, is a communication node that connects the satellite system to the terrestrial network.

The user terminals are ground-based and can be referred to as the Earth stations. They are randomly generated and randomly positioned within a ficti-

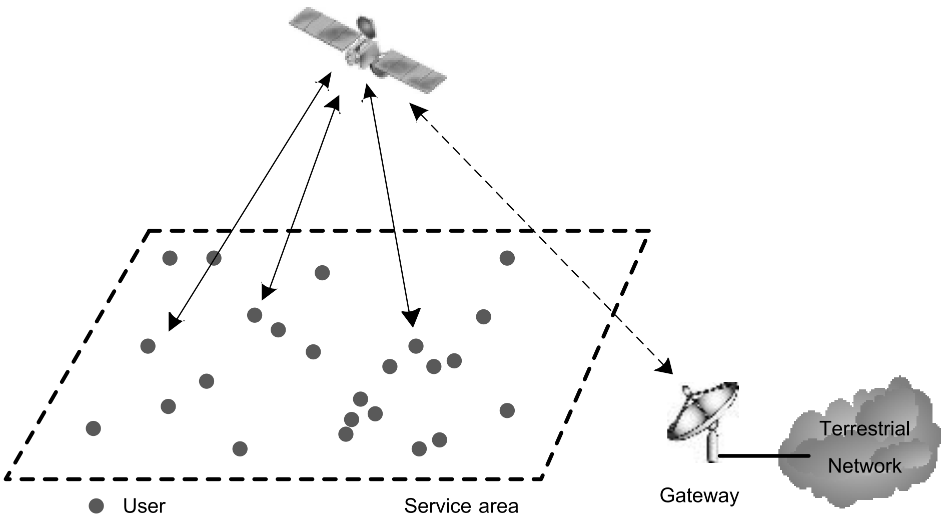


Figure 2.1: A satellite communication system.

tious rectangular service area defined by a set of geographic coordinates. The satellite's orthogonal projection is defined to be at the position $(0, 0)$ of the service area.

The satellite is equipped with antennas or antenna array that can provide a number of satellite beams, one beam for each of the users (this assumption is dropped in Chapter 5 by limiting number of beams). With the SDMA feature, the beam can be directed to any position within the service area. The beam shape is related to the antenna gain described in Section 3.

Each user is associated with a demand with which the system tries to accommodate by assigning a frequency or a block of contiguous frequencies $f_n, f_{n+1}, \ldots, f_{n+d-1}$ depending on the size of the demand d. There are a limited number of frequencies that the system can assign; nonetheless, assigning the same frequency to other users (*i.e.* frequency reuse) can be performed providing that this will not cause excessive frequency interference to the users.

It is assumed that there is no adjacent channel interference between frequency pairs (f_n, f_{n+1}) or (f_{n-1}, f_n) where n corresponds to frequency channel number. According to this, we only have to deal with co-channel interference ($f_i \cap f_j \neq \emptyset, i \neq j$).

The frequency assignment is also associated with a time duration. This time duration is given based on superframe concept, see Section 5. This superframe concept allows several users to access the system resource divided in time or so called Time Division Multiple Access (TDMA). A user could occupy the whole superframe duration or a fraction of it. In the latter case, the remaining time can be allocated to other users. Several users can thus

Chapter 2

share the same frequency but at different time. To be exact, we have to deal with both frequency and time so our problem considers $frequency \times time$ assignment and can be considered as MF-TDMA (Multiple Frequency TDMA).

The $frequency \times time$ assignment is fixed within a specific superframe instant; nonetheless, in order to support demand change of the existing users or addition or removal of users, it can be modified on the next instant. We only consider a valid assignment within an arbitrary instant.

Several superframes can be constructed simultaneously. Co-channel interference could occur if the same frequency is used at the same time and the users are geographically close to each other. The interference is cumulative; nonetheless, the system can tolerate this interference if its cumulative level does not exceed a predefined threshold, see Section 4 and 6.

The objective of the study is to serve as many users as possible. A user is considered served if it is assigned with a resource in time and frequency satisfying the technical and interference constraints.

3 Antenna gain and satellite beam

Antenna is an essential component in wireless communication systems. It is a device that is used for radiating and receiving radio waves. In addition to receiving or transmitting energy, an antenna is usually required to optimize or accentuate the radiation energy in some directions and suppress it in others [13].

An antenna is characterized by a radiation pattern and other parameters such as operating frequency, gain, aperture, efficiency and polarization. The radiation pattern is a graphical representation of the antenna radiation properties in free space. Typically, the radiation pattern is represented by a three dimensional plot or polar plots of the horizontal and vertical cross sections. The antenna gain is defined as the ratio of the intensity radiated by the antenna in the direction of its maximum output, at an arbitrary distance, divided by the intensity radiated at the same distance by a hypothetical isotropic antenna [31].

In our study, we deal with a simplified version of the radiation pattern having the antenna gain defined by Corbel and Houssin et al. [33], [78] as

$$G_{Sat}(u, v, u_0, v_0) = G_1 \cdot G_2(u, v, u_0, v_0) \cdot G_3(u, v) \tag{2.1}$$

where

$$G_1 = \eta \left(\frac{\pi D}{\lambda}\right)^2, \tag{2.2}$$

$$G_2(u,v,u_0,v_0) = \left(\frac{2J_1\left(\frac{\pi D}{\lambda}\sqrt{(u-u_0)^2+(v-v_0)^2}\right)}{\frac{\pi D}{\lambda}\sqrt{(u-u_0)^2+(v-v_0)^2}} \right)^2, \tag{2.3}$$

and

$$G_3(u,v) = \left(\frac{2J_1\left(\frac{\pi d}{\lambda}\sqrt{u^2+v^2}\right)}{\frac{\pi d}{\lambda}\sqrt{u^2+v^2}} \right)^2. \tag{2.4}$$

$J_1(x)$ represents the Bessel function of the first kind while u, v and u_0, v_0 are geographic coordinates of the user and the position where the antenna's gain is maximum. η, D, d and λ are the antenna efficiency, the antenna diameter, the diameter of the antenna's primary source and the carrier wavelength, respectively.

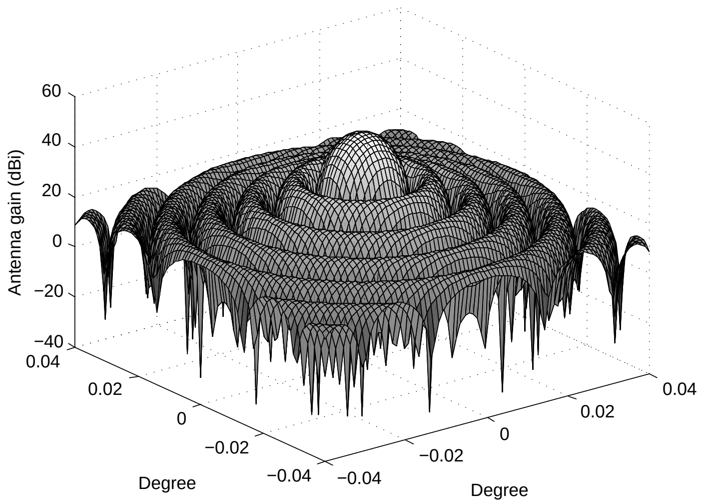


Figure 2.2: Antenna radiation pattern.

The corresponding antenna radiation pattern is shown in Figure 2.2. The gain is very high at the center and diminishes rapidly elsewhere. This results in concentration of power of satellite signal covering only a limited geographic area on Earth. We call this concentrated signal as a spot beam. The position where the antenna's gain is maximum is hereafter called the beam center.

SDMA satellite equips with antennas or antenna array that can produce a number of spot beams. With the antenna beam forming technology, these beams can be directed to the required geographic positions. By centering a beam over a user, this user will get maximum gain hence maximum signal power.

Figure 2.3 shows cross sections ($Y = 0$) of three satellite beams associated to and centered at users i, j, k located at three different geographic positions. Let's assume uniform receivers, transmitter output power and propagation loss, we can consider the received signal power from the perceived antenna gain. At position $(0, 0)$ the antenna gain from $Beam_i$ is at its maximum. It can be seen that, at this position, there exist also gains from $Beam_j$ and $Beam_k$. Interference occurs if these users share the same frequency (at the same time). The interference is cumulative in that the total interference at user i is the sum of the interferences from user j and k.

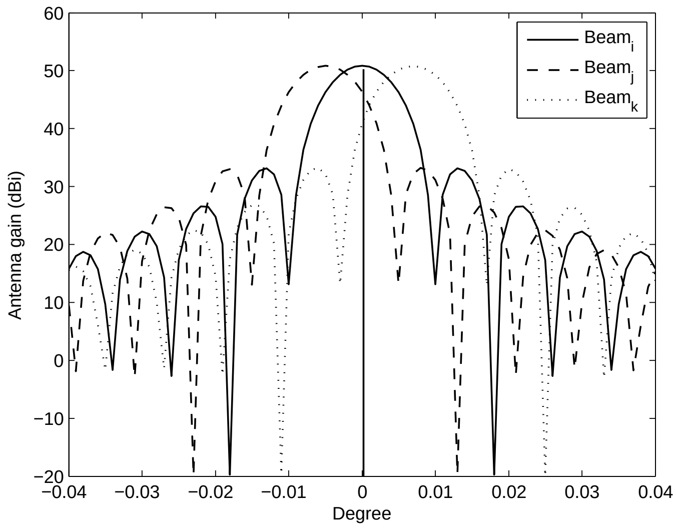


Figure 2.3: Cross section of three satellite beams.

4 Link budget

The link budget accounts all of the gains and losses from the transmitter to the receiver. It allows us to verify if the transmitted signal is correctly received at the receiver. The evaluation of the quality of the reception can be done by verifying the signal to noise ratio which is defined as the ratio of the desired signal power to the noise (unwanted signal) power. The noise can be viewed as a combination of thermal noise and interference. We thus represent this signal to noise ratio as $\left(\frac{C}{N+I}\right)$.

For a successful reception, this $\left(\frac{C}{N+I}\right)$ should not be less than a given value, a required signal to noise ratio, denoted by $\left(\frac{C}{N}\right)_{Required}$ *i.e.*

$$\frac{C}{N+I} \geq \left(\frac{C}{N}\right)_{Required} . \tag{2.5}$$

The required signal to noise ratio varies depending on type of communication technology and implementation. In our study, this value is initially fixed. It becomes variable later based on the selected symbol rate (RS), modulation and coding scheme which are altogether denoted by RsModCod.

Our problem concerns a communication link between the gateway and the user. For a typical user i, the signal to noise and interference ratio is given by [33]

$$\begin{aligned} \left(\frac{C}{N+I}\right)^{-1} &= \left(\frac{C}{N}\right)^{-1}_{Feeder} + \left(\frac{C}{I}\right)^{-1}_{Feeder} + \\ & \left(\frac{C}{IM}\right)^{-1} + \left(\frac{C}{N}\right)^{-1}_{i} + \left(\frac{C}{I}\right)^{-1}_{i} . \end{aligned} \tag{2.6}$$

$\left(\frac{C}{N}\right)_i$ and $\left(\frac{C}{I}\right)_i$ concern the link between the user i and the satellite and correspond to the user's signal to noise ratio and the user's signal to interference ratio. $\left(\frac{C}{N}\right)_{Feeder}$ and $\left(\frac{C}{I}\right)_{Feeder}$ concern the link between the gateway and the satellite and correspond to the feeder's signal to noise ratio and feeder's signal to interference ratio. $\left(\frac{C}{IM}\right)$ is the system's signal to intermodulation product ratio. These ratios are depicted in Figure 2.4 where *Up* and *Down* refer to uplink and downlink respectively.

There are two types of satellite links: forward and reverse. The forward link concerns a communication link from the gateway toward the user while the reverse link concerns the opposite direction. Each link is divided into two parts: user part and feeder part. Forward link consists of an uplink feeder part and a downlink user part while the reverse link consists of an uplink user part and a downlink feeder part.

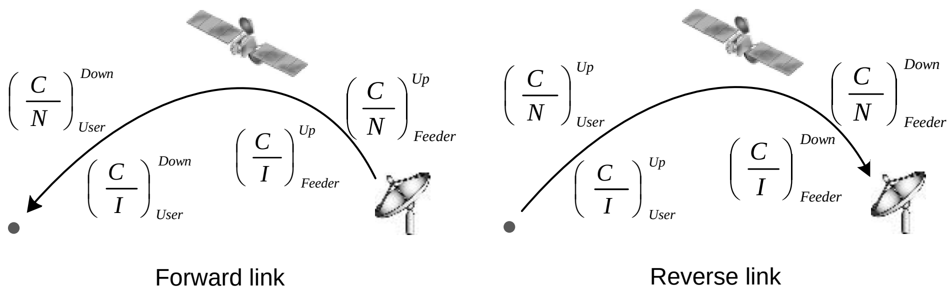


Figure 2.4: Satellite links.

Positions of the gateway and the satellite are fixed, so as the path between them. The terms $\left(\frac{C}{N}\right)_{Feeder}$, $\left(\frac{C}{I}\right)_{Feeder}$ and $\left(\frac{C}{IM}\right)$ are thus constant in our study. The $\left(\frac{C}{N}\right)_i$ are defined by

$$\left(\frac{C}{N}\right)_i^{Up} = \frac{(EIRPTerm_i/RS_i)}{L_{AtmoUp} \cdot L_{FSLUp}} \cdot \frac{G_{Sat(Beam_i \to i)}}{(T_A + T_{Rep})} \cdot \frac{1}{k}, \tag{2.7}$$

$$\left(\frac{C}{N}\right)_i^{Down} = \frac{(G_{Sat(Beam_i \to i)} \cdot PD_{Sat(Beam_i)} \cdot RS)_i}{L_{AtmoDown} \cdot L_{FSLDown}} \cdot \left(\frac{G}{T}\right)_{Term} \cdot \frac{1}{k}. \tag{2.8}$$

In the uplink, $EIRPTerm$ and RS represent the Earth terminal's effective isotropic radiated power and the utilized symbol rate. Note that different symbol rate can be selected based on the demand and the radio link quality. L_{AtmoUp} and L_{FSLUp} are the uplink atmospheric loss and the uplink free space loss. $G_{Sat(Beam_i \to i)}$ corresponds to the antenna gain of the associated satellite beam at the user's position. The antenna and repeater equivalent noise temperature are denoted by $(T_A + T_{Rep})$ while the Boltzmann constant is denoted by k.

In the downlink, the $PD_{Sat(Beam_i)}$ corresponds to the power spectral density of the antenna beam associated to user i. G/T is the figure of merit of the ground terminal which is the ratio of the antenna gain to the system noise temperature. Losses are defined in the same way as those in the uplink. The $\left(\frac{C}{I}\right)_i$ are defined by

$$\left(\frac{C}{I}\right)_i^{Up} = \frac{K_i \cdot G_{Sat(Beam_i \to i)}}{\sum_{j \in \mathit{Inf}} \cdot K_j \cdot G_{Sat(Beam_i \to j)}}, \tag{2.9}$$

$$K_i = (EIRPTerm_i/(RS_i \cdot L_{AtmoUp} \cdot L_{FSLUp})), \tag{2.10}$$

$$\left(\frac{C}{I}\right)_i^{Down} = \frac{(G_{Sat(Beam_i \to i)} \cdot PD_{Sat(Beam_i)} \cdot RS)_i}{\sum_{j \in \mathit{Inf}} \cdot (G_{Sat(Beam_j \to i)} \cdot PD_{Sat(Beam_j)} \cdot RS)_j} \tag{2.11}$$

where $j \in \mathit{Inf}$ refers to an interferer j from a set *Inf* of interferers to the user i.

Interference is more critical on the reverse link where signals and interferences from the users are groomed together. According to this, we only need to consider the frequency assignment problem on the reverse link.

5 Superframe structure

Since the time and frequency should be assigned from the shared resources, there should be a reference entity that we can work with. By this, slot, frame and superframe concept can be created. Each user i has a demand, expressed in terms of bitrate (Mbps). To fulfill this demand each user must be assigned a rectangle in the $frequency \times time$ space, called a slot. A slot can take different shapes for a given user demand. For example, a user with 10 Mbps demand could be assigned one of these slot combinations: $\{(5 \times 10), (10 \times 5), (25 \times 2)\}$. In fact these combinations form a discrete set of assigned frequencies together with the corresponded time duration. Slots are placed inside a larger fixed-size rectangle, called a superframe which is delimited by the available bandwidth *BWAvail* and a time duration *FrameDuration*. Interference is not allowed inside a superframe thus there is no overlapping among the associated slots.

In practice users may share the same satellite beam. In this case, their slots should be grouped inside a logical structure called a frame. A frame has a fixed time duration which equates *FrameDuration* but varies in frequency depending on how the associated slots are grouped. A frame is associated exactly to one satellite beam.

A superframe houses a number of frames thus accommodates a group of users that occupy segmentation of time and frequency. It is also repeated over time. If an instant of superframe starts at time T, the next instant of it will start at time $T+$*FrameDuration*. Content of the superframe can be changed between the instants. This allow adjustment of user demand, adding or removing users.

At any instant, more than one superframe can be created in order to accommodate other groups of users. In this sense, the frequency resource is reused. This frequency reuse may create interference among users sharing the same frequency at the same time. We consider frequency assignment in an arbitrary superframe instant.

Figure 2.5 gives an example of two superframes A and B at time instants N and $N + 1$. At time instant N, Frame 4 of Superframe A consists of Slot 1 and 2 while Frame 4 of Superframe B consists of Slots 3 and 4. It is mandatory that all slots within the same superframe should not overlap. In other words, there is no interference between Slot 1 and 2 and between Slot 3 and 4. Interference presents when slots from different superframes overlap in time or frequency or both. In this example, there is interference between slot pairs (1-3), (2-3) and (2-4).

Since interference is depended on the overlapping in time and frequency, we define an interference overlapping ratio r_{ij} as

$$r_{ij} = o_{ij}/\text{Area}_i, \tag{2.12}$$

while o_{ij} denotes an overlapping area between Slot i and j and Area_i is the area of Slot i.

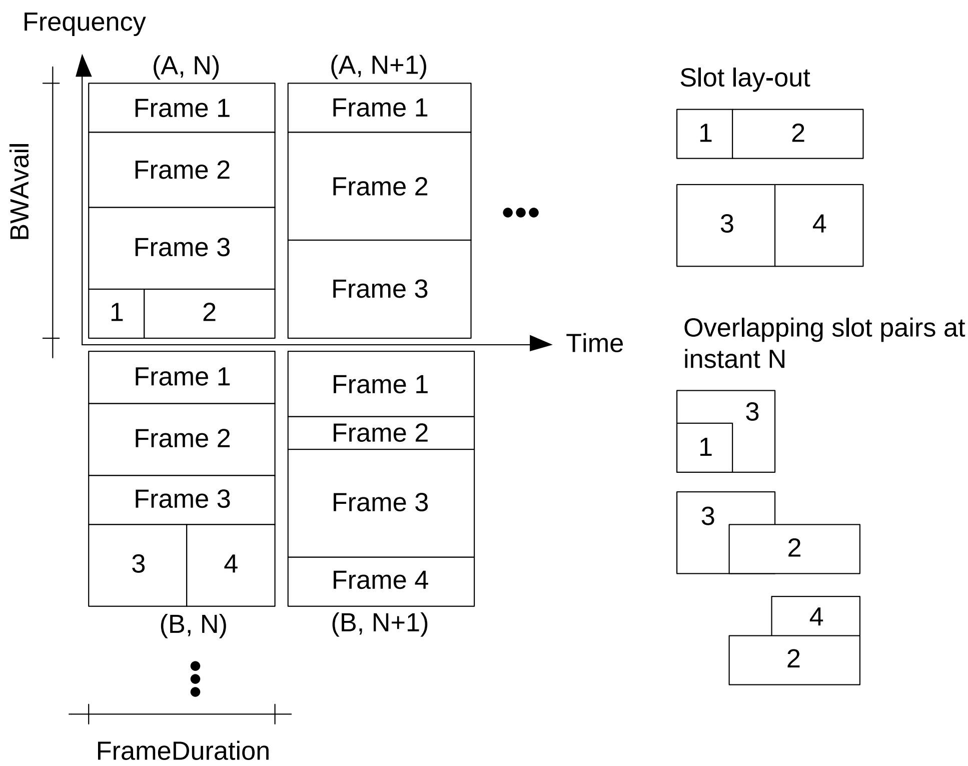


Figure 2.5: Superframe structure.

With this r_{ij}, Equations (2.9) and (2.11) can be rewritten as

$$\left(\frac{C}{I}\right)_i^{Up} = \frac{K_i \cdot G_{Sat(Beam_i \to i)}}{\sum_{j \in Inf} r_{ij} \cdot K_j \cdot G_{Sat(Beam_i \to j)}}, \tag{2.13}$$

$$\left(\frac{C}{I}\right)_i^{Down} = \frac{(G_{Sat(Beam_i \to i)} \cdot PD_{Sat(Beam_i)} \cdot RS)_i}{\sum_{j \in Inf} r_{ij} \cdot (G_{Sat(Beam_j \to i)} \cdot PD_{Sat(Beam_j)} \cdot RS)_j} \tag{2.14}$$

Interference overlapping ratio r_{ij} indicates how much user i get interfered by user j. Consider the slot pair (1-3), r_{13} = 1 while $r_{31} < 1$.

Interference is also cumulative in that (1-3) and (2-3) contribute to cumulative interference in Slot 3 while (2-3) and (2-4) contribute to cumulative interference in Slot 2.

Figure 2.6 shows frame and slot structure. Their associated satellite beams are presented by circles. Slot and frame parameters are provided with their units in order to give a general idea on their sizes.

Several users or slots can share the same satellite beam. In this case, the user which is closer to the center of the beam will get higher antenna gain (corresponding to higher signal power). In a dedicated beam configuration,

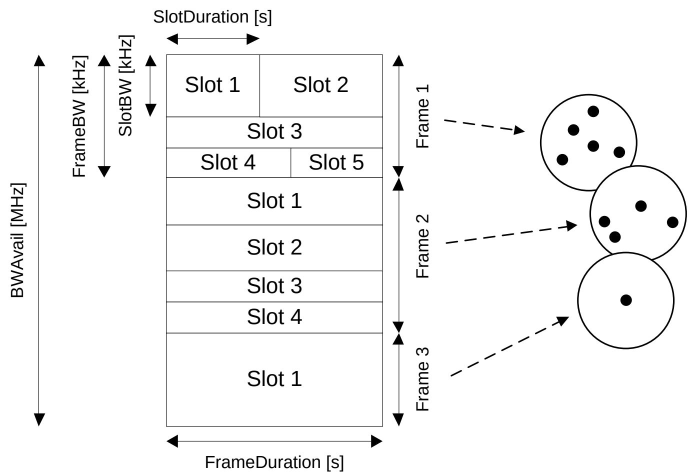


Figure 2.6: Frames and slots.

each user is assigned to a dedicated beam which is centered to the user's location. This gives each user the highest possible antenna gain.

Two remarks can be made concerning our study with respect to the superframe structure described in this section:

First, in this study we consider only the static problem, i.e. assignment of slots to frames and then superframes for a fixed time instant N. Hence the time flexibility we may use is limited to the *FrameDuration* parameter.

Second, for mathematical modeling and algorithmic solving the frame and superframe concepts are not necessary. We just need to specify for each user the size of its slot according to its demand and its position in the time. However the frame and superframe concepts are useful for an engineer to visualize the non-interference between slots (i.e. they belong to the same frame/superframe) and the assignment of users to beams (i.e. they belong to the same frame). In practice, the frame concept is not so precisely defined. In most of our methods the concept of frame and superframe will be ignored, concentrating on slot assignment in frequency and time taking account of interference constraints (see Section 6) and with additional constraints on slot positioning (see Section 7).

6 Interference constraints

Considering the reverse link, Equations (2.5), (2.6) and (2.13) can be rewritten in a linear form. By setting

$$A = \left(\frac{C}{N}\right)^{-1}_{Feeder} + \left(\frac{C}{I}\right)^{-1}_{Feeder} + \left(\frac{C}{IM}\right)^{-1},$$
$$B = \left(\frac{C}{N}\right)^{-1}_{i},$$
$$D = \left(\frac{C}{N}\right)_{Required},$$

we can express the Equation (2.5) as

$$\frac{1}{A + B + \frac{\sum_{j \in Inf} r_{ij} \cdot K_j \cdot G_{Sat(Beam_i \to j)}}{K_i \cdot G_{Sat(Beam_i \to i)}}} \geq D,$$

and finally:

$$\sum_{j \in Inf} r_{ij} \cdot \delta_{ij} \leq \alpha_i \tag{2.15}$$

where

$$\delta_{ij} = D \cdot K_j \cdot G_{Sat(Beam_i \to j)}, \tag{2.16}$$
$$\alpha_i = K_i \cdot G_{Sat(Beam_i \to i)} \cdot (1 - AD - BD). \tag{2.17}$$

α_i involves constants and parameters related only to user i while δ_{ij} involves constants, parameters of user j and gain from $Beam_j$ to user i.

If we conceive α_i as the acceptable interference threshold of user i and δ_{ij} as the interference coefficient of user j towards user i, the Equation (2.15) can be considered as the cumulative interference constraints for the user i. This cumulative interference constraints will be used by most of the models presented in this thesis.

The α_i and δ_{ij} depend mainly on the user's position and beam's position. When the beam center is fixed, parameters α_i and δ_{ij} are fixed and the interference level between two users solely depends on their overlapping in time and frequency. Nevertheless; these parameters may become decision variables when it is possible to move (quasi-continuously) the beams.

For example, consider two users i, j located at coordinates (u_i, v_i) and (u_j, v_j) which are at $(0.0314, 0.0008)$ and $(0.0398, -0.02)$. The satellite is at position $(0, 0)$ and provides two beams, each center at users i, j e.g. $(b\_u_i, b\_v_i) = (0.0314, 0.0008)$ and $(b\_u_j, b\_v_j) = (0.0398, -0.02)$. If a beam is moved by a constant value Δ. The corresponding antenna gains G_i and G_j, the threshold α and the coefficient δ are changed as shown in the Table 2.1 below.

Table 2.1: Parameter changes as of beam move.

		$\times 10^{5}$		$\times 10^{-19}$		$\times 10^{-20}$	
(b_u_i, b_v_i)	(b_u_j, b_v_j)	G_i	G_j	α_i	α_j	δ_{ij}	δ_{ji}
(u_i, v_i)	(u_j, v_j)	1.04	0.88	8.53	7.05	5.53	6.51
$(u_i, v_i + \Delta)$	(u_j, v_j)	0.37	0.88	2.20	7.05	0.45	6.51
(u_i, v_i)	$(u_j, v_j + \Delta)$	1.04	0.31	8.53	1.67	5.53	0.53

7 Technical constraints

Although the frame structure gives two degrees of freedom on time and frequency resource assignment (recall that that a slot must be of rectangular shape); in reality, there are hardware limitations in a way that we cannot freely utilize some configurations in the assignment. These limitations give rise to additional frame structure constraints listed as following [33] (see also Figure 2.7):

(i) The size of the assigned frequency(ies) should be conserved during the frame duration *e.g.* if a slot is assigned with two frequencies during the first half of the frame, the remaining half can only be assigned with two frequencies basis. It is not allowed to slice the two frequency chunk into two assign more than two frequencies.

(ii) A slot cannot be to assigned with multiple frequencies taking different $SlotDuration$. The slot assignment should be rectangular in shape.

(iii) A slot should be continuous in time.

8 Problem summary and treatments

Summarily, given n the number of users, $U = \{1, \dots, n\}$ a set of users, (f_i, t_i) the frequency/time sets assigned to user i, α_i the acceptable interference threshold of user i, δ_{ij} the interference coefficient of user j towards user i and r_{ij} the overlapping ratio of user i by user j, the problem can be formulated as

$$\max \quad |\{i \in U | f_i \neq \emptyset, t_i \neq \emptyset\}|$$

subject to

Chapter 2

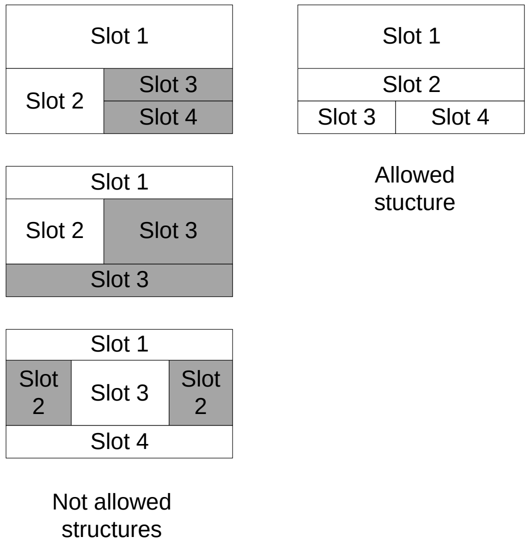


Figure 2.7: Technical constraints on frame structure.

Chapter 2

$$\sum_{j \neq i} r_{ij} \cdot \delta_{ij} \leq \alpha_i \quad \forall i, j \in U$$

and the given technical constraints.

We will treat the problem starting from its simplest form by relaxing a part of constraints and requirements. The simplest problem concerns assignment only one frequency to user using binary interference constraints. Difficulty and complexity increases when we deal with multiple frequency assignment, cumulative interference, and frequency-time assignment. According to these variations, we classify the problems into

- Single carrier frequency assignment
- Multiple carrier frequency assignment
- Frequency-time assignment

Single carrier frequency assignment is covered in the Chapter 3. Interference will be treated first as binary and later as cumulative. Moreover, we will fully consider the possibility of modifying the interference by moving the beams which will yield a mixed discrete / continuous non linear optimization problem.

We will drop the single frequency demand, but, as a counterpart, we will consider fixed beam centered configuration in Chapter 4. Users can ask for more than one frequency. Both binary and cumulative interference constraints will be covered. Later in the chapter, we treat both frequency and time in the assignment.

Chapter 5 considers full assignments and constraints provided by the industry (except beam moving). Two algorithms will be proposed and compared.

3 Single carrier models

1 Introduction

In this chapter we consider the frequency assignment problem in a simple form by discarding several constraints and requirements. All users have a uniform demand which equates 1 and a user is considered served if it is assigned a frequency. Apart from this, we omit the concept of *FrameDuration* and superframe configuration *i.e.* an assigned user occupies a frequency all the time. Interference occurs if users share the same frequency and for each couple of interferers, the interference overlapping ratio is 1. The objective is to serve as many users as possible.

We begin with a binary interference model. This model is treated as a baseline for performance comparison with the following two models that deal with cumulative interference.

For each of these cumulative interference models, we propose two solving methods to help achieve our objective. They are integer linear programming and greedy algorithm. Two integer linear programming models are devised. These models are solved by a commercial solver: IBM CPLEX [79]. Greedy algorithms are also elaborated.

As a counterpart for simplifying the FAP model, we consider the possibility of beam moving which utilizes SDMA technology to move some of satellite beams from their center positions. The move yields non-linear change of antenna gain and thus interference. By this change, local interference level can be reduced; as a result, more users can be assigned to the system. In addition to the discrete optimization problem of frequency assignment, Beam moving yields a non-linear continuous optimization problem that we solve through specialized methods.

The models are tested with 1,000 randomly generated instances: 100 instances of $20, 40, ..., 200$ users. For each instance, users' geographic coordinates are randomly generated. α and δ values are then calculated based on these coordinates, satellite beam positions, and constants as shown in Equations (2.16) and (2.17). These instances are also used in Chapter 4.

Test results are shown and compared in the Computational experiments section. Conclusions are provided at the end of the chapter.

This chapter is issued from publications [90], [92], [89], in collaboration with Frédéric Messine, Assistant Professor at INPT/IRIT, Toulouse. This study is also a follow-up to a collaboration between Thales Alenia Space and LAAS-CNRS that considered a further simplified model ignoring the beam moving possibilities [77], [78].

Section 2 is devoted to single frequency models with binary interference while cumulative interference models are considered in Section 3. Section 4 presents the beam moving algorithm. Section 5 provides computational experiments on a realistic set of data instances.

2 Single carrier model with binary interference

This model is created as a baseline for performance comparison with other two models. As it can be seen from Chapter 2 that the satellite beam's footprint is circular in shape. It gives the highest gain at the beam center and the gain diminishes rapidly when moving out of this position. The gain difference is almost 40dB at about 0.01 degree (in geographic coordinate system) away from the beam center. According to this, a beam radius can be chosen in that if we place the beams next to each other in a grid manner similar to the one shown in Figure 3.1, the interference level is low enough (considered as none) that we can reuse the same frequency f in the second-tier neighbours [33].

The satellite service area defined by a set of geographic coordinates $u = [-0.043980, 0.048520]$ and $v = [-0.021152, 0.012348]$ can be covered by 40 beams. The beam positions are fixed and regularly spaced. We call this a *fixed-beam* configuration. Assigning the same frequency to any adjacent beams is not allowed since it creates interference. This pairwise interference is generally referred to as binary interference.

The beams and their binary interference relations can be represented by a graph $G = (V, E)$; in which, a vertex $v \in V$ corresponds to a satellite beam and an edge $e \in E$ between two vertices $v, w \in V$ corresponds to interference if both use the same frequency. The graph can be defined by a $p \times q$ matrix called an incidence matrix, where p and q are the number of vertices and edges respectively. The corresponding interference graph with 40×40 incidence matrix is shown in Figure 3.2.

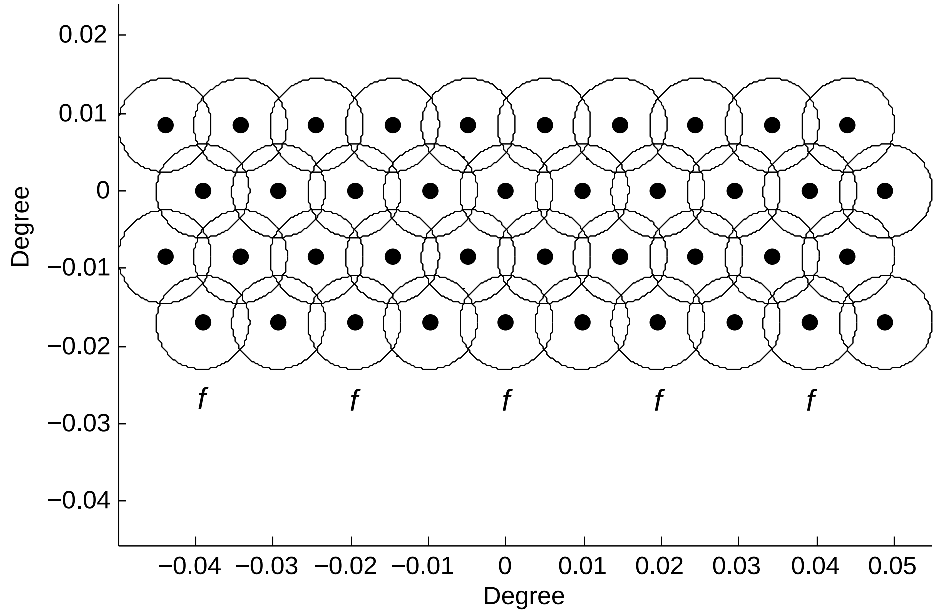


Figure 3.1: Placement of antenna beams.

We are interested in assigning frequencies to all of the beams in which the lowest number of frequencies are used. This problem is indeed a graph (vertex) coloring problem. Answering to the question if a graph is k-colorable (which corresponds to the simplest variant of the problem considered in this thesis) is already NP-complete [59].

To color the graph, we use DSATUR algorithm [25] which is a well-known and efficient constructive heuristic [1]. The algorithm provides an exact vertex coloring by sequentially coloring the vertices according to their saturation degrees. As a result, total of four colors are needed. The corresponding frequency assignment is shown in the Figure 3.3.

On the other hand, if we assign two frequencies per beam. With four colors corresponding to four frequency groups, each group assigned to 10 beams; the maximum number of users that can be served is 80: maximum 2 users per beam (providing that each should be assigned the nearest beam center). Nonetheless, if the users are not regularly distributed such as in the way that there are more than two users in some beams or there are less than two in others; the number of served users is reduced. Beams with less than two associated users yield waste of frequency resource.

[1]There are recently improved DSAT-based algorithms, the interested reader is referred to [127].

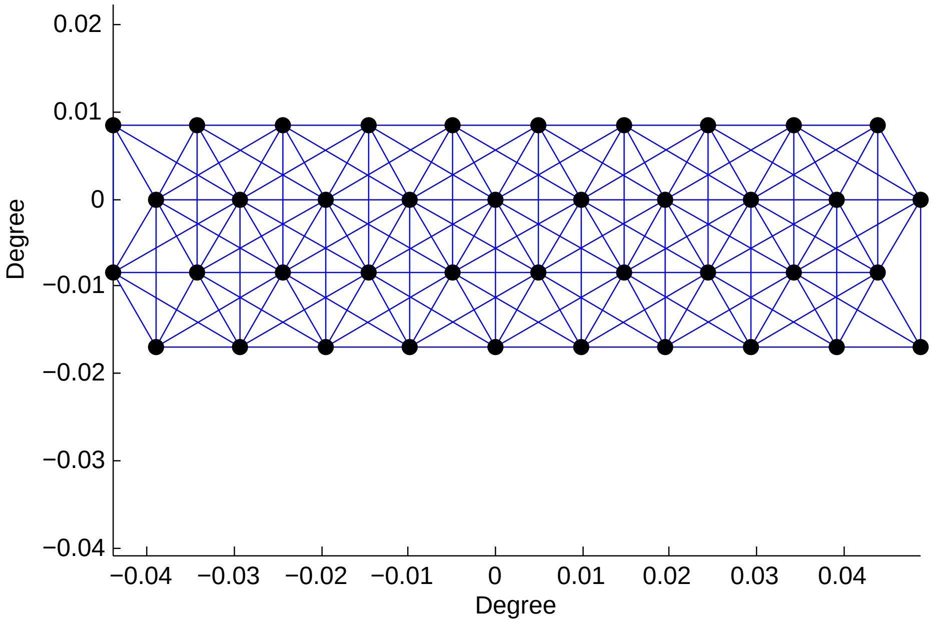


Figure 3.2: Interference graph.

3 Single carrier models with cumulative interference

In this section, we consider two models that deal with cumulative interference constraints which generalize the binary interference constraints. The first one is an extension of the baseline. The second one drops the fixed-beam configuration and uses SDMA-beam instead. Both models employ the same number of frequencies as the baseline.

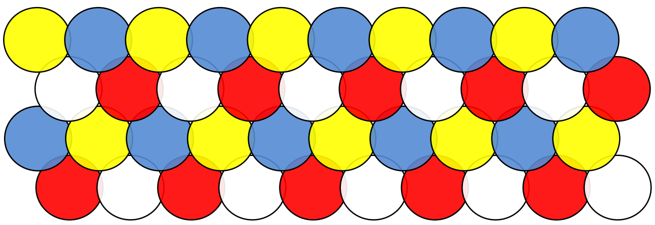


Figure 3.3: Assignment by four frequencies.

3.1 Fixed-beam varying-frequency model

In the first model, we preserve the 40 fixed-beam formation of the baseline (Model 1) but instead of fixing two frequencies per beam (leading to waste of frequency in beams having less than two associated users) we freely assign frequencies inside the beam. More than two users could be served per beam.

The frequency assignment is based on the cumulative interference concept given by Equation (2.15) but, in this case, with $r_{ij} = 1$ if i and j use the same frequency (note that the slot time equates *FrameDuration* for all users).

We limit the number of available frequencies to 8 which is equal to that of the baseline. The user could be assigned any of the 8 frequencies as long as the cumulative interference constraints are not violated. There is another exception in that users reside in the same beam should not be assigned the same frequency. This corresponds to the requirement in Chapter 2 providing that no resource overlapping or no interference is allowed inside a beam.

This model can be formulated as a combinatorial optimization problem as

$$\max \quad |\{i \in U | f_i \neq 0\}|$$

subject to

$$f_i \in \{0, 1, \ldots, F\},$$

$$f_i = f_j = 0 \text{ or } f_i \neq f_j \quad \forall i, j \in U_s, i \neq j \text{ and } \forall s \in S,$$

$$\sum_{\substack{j \in U \setminus \{i\} \\ f_j \neq 0 \\ f_j = f_i}} \gamma_{ij} \leq \beta_i \quad \forall i, j \in U,$$

giving that

- n the number of users,
- $U = \{1, \ldots, n\}$ a set of users,
- F the number of frequencies,
- m the number of satellite beams,
- $S = \{1, \ldots, m\}$ a set of satellite beams,
- U_s the set of users associated to the satellite beam s,
- β_i the acceptable interference threshold for user i (given that i is assigned to its nearest beam center),
- γ_{ij} the interference coefficient of user j towards user i (given that i is assigned to its nearest beam center).

3.2 SDMA-beam varying-frequency model

In the second model, we do not fix the beam positions. Instead, by assuming that unlimited number of beams can be generated, we use SDMA feature to center them over the users: one beam for each user. The user could be assigned any of the 8 frequencies as long as the cumulative constraints are not violated.

By centering the beam over the user, the user gets the highest gain. This yields higher user's acceptable interference threshold (*cf.* Chapter 2, Section 6) and in some case lower interference level. This model can be formulated as

$$\max \quad |\{i \in U | f_i \neq 0\}|$$

subject to

$$f_i \in \{0, 1, \ldots, F\},$$

$$\sum_{\substack{j \in U \setminus \{i\} \\ f_j \neq 0 \\ f_j = f_i}} \delta_{ij} \leq \alpha_i \quad \forall i, j \in U,$$

giving that

- n the number of users,
- $U = \{1, \ldots, n\}$ a set of users,
- F the number of frequencies,
- α_i the acceptable interference threshold for user i (given that a dedicated beam is centered on i),
- δ_{ij} the interference coefficient of user j towards user i (given that a dedicated beam is centered on i).

SDMA-beam model is expected to outperform the fixed-beam model. How the SDMA can provide improvement is illustrated in Figure 3.4. In the figure, in both models, three users U_A, U_B and U_C are located at the same geographic positions but the beam configuration is different. There are two fixed-position beams in the left model and three SDMA-beams each centered at the users in the right model. Note that each circle is just a simplification of a beam whose size represents a certain value of gain relative to the maximum at the beam center. Assuming that they are assigned with frequency f_1, f_2, f_1 respectively. U_A and U_C interfere with each other but the carrier to interference ratio (C/I) is higher in the SDMA case: the uplink signal power (C) of U_A is higher from having a dedicated beam centered to it. The uplink interference power (I) from U_C towards the beam of U_A is farer (hence weaker) compared to that of the fixed-beam model. This is the same case for U_C. The performance improvement is shown numerically in Section 5.

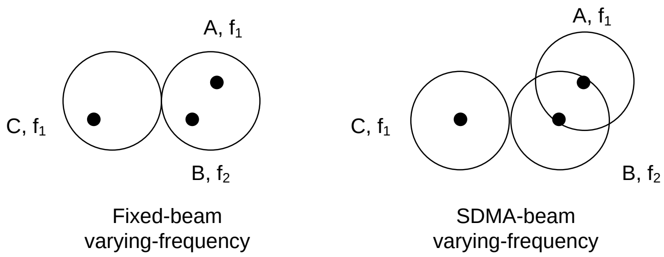


Figure 3.4: SDMA benefit.

For each of the two models mentioned above, we propose two frequency assignment methods which are integer linear programming and greedy algorithm. Both of them are presented below.

3.3 Integer linear programming

The linear interference constraints derived in Chapter 2 allow us to formulate the problems using Integer Linear Programming (ILP). Using the same notations given in the previous section, and by defining a binary decision variable x_{if} as

$$x_{if} = \begin{cases} 1 & \text{if user } i \text{ is assigned with frequency } f, \\ 0 & \text{otherwise,} \end{cases}$$

the problem can be represented by the following ILPs:

3.3.1 Fixed-beam varying-frequency model

$$\max \sum_{i=1}^{n} \sum_{f=1}^{F} x_{if} \tag{3.1}$$

$$\sum_{f=1}^{F} x_{if} \leq 1 \quad i = 1, \ldots, n \tag{3.2}$$

$$\sum_{i \in U_s} x_{if} \leq 1 \quad s = 1, \ldots, m \quad f = 1, \ldots, F \tag{3.3}$$

$$\sum_{j=1}^{n} \gamma_{ij} x_{jf} \leq \beta_i + M_i(1 - x_{if}) \quad i = 1, \ldots, n \quad f = 1, \ldots, F \tag{3.4}$$

$$x_{if} \in \{0, 1\} \quad i = 1, \ldots, n \quad f = 1, \ldots, F \tag{3.5}$$

The objective (3.1) maximizes the number of assigned users while constraints (3.2) restrict that at most one frequency has to be selected for each user. Constraints (3.3) impose that only different frequencies are allowed inside a beam. Constraints (3.4) concerns the cumulative interference. The constant M_i has to be large enough to withdraw these constraints if i is not assigned a frequency f ($x_{if} = 0$). More precisely, we set $M_i = \sum_{j=1}^{n} \gamma_{ij} - \beta_i$.

3.3.2 SDMA-beam varying-frequency model

$$\max \sum_{i=1}^{n} \sum_{f=1}^{F} x_{if} \tag{3.6}$$

$$\sum_{f=1}^{F} x_{if} \leq 1 \quad i = 1, \ldots, n \tag{3.7}$$

$$\sum_{j=1}^{n} \delta_{ij} x_{jf} \leq \alpha_i + N_i(1 - x_{if}) \quad i = 1, \ldots, n \quad f = 1, \ldots, F \tag{3.8}$$

$$x_{if} \in \{0, 1\} \quad i = 1, \ldots, n \quad f = 1, \ldots, F \tag{3.9}$$

This model is similar to the previous one except that there is no constraint inside a satellite beam. The α and δ are different from β and γ since the beams are now centered at the users. In this model, we could also set $N_i = \sum_{j=1}^{n} \delta_{ij} - \alpha_i$.

3.4 Greedy algorithms

Solving the ILP formulations provides optimal solutions for small instances. For large-sized instance, the ILP solver requires long calculation time to solve to optimality. In fact, for our problem, short calculation time is preferred in order to cope with demand change or user mobility (adding or removing users between superframes). According to this, we consider heuristic approach such as greedy algorithm.

Greedy algorithm is simple, fast, and based on a no-look-back principle that the decision made in the previous steps cannot be changed. In other words, the decision in the current or next steps should not nullify the past decisions. Nonetheless, the performance of the greedy algorithm could not be guaranteed and is mostly depended on how the algorithm is customized to the problem or problem data.

We propose greedy algorithms taking n (the number of users) iterations to assign each user a frequency from a set of available frequency $\{1, \ldots, F\}$. At each iteration, a user is selected according to a given criterion named *user priority rule*. Then, the selected user is either assigned a frequency or rejected according to a second criterion, the *frequency priority rule*. The assigned frequency should not created excessive interference to the already assigned users. The greedy algorithms are provided as following:

3.4.1 Fixed-beam varying-frequency model

Greedy algorithm for this model is provided in Algorithm 1. Let Q denotes a set of users that have not been assigned a frequency yet. Initially we have $Q = U$. f_i denotes the frequency allocated to user i. All users are initialized with $f_i = 0$. At each step of the greedy algorithm, a user i is removed from Q and is either rejected or assigned a frequency if $1 \leq f_i \leq F$ and $f_i = 0$ indicates that user i is rejected.

Input: n, F, β, γ
Output: f

```
f_i ← 0, ∀i = 1, ..., n
for q = 1 to n do
    i ← selectUser(n, F, β, γ, f)
    f_i ← selectFrequency(i, n, F, β, γ, f)
end
```

Algorithm 1: Greedy algorithm

There are many ways to select the users in *selectUser* function. Users could be ordered statically in that once the order is fixed, it remain unchanged throughout the calculation. Or users could be ordered dynamically in that the order is determined at each iteration based on the predefined rule. For example, we may use the interference margin, where the margin $M(i, f)$ of a user $i \in Q$ for a frequency f is given by $M(i, f) = \beta_i - \sum_{\substack{j \in U \setminus Q \cup \{i\} \\ f_j = f}} \gamma_{ij}$.

This margin corresponds to the positive or negative slack of the cumulative interference constraint for user i if it is assigned a frequency f.

As a preliminary result, we observed that the user priority rule aimed at selecting first the most constrained users in terms of available frequencies while it is well known that, with this environment, the DSATUR algorithm for standard graph coloring problem gives bad results. We thus consider a kind of hybrid reverse DSATUR rule by alternately selecting (1) the user having the largest number of available frequencies and (2) the user having maximum interference with the previously assigned users.

In fact, we tested two following user priority rules:

- Lexicographic: the user with the smallest number is selected,
- Hybrid: the user having the largest number of available frequencies is selected. A frequency f is available for user $i \in Q$ if $M(i, f) \geq 0$ and if for all users $j \in U \setminus Q$ that have already been assigned frequency f, $M(j, f) \geq 0$. In case of a tie, we select the user having the largest total margin for all its available frequencies. Let i denotes the selected user with this rule. For the next iteration, we select the user having maximum interference with i, *i.e.* the user j maximizing $\gamma_{ij} + \gamma_{ji}$ and we alternate the two rules.

For the frequency selection (*selectFrequency* function), we tested two following frequency priority rule:

- Lexicographic: the smallest available frequency is selected,
- Most used: the most used available frequency is selected. In case of a tie, we select the frequency f that maximizes the sum of margins $M(j, f)$ for all users $j \in Q$.

In both user and frequency selections, we verify also that the frequency has not already been used inside the same satellite beam.

3.4.2 SDMA-beam varying-frequency model

In this model, we assign each user a dedicated satellite beam. Greedy algorithms for this model is similar to the previous one except that we

- Replace β and γ with α and δ,
- Do not consider frequency reuse inside a satellite beam.

The proposed greedy algorithms run in $O(n^2F)$ time.

Chapter 3

4 Beam moving algorithm

4.1 Beam moving procedure given a fixed frequency assignment

To further improve the results from the ILPs and greedy algorithms, we propose a subsequent non-linear local optimization, called Beam Moving algorithm. This algorithm exploits the benefit of SDMA technology by moving a number of satellite beams in high interference areas from their center positions.

In fact the δ_{ij} and α_i in Equations (2.16) and (2.17) can be written as functions of user position (u, v) and beam position (b_u, b_v) which are

$$\delta_{ij} = D \cdot K_j \cdot G_{Sat}(u_i, v_i, b\_u_j, b\_v_j), \tag{3.10}$$
$$\alpha_i = K_i \cdot G_{Sat}(u_i, v_i, b\_u_i, b\_v_i) \cdot (1 - AD - BD). \tag{3.11}$$

The terms D and $(1 - AD - BD)$ are constant. We will keep the user position fixed but alter the beam position; as a result, both δ_{ij} and α_i change. Nonetheless, the changes are non-linear as of the non-linear antenna gain shown previously in Figure 2.2.

Beam Moving algorithm is shown in Algorithm 2. It takes the output solution (f) from either ILP or greedy algorithm and the user position u, v as its input, identifies the unassigned users, and, for each of these users, moves the most k interfering beams and tries to reassign that user a frequency $f_{BM} \in F$.

Let i denotes an unassigned user from the previous calculation ($f_i = 0$), the algorithm selects (Step 5) a test frequency f_{BM}, and identifies (Steps 6-7) a set of interferers U containing all users j having $x_{jf} = 1, \forall j \in U$ (all unassigned users are excluded). A set B which contains beam positions of U is also created.

U and the corresponding B are then tested with *LinkBudget* function. This function gives "margins" between the interferer U's current signal to noise ratios and their required signal to noise ratio thresholds. The idea is to check if these margins are not too low before we actually perform the beam move. To proceed with beam move, these margins should not be lower than *MAXINEG* parameter. If not, the remaining frequencies are tried or the user i is rejected. This margin test is based on the fact that when a beam is moved from its center, the corresponding antenna gain is reduced and the associated user's signal to noise ratio is decreased. Moving beams of these interferers will decrease their signal to noise ratio margins. If the margins are too low, the beams could not be moved too far from their centers.

Input: $F, u, v, f, \alpha, \delta, k$, *MAXINEG*, *UTVAR*
Output: Updated f, b_u, b_v

```
b_u_i ← u_i, ∀i = 1, ..., n
b_v_i ← v_i, ∀i = 1, ..., n
for i = 1 to n do
    if fi = 0 then
        for f_BM = 1 to F do
            U ← [u_j; v_j] | f_j = f_BM   ∀j = 1, ..., n
            B ← [b_u_j; b_v_j] | f_j = f_BM   ∀j = 1, ..., n
            margin ← LinkBudget(U, B)
            if min(margin) > MAXINEG then
                bool, bsol ← BeamMove(i, U, B, k, UTVAR)
                if bool = 1 then
                    b_u, b_v ← bsol
                    f_i ← f_BM
                    Break
                else
                    f_i ← 0
                end
            end
        end
    end
end
```

Algorithm 2: Beam Moving algorithm

Let $K \subseteq U$ consists of a set of users whose beams will be moved. The parameter k defines the number of strongest interferers to the unassigned user i that are included in the set K.

The parameter *UTVAR* $\in (0, 1)$, if set to 1, tells the algorithm to replace the least interferer in the set K with i thus including beam of user i in the move.

In Step 10 (see also Algorithm 3), the beams of users in the set K are continuously moved from their center positions $(u_0^{(k)}, v_0^{(k)})$. In each move the new positions are evaluated if the corresponding signal to noise ratios violate the link budget constraints. Any move that violates these constraints is rejected.

Input: $i, U, B, k,$ *UTVAR*
Output: $bool, bsol$

$d \leftarrow distance(B, i)$
sort d
if UTVAR $= 1$ **then**
    $x_0 \leftarrow [B_j; B_i] \quad \forall j = 1, \ldots, k-1$ (according to ordering index d)
else
    $x_0 \leftarrow [B_j] \quad \forall j = 1, \ldots, k$ (according to ordering index d)
end
while $LinkBudget(U, x_0) > 0$ *and* $Iteration <$ MAXITER **do**
    Move x_0
    $Iteration \leftarrow Iteration + 1$
    if $LinkBudget(u_i, b_i) > 0$ **then**
        $bool \leftarrow 1$
        Break
    end
end
$bsol \leftarrow [b; x_0]$

Algorithm 3: BeamMove function

The move, while reducing signal to noise ratios of the interferers, could benefit the unassigned user by reducing its tentative interference level. The signal to noise ratio of the user i is also tested, which if passes (allowing assigning the test frequency f to user i), the move terminates. Perhaps this could be easily viewed as Robin Hood's concept "robs from the rich and gives to the poor".

The move problem we want to solve can be represented as:

$$\min \sum_{k \in K} \| (u_0^{(k)} - u_k)^2 + (v_0^{(k)} - v_k)^2 \|^2, \tag{3.12}$$

$$\left(\frac{C}{N+I}\right)(u_i, v_i, u_0^{(i)}, v_0^{(i)}) \geq \left(\frac{C}{N}\right)_{Required}, \tag{3.13}$$

subject to

$$\left(\frac{C}{N+I}\right)(u_k, v_k, u_0^{(k)}, v_0^{(k)}) \geq \left(\frac{C}{N}\right)_{Required} \qquad \forall k \in K. \tag{3.14}$$

If a decent move could not be found within a number of iterations defined by *MAXITER*, each of the remaining frequencies is tried. If all frequencies have been tried and there is no possible solution, the user i is rejected and the algorithm moves to next unassigned users.

Figure 3.5 shows an example of frequency assignment for 5 users with their beams centered on them. Four users can be allocated with the Frequency 1 or 2 as shown next to the users. Frequency 0 means that the user cannot be assigned a frequency. The corresponding α_i and δ_{ij} are shown in Table 3.1.

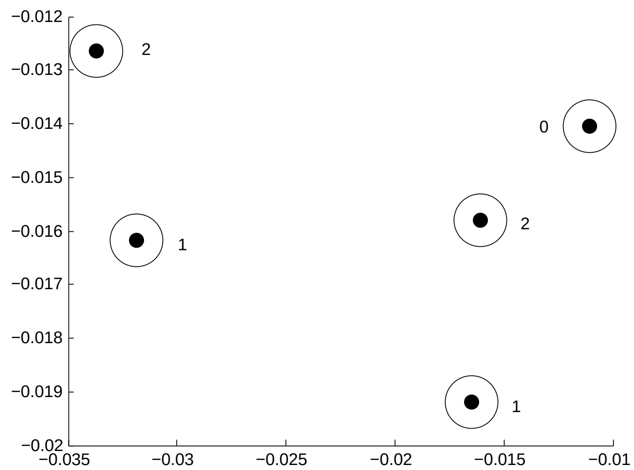


Figure 3.5: A frequency assignment example.

If we assign a frequency to the unassigned user, the cumulative interference will surpass the acceptable interference threshold (the difference becomes negative) as shown in the Table 3.2 with Frequency Set 2 and 3. These allocations are not allowed.

Result of Beam Moving algorithm is shown in Figure 3.6. Frequency 1 was tested at the unassigned users. In order to reduce interference incurred from this assignment, beams of the two interferers (those already assigned with Frequency 1) and the unassigned user's beam itself are moved. The move allows us to confirm Frequency 1 assignment. Note that the algorithm searches for the minimum move distance from the center positions.

Table 3.1: α and δ of the users in the given example.

i	$\alpha_i \times 10^{19}$	$\delta_{ij} \times 10^{19}$				
1	9.10	0	1.27	115.86	12.29	0.04
2	8.08	1.14	0	1.07	0.63	86.58
3	9.31	118.30	1.21	0	56.73	0
4	9.64	12.93	0.73	58.47	0	0.67
5	8.05	0.03	86.29	0	0.57	0

Table 3.2: Cumulative interference constraints of the users in different frequency sets.

i	Freq set 1	Constraints *	Freq set 2	Constraints *	Freq set 3	Constraints *
1	1	7.83	1	-4.46	1	7.83
2	1	6.94	1	6.31	2	6.94
3	2	9.31	2	9.31	2	-47.42
4	**0**	-	**1**	-4.03	**2**	-49.50
5	2	8.04	2	8.04	2	7.47

* $(\alpha_i - \sum_{j \in Interf} \delta_{ij})$

4.2 Closed-loop implementation

The ILP solver or the greedy algorithm would have more possibility to find the optimal solution or provide a better feasible solution if an initial feasible solution is given. Consider an iteration as a combination of ILP - Beam Moving algorithm or Greedy algorithm - Beam Moving algorithm. We propose the closed-loop implementation in that, in the next iteration of ILP or greedy algorithm, the frequency assignment result from Beam Moving algorithm is used as their initial solution. The moved beam positions are used for updating the α_i and δ_{ij} values. This method is actually a hill-climbing heuristic for solving the integrated frequency assignment/beam positioning mixed-integer non linear problem.

The ILP starts with the new initial solution, continues to improve the solution, and by the given CPU time, outputs the best found solution.

We implemented two variations of greedy algorithm. The first variation (Greedy 1) considers both the frequency assignment result and the updated α_i and δ_{ij} values and works further on the unassigned users. The second

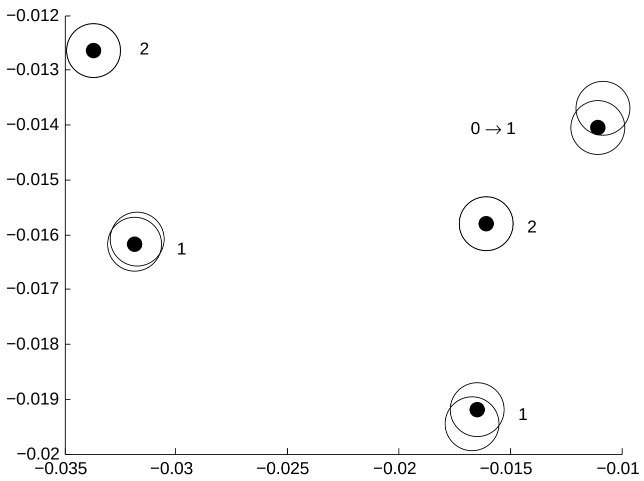


Figure 3.6: An example on beam moving with an additional frequency assignment.

variation (Greedy 2) only considers the updated α_i and δ_{ij} values and restarts the frequency assignment from scratch.

5 Computational experiments

The ILP formulations are solved using IBM/ILOG CPLEX [79]. The greedy algorithm is coded in C++. We tested the proposed algorithms with $F = 8$; increasing stepwise the number of users by 20 from 20 to 200 users with 100 instances each. The user positions are randomly generated and uniformly distributed over the service area defined by a set of geographic coordinates $u = [-0.043980, 0.048520]$ and $v = [-0.021152, 0.012348]$. The results were obtained on a 2.7GHz Intel Core i5 machine with 4GB RAM. The CPU times for the ILP resolutions were limited to 60s, 120s, and 180s after which the best integer solution was obtained. The CPU times for the greedy algorithm were negligible. The Beam Moving algorithm was performed with the maximum of 40 iterations for each unassigned users with no limitation on the calculation time.

The Beam Moving algorithm is coded in Matlab [119]. The function *fmincon* with active-set algorithm is used for computing the minimum move distance according to the given non-linear constraints.

The proposed models are denoted as follows:

- Model 1: fixed-beam binary interference

- Model 2: fixed-beam varying-frequency
- Model 3: SDMA-beam varying-frequency

We first present a comparison of the greedy algorithms from both Model 2 and Model 3. Table 3.3 reports the average number of accepted users over 1,000 instances. The results of the greedy algorithms are very close. It is difficult to give better results than the simple lexicographic rules which is consistent with the results previously obtained in [78]. The algorithm that uses Hybrid and Most used rules gives the best result. As of this, we use it for further performance comparison with the results from ILP and Beam Moving.

Table 3.3: Average number of accepted users over 1,000 instances.

	Fixed-beam	SDMA-beam
Lexicographic (user + frequency)	64.20	93.83
Lexicographic (user) + Most used (frequency)	64.28	93.84
Hybrid (user) + Most used (frequency)	65.38	94.19

Figure 3.7 presents performance comparison in term of the average number of accepted (served) users. Results of Model 1 are saturated at close to 80 users which is the maximum number of users that the model can support. Model 2 Greedy gives slightly better results than that of Model 1. A lot of improvement can be seen in Model 3 Greedy compared to both Model 2 Greedy and Model 2 ILP. This confirms the benefit of using SDMA since even a simple greedy algorithm can take advantage of the offered flexibility to assign more users than with the fixed-beam technology. There is indeed a significant improvement from Model 3 greedy solutions compared to the Model 2 solutions that were proved optimal by Model 2 ILP.

Model 3 ILP gives the best results; nonetheless, the performance gap between it and Model 3 Greedy is small at up to 120 users. The gap increases at 140-180 users then at 200 users the gap decreases again. The gap for Model 2 pair follows the same manner. The degradation indicates that the ILP needs longer time to solve the problem. The difference between Model 2 ILP and Model 3 ILP is the prohibition of using the same frequency inside the same beam. This makes the Model 2 ILP a difficult problem to solve than Model 3 ILP. This can also be seen from the number of optima found from these models in Table 3.4.

In order to determine the performance of the ILP, we increased the solving time to 120s and 180s. We can gain a few more optima at 120 and 140 users but none afterwards, see Table 3.5.

Chapter 3

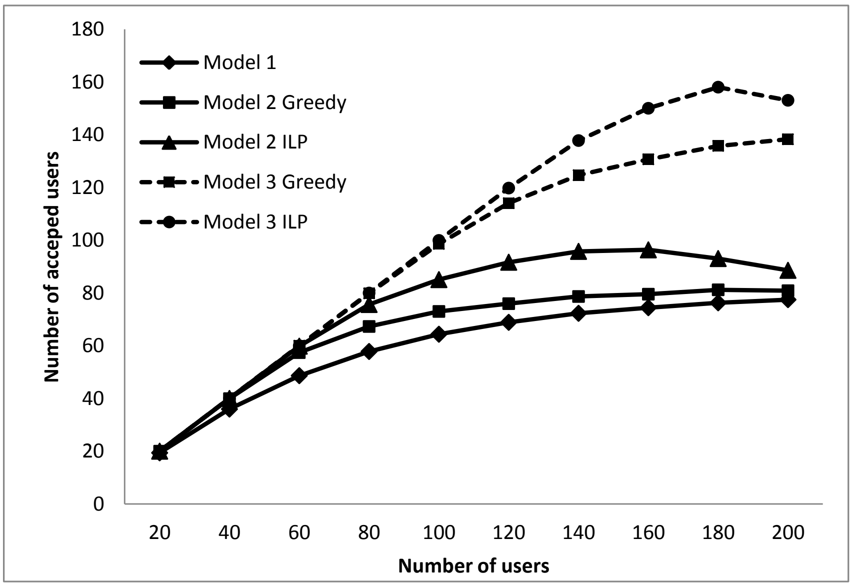


Figure 3.7: Average number of accepted users.

Table 3.4: Number of optima provided by ILP Model 2 (fixed-beam) and Model 3 (SDMA-beam).

Number of users	20	40	60	80	100	120	140	160	180	200
Fixed-beam Model	100	100	95	11	0	0	0	0	0	0
SDMA-beam Model	100	100	100	100	100	97	54	0	0	0

Table 3.6 presents lower bounds and upper bounds for ILP180s. These large gaps signify that the ILP formulation yields poor relaxations.

To proceed to the subsequent improvement, firstly, we tested 32 configurations of *k*-*MAXINEG*-*UTVAR* for the Beam Moving algorithm over 20 instances of 200 users. Test results are provided in Figure 3.8. It can be seen that increasing any of k (the number of moved beams: from 3 to 10) or *MAXINEG* (the minimum margin to the required signal to noise ratio: from 1 to 2) or enabling *UTVAR* (exclude (0) or include (1) the unassigned user's beam to the move) yields higher number of reassigned users, at an expense of longer calculation time. Both configuration 7-2-0 ($k = 7$, $MAXINEG = 2$, $UTVAR = 0$) and 6-2-1 provide good performances with acceptable calculation times. We pick configuration 7-2-0 for improving the results from the ILP and greedy

Table 3.5: Number of optima provided by ILPs.

Number of users	20	40	60	80	100	120	140	160	180	200
ILP60s	100	100	100	100	100	97	54	0	0	0
ILP120s	100	100	100	100	100	98	61	0	0	0
ILP180s	100	100	100	100	100	100	67	0	0	0

Table 3.6: Average upper and lower bounds for ILP180s.

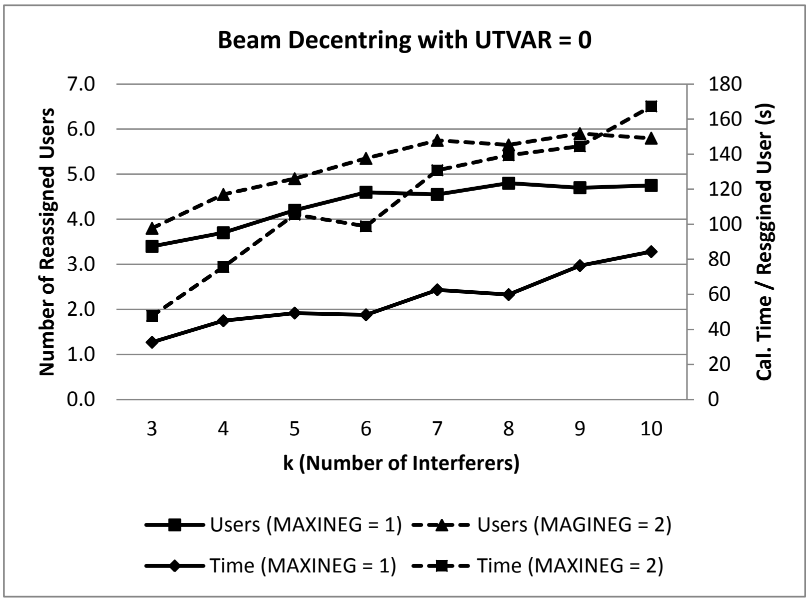
n	LB	UB	$\%(UB-LB)/UB$		
			min.	avg.	max.
120	119.79	119.81	0.00	0.02	1.67
140	138.17	139.18	0.00	0.71	3.76
160	151.07	158.21	1.25	4.46	7.50
180	160.69	177.19	5.06	9.25	13.22
200	165.22	194.36	9.33	14.90	23.59

algorithm.

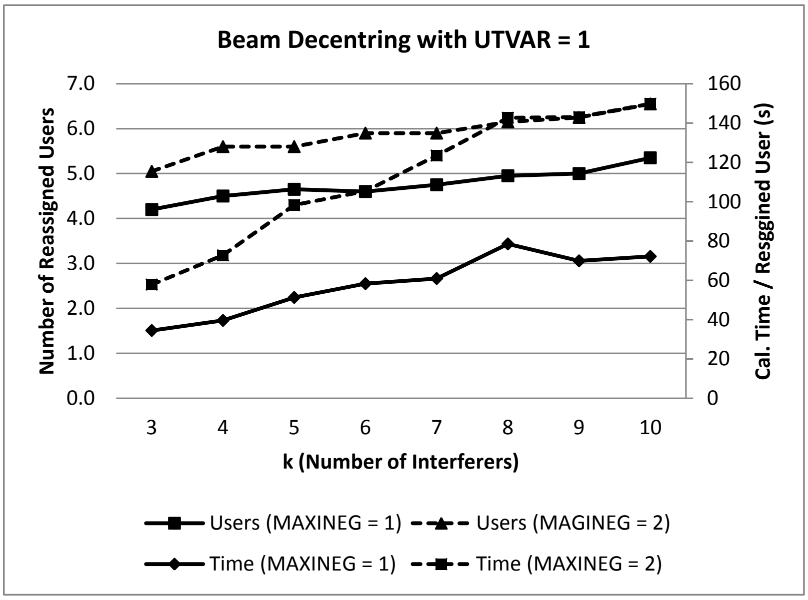
Figure 3.9 displays, for each algorithm and number of users, the average number of accepted users in the computed frequency assignment plans. The greedy algorithm performs as good as the other two ILPs at up to 120 users (ILP can solve to optima for all or almost all of 100 instances up to this point). For 140-200 users, the performance gap becomes larger as the number of user increases. Performance degradation is found in ILP60s at 200 user instances, contrast to that of ILP180s. This signifies that, though not reaching the optima, the ILP needs more time for a larger instance to provide a better results.

Beam Moving gives performance improvement for both greedy algorithm and ILP. Significant improvements can be seen in the greedy algorithm case. It could provide comparable results at 200 users compared to ILP60s. Nonetheless, the algorithm's calculation time is high, see Table 3.7 and 3.8.

The results for closed-loop simulations are shown in Table 3.9. Greedy 1 continuously improves the solutions over the iterations and approaches saturation after Iteration 3. Degraded performance is found for Greedy 2 in ILP Iteration 2 and 3. These are caused by restarting frequency assignment from scratch. For both ILPs, small improvement can be seen in the second iteration but no improvement in the third. ILPs converge to the saturation faster than Greedy algorithms.



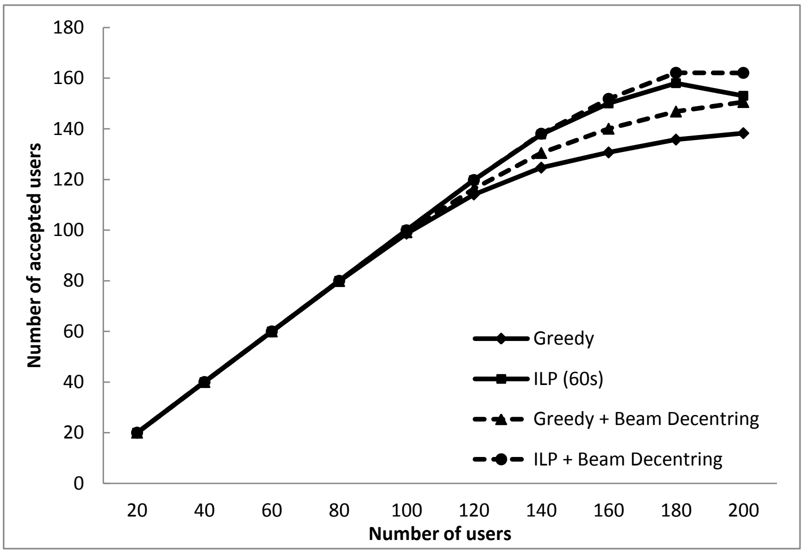
(a)



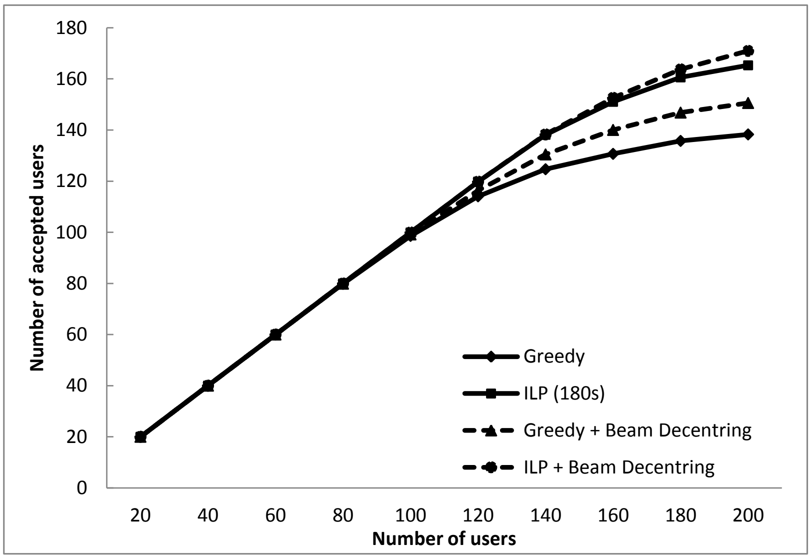
(b)

Figure 3.8: Average number of reassigned users and calculation time per reassigned user for different beam moving configurations over 20 instances of 200 users with, (a) UTVAR=0 and (b) UTVAR=1.

Chapter 3



(a)



(b)

Figure 3.9: Average number of accepted users before and after beam moving for Greedy algorithm and (a) ILP 60s or and (b) ILP 180s.

Table 3.7: Average calculation time (s) performed by Beam Moving algorithm.

Number of users	80	100	120	140	160	180	200
Greedy	9.19	22.85	67.60	241.67	570.69	1017.28	1542.53
ILP60s	-	-	13.57	29.65	125.26	365.21	1032.01
ILP180s	-	-	-	28.40	114.91	272.85	622.00

Table 3.8: Average calculation time (s) per reassigned users performed by Beam Moving algorithm.

Number of users	80	100	120	140	160	180	200
Greedy	9.19	12.45	31.91	44.90	62.26	95.75	129.68
ILP60s	-	-	13.57	26.84	68.70	108.54	119.76
ILP180s	-	-	-	28.40	67.66	104.86	125.12

6 Conclusion

The baseline model with binary interference is simple and clearly shows the relation between the frequency assignment problem and graph coloring. It serves as a starting point for developing the subsequent models and solving methods which are ILP formulations, greedy algorithms and non-linear continuous algorithms. All of these consider cumulative interference which makes the calculation a lot more complicated but more realistic.

The greedy algorithm, though simple, but is very fast and efficient enough to provide comparable results to ILP up to a certain number of users. ILP can give optimal results but require long calculation time. Nonetheless, ILP could be improved further using a technique such as column generation.

By utilising SDMA, the Beam Moving algorithm offers performance improvement for both ILP and greedy algorithm; the latter gains significant improvement. Closed-loop implementation provides further improvement yet marginal and requires long CPU time. To improve these results, a fast heuristic to solve the continuous optimization problem could be designed. Furthermore, an integrated approach where frequency allocation and beam position are determined simultaneously and not sequentially, could be proposed. This yields highly complex mixed non-linear integer programming formulations.

Table 3.9: Average percentage of accepted users over 100 instances of 200 users.

	Iteration 1		Iteration 2		Iteration 3	
	ILP	BD *	ILP	BD	ILP	BD
Greedy 1	69.15	75.29	76.05	76.05	76.20	76.20
Greedy 2	69.15	75.29	70.27	71.71	70.94	72.37
ILP 60s	76.53	81.05	81.58	81.84	81.84	-
ILP 180s	82.66	85.49	85.53	85.53	85.53	-
No. ** (Greedy 1)	-	100	73	24	24	1
No. ** (Greedy 2)	-	100	7	93	19	93
No. ** (60s)	-	100	14	13	0	-
No. ** (180s)	-	100	4	3	0	-

* (Beam moving), ** (Number of improved solutions)

Multiple carrier models

4

1 Introduction

By means of multiple carrier models, the frequency assignment problems are not limited to unity user demand. User can request for one or more frequency and, in case of multiple frequencies, the frequencies should be assigned in blocks of $f_n, f_{n+1}, \ldots, f_{n+d_i-1}$ where d_i is the frequency demand for user i. Interference will be treated first as binary and then cumulative. Unlike the single carrier models that the interference overlapping ratio is 1 (one frequency over same period of time), here, the interference overlapping ratio is between 0 and 1 inclusive. The objective remains the same that is to serve as many users as possible. Nonetheless, in some parts of this chapter, another objective is included which is to provide complete assignment to all users using the lowest number of frequencies.

Since utilising SDMA to center the beams over the users gives superior performance than the fixed beam cases, we will consider only SMDA-based models.

We will begin with binary interference models in Section 2. First we underline the similarities between the multiple frequency assignment problem and a class of scheduling problem and then as an interval graph coloring problem. At the end of the section, we also propose an ILP formulation for the problem. In Section 3, the scheduling model and the ILP model are extended further to cope with the cumulative interference. We also propose other ILP variations by treating the frequency index and overlapping area differently.

All the models proposed so far consider frequency assignment within the same time period. In Section 4, we attempt to include assignment in time. User demand then consists of both frequency and time. We propose a 2-dimensional (2D) ILP model for this problem.

The models are tested with 1,000 randomly generated instances: 100 instances of $20, 40, ..., 200$ users, which are the same set of instances used in Chapter 3. Test results are shown and compared in the Computational experiments section. Conclusions are provided at the end of the chapter.

2 Multiple carrier models with binary interference

In this section we presents three models for the multiple carrier frequency assignment problem with binary interference constraints. Since frequency assignment is closely related to graph coloring; we can model the problems in several ways. Here, we provide models based on scheduling, interval coloring, and ILP.

2.1 Binary interference

In the previous chapter, we consider binary interference between the satellite beams which are placed in a fixed-pattern manner. This binary interference is no longer valid for the SDMA-beam case where beams are centered at users.

Instead of creating new binary interference matrices, we construct them based on our existing test instances (100 instances of each of $20, 40, ...100$ users). Binary interference between user i and j denoted by Δ_{ij} is generated from interference coefficients δ_{ij} according to the following definition:

$$\Delta_{ij} = \begin{cases} 1, & \text{if} \quad \delta_{ij} \geq LF \cdot \overline{\delta_{ij}} \vee \delta_{ji} \geq LF \cdot \overline{\delta_{ij}}, \\ 0, & \text{otherwise.} \end{cases} \tag{4.1}$$

$LF \in [0, 1]$ corresponds to a loading factor and $\overline{\delta_{ij}} = \frac{\sum \delta ij}{\sum k_{ij}} \quad | \quad k_{ij} = 1$ if $\delta_{ij} \geq 0$ corresponds to the mean of all non-zero δ_{ij} values.

LF determines how load the binary interference matrix is. If it is set to 1, binary interference between user i, j presents if either one of their interference coefficients is not less than the mean $\overline{\delta_{ij}}$. The lower the LF the more load the binary interference matrix is.

Note that $\delta_{ij} \neq \delta_{ji}$ since we consider the perceived interference at the satellite while distances between the satellite and the two users are different, but, for binary interference, it is necessary to set $\Delta_{ij} = \Delta_{ji}$. According to Equation (4.1), they are set to 1 if any of the $(\delta_{ij}, \delta_{ji})$ pair is greater than the $LF \cdot \overline{\delta_{ij}}$ value.

Nonetheless, to ensure that the binary interference is a good estimation of the actual interference, we need to verify that each of the feasible solutions based on this binary interference constraints does not violate the actual cumulative interference constraints. In case of violation, LF should be reduced in order to induce more binary interference relations.

2.2 Scheduling

According to Pinedo [133], scheduling is a decision making process that deals with the allocation of resources to tasks over given time periods while the resources and tasks can take many different forms so as to the objectives.

Scheduling is a large and broad subject and mostly concern industrial applications. In a paper proposed by de Werra [39], a link between chromatic scheduling, graph coloring, and frequency assignment is established. Nonetheless, to our knowledge, research considering scheduling and frequency assignment problem explicitly is rare. Yet, we find that it is interesting to model our frequency assignment problem as a scheduling problem.

If we treat a user as a task or an operation and user demand (in number of frequencies) as the task's processing time. Maximizing the number of assigned users in frequency assignment problem could be viewed as maximizing the number of scheduled tasks or operations having their common deadlines equal to the number of available frequencies in scheduling problem. Binary interference constraints between each couple of users can be treated as non-overlapping constraints between each couple of tasks.

If instead we consider a common parameter in scheduling problem, the makespan $(C_{\mathtt{max}})$, we could link it to frequency assignment problem in that minimizing the total makespan of the schedule gives us the minimum number of frequencies used for a complete frequency assignment (all users are assigned).

In fact, our multiple carrier frequency assignment problem with binary interference could be viewed as a disjunctive scheduling problem without precedence constraints.

A disjunctive scheduling is also commonly defined as a set of uninterrupted tasks with fixed durations that have to be performed on a set of machines while the machine can handle one task at a time [27]. The goal in this case is to order the tasks on the different machines according to the objective such as minimizing the total makespan of the schedule. Binary interference in our frequency assignment problem can be treated as disjunctive constraints in that two interfering users refer to two non-overlapping tasks. In non-interference case, tasks can be overlapped. As we explain in the next section, this overlapping can be viewed as having tasks processed on different machines.

2.2.1 Disjunctive graph and clique

The problem can be represented by a disjunctive graph $G = (V, E)$ in which the vertices represent the users and an edge between two vertices represent their binary interference pair. Each edge of this disjunctive graph is treated as a disjunctive or non-overlapping constraint in scheduling.

We can model our scheduling by directly including each of these constraint pairs. Consider n as a number of vertices, there are at most $n(n-1)/2$ disjunctive constraints. Alternatively, we can choose to deal with a group of constraints using maximal cliques concept.

A clique in a graph is defined as a subset of the vertices such that every two vertices in the subset are connected by an edge. A maximal clique is a clique that cannot be extended by including one more adjacent vertex, that is, a clique which does not exist exclusively within the vertex set of a larger clique.

A maximal clique in this case consists of a group of users that are all interfered, i.e., in terms of scheduling, a machine shared by a group of tasks. By dealing with maximal cliques in the scheduling is similar to dealing with groups of constraints simultaneously which may lead to more efficient approaches. Among others (see e.g. [15]), the well known edge finding technique is able to detect implied precedence relations given the time windows of a set of tasks sharing the same machine.

The problem is that there may be an exponential number of maximal cliques given an arbitrary binary interference graph.

Below is an example of a disjunctive graph with $\{1,2\}$, $\{2,3\}$, $\{2,4\}$, $\{3,5\}$ and $\{4,5,6\}$ as its maximal cliques.

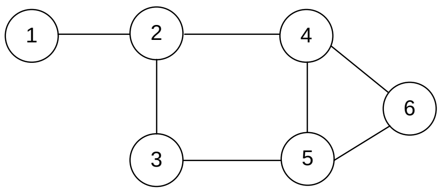


Figure 4.1: A disjunctive graph.

A clique can be viewed as a machine in disjunctive scheduling. In this example, there are 5 machines $m_1, ..., m_5$. We can represent these machines by a binary matrix with rows for machines $m_1, ..., m_5$ and columns for tasks $1, ..., 6$ as

$$\begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix}.$$

Tasks associated to the same machine cannot be overlapped. According to this disjunctive graph, if we model the disjunctive constraints directly, 7 pairs of constraints are needed. If we use maximal clique concept, we can model the constraints based on 5 cliques.

2.2.2 Disjunctive scheduling models

By considering a user as a task i and its frequency demand d_i as a processing time, we can model the equivalent scheduling problem as following:

For a minimum makespan problem, let the variable f_i denotes the starting time of the task i and the problem can be modelled as

$$\min \quad C_{\mathtt{max}} \tag{4.2}$$

subject to

$$C_{\mathtt{max}} - f_i \geq d_i, \quad \forall i, \tag{4.3}$$

$$f_i \geq 0, \quad \forall i, \tag{4.4}$$

$$f_i - f_j \geq d_j \vee f_j - f_i \geq d_i, \quad \forall i, j \in I. \tag{4.5}$$

Set I contains all the pairs of tasks that are independent (non-overlapping) to one another. These pairs correspond to the pairs of vertices of the disjunctive graph mentioned above.

In case of cliques, set I is replaced with a number of sets which each contains tasks that are all independent. Constraint (4.5) becomes

$$f_i - f_j \geq d_j \vee f_j - f_i \geq d_i \quad \forall i, j \in C, \forall C \in \mathcal{C}. \tag{4.6}$$

where $\mathcal{C}$ is the set of all considered cliques.

Equations (4.5) and (4.6) are equivalent provided that $\mathcal{C}$ is a set of cliques of the disjunctive graph such that each disjunctive constraint is included in a clique of $\mathcal{C}$. This holds true in particular if $\mathcal{C}$ is the set of maximal cliques. The minimum makespan problem described above is equivalent to minimising the number of frequencies to serve all users with no interference.

Maximizing the number of scheduled tasks with a common due date BW is equivalent to minimizing the number of tardy tasks with a common due date BW, which is

$$\min \sum_{i=1}^{n} U_i \tag{4.7}$$

$$U_i = \begin{cases} 1, & \text{if} \quad f_i + d_i \geq BW, \\ 0, & \text{otherwise,} \end{cases} \tag{4.8}$$

subject to (4.5) and (4.6).

This scheduling problem is equivalent to the FAP with maximization of the number of served users with no interference. Remark that when the interference graph is complete, there is a single maximal clique, i.e. a single machine. The corresponding scheduling problem is denoted $1|D_i = D|\sum U_i$ and is simply solved by sorting the tasks with the SPT rule (shortest processing time) and by scheduling the maximum number of tasks before D in that order. For the general case, the problem is NP-hard as it includes for instance the job-shop problem. [1]

These models are solved by CP Optimizer in [79]. Results are shown in the Computational experiments sections.

2.3 Interval graph coloring

Graph coloring is one of the most studied NP-hard combinatorial optimization problems [59]. There are a large number of variations and generalizations of graph coloring to cope with different applications. This includes interval coloring which is introduced by Punter [134].

Punter [134] introduced the concept of vertex-composite graphs coloring to model a school timetabling problem with lectures of different lengths. According to Golumbic [64], this concept is a version of an interval coloring of a weighted graph. This idea is further elaborated in Clementson and Elphick [32] and de Werra and Hertz [40] while an exact coloring algorithm is provided by Čangalović and Schreuder [155].

In [22], Bouchard et al. generalizes interval coloring to determine a bandwidth coloring with minimum difference between the largest and the smallest colors used. This is, in fact, a minimum span problem.

Our frequency assignment problem can be modelled by a weighted graph $G = (V, E, c)$ with the vertices V representing users, the edges E representing binary interference, and the weights c representing the frequency demand. A graph G is considered having an interval k-coloring if $c_i \in c$ distinct and consecutive integers from the set $\{1, 2, \dots, k\}$ are assigned to each $v_i \in V$ in such as way that no two adjacent vertices have a color in common. The interval chromatic number of the graph G denoted by $\chi(G)$ is the smallest number k such that G has an interval k-coloring; the corresponding coloring is optimal.

Finding the interval chromatic number of the graph is similar to solving a frequency assignment problem by assigning frequency to all of the users using the lowest number of frequencies.

[1] The notation D_i is used instead of d_i which is generally encountered in scheduling since in this thesis d_i stands for the user demand.

Bouchard et al. [21] discussed about k-interval graph coloring problem with a minimum number of conflicting edges whereas an edge is said to be conflicting if the two corresponding vertices have common colors. Assigning as many users a possible based on a number of available frequencies could also be dealt by k-interval graph coloring; nonetheless, in this case, not all the vertices are colored. Instead, the problem is to maximize number of colored vertices.

2.4 Integer linear programming

For binary interference case, overlapping between two users i, j is allowed if they are not connected by an edge in the interference graph. Define a set D which contains pairs of i, j whose edges exist in this graph. Let f_i and f_j the starting frequency of user i, j; d_i and d_j their corresponding frequency demands, BW the available bandwidth and

$$x_i = \begin{cases} 1 & \text{if user } i \text{ is assigned,} \\ 0 & \text{otherwise.} \end{cases}$$

The ILP formulation is given as

$$\max \sum_{i \in U} x_i \tag{4.9}$$

$$f_j \geq f_i + d_i - BW(1 - y_{ij}) - BW(2 - x_i - x_j) \qquad \forall i, j \in D \tag{4.10}$$

$$y_{ij} + y_{ji} \leq 1 \qquad \forall i, j \in U, d_i \geq d_j \tag{4.11}$$

$$y_{ij} \in \{0, 1\} \qquad \forall i, j \in U \tag{4.12}$$

$$x_i \in \{0, 1\} \qquad \forall i \in U \tag{4.13}$$

$$0 \leq f_i \leq BW - 1 \qquad \forall i \in U \tag{4.14}$$

We define binary variable y_{ij} and y_{ji} such that $y_{ij} = 1 \implies f_j \geq f_i + d_i$, constraints (4.10) impose non-overlapping between $i, j \in D$. Here f_i variables do not need be integer as x_i, y_{ij} and d_i being integer ensure that integer values can be derived from any continuous f_i value satisfying constraints (4.10).

3 Multiple carrier models with cumulative interference

To cope with cumulative interference, models presented in the previous sections can be extended. Binary interference constraints should be replaced with cumulative interference constraints. Overlapping is allowed as long as the interference does not exceed the limit value.

3.1 Scheduling

By using cumulative constraints, we can no longer use concept of maximal clique. Nonetheless, the problem can still be treated as a scheduling problem but how the constraints are handled is different. Tasks can be overlapped and the overlap length should be taken into consideration in the constraints.

Let o_{ij} the overlap length between task i and j. The minimum makespan problem can be modelled as

$$\min \quad C_{\texttt{max}} \tag{4.15}$$

subject to

$$C_{\texttt{max}} - f_i \geq d_i, \quad \forall i, \tag{4.16}$$

$$f_i \geq 0, \quad \forall i, \tag{4.17}$$

$$o_{ij} = \begin{cases} d_i, & \text{if} \quad f_i \geq f_j \wedge f_j + d_j \geq f_i + d_i, \\ d_j, & \text{if} \quad f_j \geq f_i \wedge f_i + d_i \geq f_j + d_j, \\ f_i + d_i - f_j, & \text{if} \quad f_j \geq f_i \wedge f_j + d_j \geq f_i + d_i, \\ f_j + d_j - f_i, & \text{if} \quad f_i \geq f_j \wedge f_i + d_i \geq f_j + d_j, \\ 0, & \text{otherwise}, \end{cases} \tag{4.18}$$

$$\sum_{j \neq i} o_{ij} \delta_{ij} \leq d_i \alpha_i, \tag{4.19}$$

while the variable f_i denotes the starting time of the task i and d_i denotes task i's processing time. Equation (4.19) provides cumulative interference constraints.

Minimizing the number of tardy tasks with a common due date BW can be modelled as

$$\min \sum_{i=1}^{n} U_i \tag{4.20}$$

$$U_i = \begin{cases} 1, & \text{if} \quad f_i + d_i \geq BW, \\ 0, & \text{otherwise}, \end{cases} \tag{4.21}$$

Chapter 4

subject to

$$o_{ij} = \begin{cases} d_i, & \text{if} \quad f_i \geq f_j \wedge f_j + d_j \geq f_i + d_i, \\ d_j, & \text{if} \quad f_j \geq f_i \wedge f_i + d_i \geq f_j + d_j, \\ f_i + d_i - f_j, & \text{if} \quad f_j \geq f_i \wedge f_j + d_j \geq f_i + d_i, \\ f_j + d_j - f_i, & \text{if} \quad f_i \geq f_j \wedge f_i + d_i \geq f_j + d_j, \\ 0, & \text{otherwise}, \end{cases} \tag{4.22}$$

$$\sum_{j \neq i} o_{ij} \delta_{ij} \leq d_i \alpha_i, \tag{4.23}$$

while the variable y_i denotes the completion time of task i.

In CP Optimizer [79], instead of modelling the overlapping o_{ij} directly as provided above, the o_{ij} can be handled easily by the function *IloOverlapLength*. The cumulative interference constraints can be rewritten as

$$\sum_{j \neq i} \text{IloOverlapLength}(i, j) \delta_{ij} \leq d_i \alpha_i. \tag{4.24}$$

3.2 Integer linear programming

We can extend the ILP model given in the binary interference section to cope with cumulative interference. To do this, we have to compute the overlapping o_{ij}. In fact, the contiguous frequency and overlapping area can be treated in several ways. We will start with the extended ILP model as below. Then we will provide other alternatives in the subsequent subsections.

3.2.1 Direct frequency model

Let f_i and f_j the starting frequency of user i, j; d_i and d_j their corresponding frequency demands, BW the available bandwidth and

$$x_i = \begin{cases} 1 & \text{if user } i \text{ is assigned}, \\ 0 & \text{otherwise}. \end{cases}$$

The ILP formulation is given as

$$\max \sum_{i \in U} x_i \tag{4.25}$$

$$f_j \geq f_i - BW y_{ij} \qquad \forall i, j \in U, d_i \geq d_j \tag{4.26}$$

$$f_i + d_i \geq f_j + d_j - BW y_{ji} \qquad \forall i, j \in U, d_i \geq d_j \tag{4.27}$$

$$y_{ij} + y_{ji} \leq 1 \qquad \forall i, j \in U, d_i \geq d_j \tag{4.28}$$

$$o_{ij} \geq f_j + d_j - f_i - BW(1 - y_{ij}) - BW(1 - x_j) \qquad \forall i, j \in U, d_i \geq d_j \tag{4.29}$$

$$o_{ij} \geq f_i + d_i - f_j - BW(1 - y_{ji}) - BW(1 - x_j) \qquad \forall i, j \in U, d_i \geq d_j \tag{4.30}$$

$$o_{ij} \geq d_j - BW(y_{ij} + y_{ji}) - BW(1 - x_j) \qquad \forall i, j \in U, d_i \geq d_j \tag{4.31}$$

$$o_{ij} = o_{ji} \qquad \forall i, j \in U, d_i \geq d_j \tag{4.32}$$

$$\sum_{j \neq i} o_{ij} \delta_{ij} \leq \alpha_i d_i \qquad \forall i \in U \tag{4.33}$$

$$y_{ij} \in \{0, 1\} \qquad \forall i, j \in U \tag{4.34}$$

$$x_i \in \{0, 1\} \qquad \forall i \in U \tag{4.35}$$

$$f_i \in \{0, \dots, BW - 1\} \qquad \forall i \in U \tag{4.36}$$

$$o_{ij} \geq 0 \qquad \forall i, j \in U \tag{4.37}$$

We define binary variable y_{ij} and y_{ji} for $d_i \geq d_j$ such that $y_{ij} = 0 \implies f_j \geq f_i$ (4.26) and $y_{ji} = 0 \implies f_i + d_i \geq f_j + d_j$ (4.27). With, additional constraints (4.28), the overlap o_{ij} can be one of the following cases:

- $y_{ij} = 1$, $y_{ji} = 0$ and $o_{ij} \geq f_i + d_i - f_j$,
- or $y_{ij} = 0$, $y_{ji} = 1$ and $o_{ij} \geq f_j + d_j - f_i$,
- or $y_{ij} = 0$, $y_{ji} = 0$ and $o_{ij} \geq d_j$,

which are stated by constraints (4.29-4.31). Nonetheless, contrarily to the binary interference case, variables f_i have to be integer, otherwise the continuous values can be used to decrease the overlapping.

3.2.2 Frequency-indexed model (INDEXED1)

Instead of modelling the f_i and x_i separately, we index the frequency statically from 1 to BW and use the variable x_{if} which is defined as

$$x_{if} = \begin{cases} 1 & \text{if user } i \text{ starts at frequency } f, \\ 0 & \text{otherwise.} \end{cases}$$

Then the ILP formation can be written as

Chapter 4

$$\max \sum_{i \in U} \sum_{f=1}^{BW} x_{if} \tag{4.38}$$

$$\sum_{f=1}^{BW} x_{if} \leq 1 \qquad \forall i \in U \tag{4.39}$$

$$o_{ij} - d_j x_{if} + d_j(1 - \sum_{f'=1}^{BW} x_{jf'}) + \sum_{f'=0}^{f-1} (f - f') x_{jf'} + \sum_{f'=f+d_i+1-d_j}^{BW-d_j} (f' + d_j - f - d_i) x_{jf'} \geq 0 \qquad \forall i, j \in U, d_i \geq d_j, \forall f \tag{4.40}$$

$$o_{ij} = o_{ji} \qquad \forall i, j \in U, d_i \geq d_j \tag{4.41}$$

$$\sum_{j \neq i} o_{ij} \delta_{ij} \leq \alpha_i d_i \qquad \forall i \in U \tag{4.42}$$

$$x_{if} \in \{0, 1\} \qquad \forall i \in U, \forall f \tag{4.43}$$

$$o_{ij} \geq 0 \qquad \forall i, j \in U \tag{4.44}$$

We call this a frequency-indexed model or "INDEXED1". The overlapping constraints (4.40) sets a lower bound on the overlapping o_{ij} between i and j when the demand of i is not less than the demand of j ($d_i \geq d_j$). In this case, for each frequency f, if i starts at f ($x_{if} = 1$) and if j is assigned to a non zero set of frequencies ($1 - \sum_{f'=1}^{BW} x_{jf'} = 0$), the overlapping is larger than or equal to d_j minus the frequency units to be removed if j is not assigned inside the interval $[f, f + d_i]$. More precisely, if j starts at $f' \leq f - 1$, $o_{ij} \geq d_j - (f - f')$. If j starts after $(f + d_i) + 1 - d_j$, $o_{ij} \geq d_j - ((f' + d_j) - (f + d_i))$.

3.2.3 Frequency-indexed model (INDEXED2)

Overlapping constraints in the INDEXED1 model can be expressed by introducing continuous variables o_{ijf} which state whether i and j overlap at frequency f. In that case constraints (4.40) can be replaced by the following constraints:

$$o_{ijf} \geq \sum_{f'=\max(0, f-d_i+1)}^{\min(f, BW-d_i)} x_{if'} + \sum_{f'=\max(0, f-d_j+1)}^{\min(f, BW-d_j)} x_{jf'} - 1 \quad \forall i, j, f \tag{4.45}$$

$$o_{ij} = \sum_f o_{ijf} \quad \forall i, j \tag{4.46}$$

$$0 \leq o_{ijf} \leq 1 \quad \forall i, j, f \tag{4.47}$$

3.2.4 Frequency-indexed model (INDEXED3)

As an alternative, the o_{ij} in the INDEXED2 model can be further replaced by $o_{ijf_1f_2}$ which represents overlap when i and j start at f_1 and f_2 as:

$$o_{ij} \geq o_{ijf_1f_2}(x_{if_1} - x_{jf_2} - 1) \quad \forall i, j, f_1, f_2$$

3.2.5 On-off z_{if} model

In this model the frequency is indexed statically by $f \in \{0, \ldots, BW - 1\}$. Let $z_{if} \in \{0, 1\}$ and $z_{jf} \in \{0, 1\}$ where $i \in \{1, \ldots, n\}$, $j \in \{1, \ldots, n\}$ in that z_{if} (or z_{jf}) $= 1$ if user i (or j) uses frequency f. Define $w_{ijf} = z_{if} \cdot z_{jf}$ as an overlap when both z_{if} and $z_{jf} = 1$. The ILP can be given as:

$$\max \sum_{i \in U} x_i \tag{4.48}$$

$$w_{ijf} \geq z_{if} + z_{jf} - 1 \qquad \forall i, j, f \tag{4.49}$$

$$w_{ijf} \leq z_{if} \qquad \forall i, j, f \tag{4.50}$$

$$w_{ijf} \leq z_{jf} \qquad \forall i, j, f \tag{4.51}$$

$$0 \leq w_{ijf} \leq 1 \qquad \forall i, j, f \tag{4.52}$$

$$z_{ig} \leq z_{if} + 1 - z_{i,f-1} \qquad \forall i, \forall g > f \tag{4.53}$$

$$z_{ig} \leq z_{if} + 1 - z_{i,f+1} \qquad \forall i, \forall g < f \tag{4.54}$$

$$\sum_f z_{if} = d_i x_i \qquad \forall i, f \tag{4.55}$$

$$o_{ij} \geq \sum_f w_{ijf} \qquad \forall i, j \tag{4.56}$$

$$o_{ij} = o_{ji} \qquad \forall i, j \tag{4.57}$$

$$\sum_{j \neq i} o_{ij}\delta_{ij} \leq \alpha_i d_i \qquad \forall i, j \tag{4.58}$$

$$x_i \in \{0, 1\} \qquad \forall i \tag{4.59}$$

Constraints (4.49) to (4.52) are linear programmings representing the product $w_{ijf} = z_{if} \cdot z_{jf}$. In order to ensure that the allocation of each user is contiguous within the bandwidth BW we add contiguity constraints (4.53)-(4.54).

3.2.6 Column generation model

As an attempt to define models that can be used with column generation method, we propose a vertical model, in which we define a pattern as an

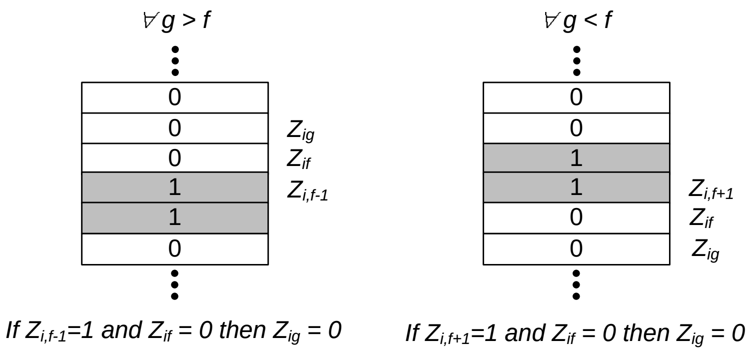


Figure 4.2: Contiguity constraints.

assignment of users-frequencies with no overlapping. Let p denotes a pattern in a set of patterns $\mathcal{P}$ and $y_p \in \{0, 1\}$ while $y_p = 1$ if the pattern p is in the solution and 0 otherwise.

Let $a_{ip} = 1$ if user i is in pattern p and 0 otherwise, $z_{ifp} = 1$ if user i uses frequency f in pattern p. By replacing z_{if} in the z_{if} model by $\sum_p z_{ifp} y_p$, the problem becomes

$$\max \sum_{i=1}^{n} \sum_{p \in \mathcal{P}} a_{ip} y_p \tag{4.60}$$

$$\sum_{p} a_{ip} y_p \leq 1 \qquad \forall i \tag{4.61}$$

$$w_{ijf} \geq \sum_{p} z_{ifp} y_p + \sum_{p} z_{jfp} y_p \qquad \forall i, j, f \tag{4.62}$$

$$0 \leq w_{ijf} \leq 1 \qquad \forall i, j, f \tag{4.63}$$

$$o_{ij} \geq \sum_{f} w_{ijf} \qquad \forall i, j \tag{4.64}$$

$$o_{ij} = o_{ji} \qquad \forall i, j \tag{4.65}$$

$$\sum_{j \neq i} o_{ij} \delta_{ij} \leq \alpha_i d_i \qquad \forall i, j \tag{4.66}$$

$$y_p \in \{0, 1\} \tag{4.67}$$

If we consider in the master only the constraints involving y_p variables, we have

$$\max \sum_i \sum_p a_{ip} y_p \tag{4.68}$$

$$\sum_p a_{ip} y_p \leq 1 \qquad \forall i \tag{4.69}$$

$$w_{ijf} \geq \sum_p z_{ifp} y_p + \sum_p z_{jfp} y_p - 1 \qquad \forall i, j, f \tag{4.70}$$

in the dual we consider only the constraints linked to y_p variables (we use λ_i for the dual variable of constraints (4.69) and γ_{ijf} for the dual variable of constraints (4.70).

This gives the following dual constraints:

$$\sum_i a_{ip} \lambda_i - \sum_i \sum_j \sum_f (z_{ifp} + z_{jfp}) \gamma_{ijf} \geq \sum_i a_{ip} \forall p \tag{4.71}$$

to find a violated dual constraint we have to find a column p such that

$$\sum_i a_{ip} \lambda_i - \sum_i \sum_j \sum_f (z_{ifp} + z_{jfp}) \gamma_{ijf} < \sum_i a_{ip} \tag{4.72}$$

or

$$\sum_i \sum_j \sum_f (z_{ifp} + z_{jfp}) \gamma_{ijf} + \sum_i (1 - \lambda_i) a_{ip} > 0 \tag{4.73}$$

As a pattern defines an overlapping schedule we cannot have z_{ifp} and z_{jfp} equal to 1 at the same time. Furthermore if $a_{ip} = 0$ (task is not in the pattern) we have also $z_{ifp} = 0$ for all f.

It follows that we have to find a subset of tasks to schedule before BW that maximizes

$$\sum_i f_i(C_i).(1 - U_i) \tag{4.74}$$

where

$$f_i(Ci) = \sum_{f=C_i-d_i}^{C_i-1} \sum_j \gamma_{ijf} - \lambda_i + 1 \tag{4.75}$$

if a column p is found such that this objective function is strictly positive it can be added to the master problem.

The subproblem is a kind of knapsack but the profit of getting an item i (scheduling a task i before BW) depends on the position of i.

4 Multiple carrier models with 2D (frequency and time) assignment

All the models proposed so far consider frequency assignment within the same time period. In this section, we include assignment in time. User demand then consists of both frequency h_i and time w_i. Both frequency and time are treated separately corresponding to the given demands. Then they are linked with the overlapping condition and interference constraints. According to this, we call the model the 2-dimensional (2D) ILP model.

4.1 Mathematical model

The problem can be represented mathematically as

$$\max \quad |\{i \in U | f_i \neq 0 \wedge t_i \neq 0\}| \tag{4.76}$$

subject to

$$f_i = \begin{cases} (q_i, q_i+1, \ldots, q_i + |f_i| - 1), q_i \in \{1, \ldots, (BW - h_i)\}, \\ \emptyset, \end{cases} \tag{4.77}$$

$$t_i = \begin{cases} (p_i, p_i+1, \ldots, p_i + |t_i| - 1), p_i \in \{1, \ldots, (\mathit{FrameDuration} - w_i\}, \\ \emptyset, \end{cases} \tag{4.78}$$

$$|f_i| = h_i \text{ and } |t_i| = w_i \text{ or } f_i = t_i = \emptyset \tag{4.79}$$

$$f_i = f_j = 0 \text{ or } f_i \cap f_j = 0 \text{ if } t_i \cap t_j \neq 0 \text{ or}$$

$$t_i \cap t_j = 0 \text{ if } f_i \cap f_j \neq 0 \quad \forall i, j, i \neq j,$$

$$f_i \in \{0, \ldots, \mathit{BW} - h_i\} \quad \forall i, j, \tag{4.80}$$

$$t_i \in \{0, \ldots, \mathit{FrameDuration} - w_i\} \quad \forall i, j, \tag{4.81}$$

$$\sum_{\substack{j \in U \setminus \{i\} \\ f_j \neq 0 \\ f_j \cap f_i \neq 0 \ \wedge \ t_j \cap t_i \neq 0}} o_{ij}\delta_{ij} \leq \alpha_i w_i h_i \quad \forall i \in U, \tag{4.82}$$

giving that

Chapter 4

- n the number of users,
- $U = \{1, \dots, n\}$ a set of users,
- *FrameDuration* the frame duration,
- *BW* the available bandwidth,
- w_i the user demand in time,
- h_i the user demand in frequency,
- α_i the acceptable interference threshold for user i,
- δ_{ij} the interference coefficient of user j towards user i.

4.2 Direct frequency-time indexed model for 2D

Let f_i and f_j the starting frequency of user i, j; h_i and h_j their corresponding frequency demands; w_i and w_j their corresponding time demands, BW the available bandwidth, *FrameDuration* the available frame duration and

$$x_i = \begin{cases} 1 & \text{if user } i \text{ is assigned,} \\ 0 & \text{otherwise,} \end{cases}$$

$$x_{it} = \begin{cases} 1 & \text{if user } i \text{ uses time } t, \\ 0 & \text{otherwise.} \end{cases}$$

The ILP formulation is given as

$$\max \sum_{i \in U} x_i \tag{4.83}$$

$$f_j \geq f_i - BW y_{ij} \qquad \forall i,j \in U, d_i \geq d_j \tag{4.84}$$

$$f_i + h_i \geq f_j + h_j - BW y_{ji} \qquad \forall i,j \in U, d_i \geq d_j \tag{4.85}$$

$$y_{ij} + y_{ji} \leq 1 \qquad \forall i,j \in U, d_i \geq d_j \tag{4.86}$$

$$o_{ij} \geq f_j + h_j - f_i - BW(1 - y_{ij}) - BW(1 - x_j) \qquad \forall i,j \in U, d_i \geq d_j \tag{4.87}$$

$$o_{ij} \geq f_i + h_i - f_j - BW(1 - y_{ji}) - BW(1 - x_j) \qquad \forall i,j \in U, d_i \geq d_j \tag{4.88}$$

$$o_{ij} \geq h_j - BW(y_{ij} + y_{ji}) - BW(1 - x_j) \qquad \forall i,j \in U, d_i \geq d_j \tag{4.89}$$

$$o_{ij} = o_{ji} \qquad \forall i,j \in U, d_i < d_j \tag{4.90}$$

$$o_{ij} \geq 0 \qquad \forall i,j \in U \tag{4.91}$$

$$x_i \geq x_{it} \qquad \forall i \in U, \forall t \tag{4.92}$$

$$x_i \leq \sum_t x_{it} \qquad \forall i \in U, \forall t \tag{4.93}$$

$$\sum_t x_{it} = w_i - w_i(1 - x_i) \qquad \forall i \tag{4.94}$$

$$x_{is} \leq x_{it} - x_{i,t-1} + 1 \qquad \forall i, s > t \tag{4.95}$$

$$x_{is} \leq x_{it} - x_{i,t+1} + 1 \qquad \forall i, s < t \tag{4.96}$$

$$o_{ijt} \geq o_{ij} - \min(h_i, h_j)(2 - x_{it} - x_{jt}) \qquad \forall i,j,t \tag{4.97}$$

$$Go_{ij} = \sum_t o_{ijt} \qquad \forall i,j \tag{4.98}$$

$$\sum_{j \neq i} Go_{ij} \delta_{ij} \leq \alpha_i w_i h_i \qquad \forall i \in U \tag{4.99}$$

$$y_{ij} \in \{0,1\} \qquad \forall i,j \in U \tag{4.100}$$

$$x_i \in \{0,1\} \qquad \forall i \in U \tag{4.101}$$

$$x_{it} \in \{0,1\} \qquad \forall i \in U \tag{4.102}$$

$$f_i \in \{0, \ldots, BW - 1\} \qquad \forall i \in U \tag{4.103}$$

$$t \in \{0, \ldots, \mathit{FrameDuration} - 1\} \tag{4.104}$$

Similar to 1D case, we define binary variable y_{ij} and y_{ji} for $d_i \geq d_j$ such that $y_{ij} = 0 \implies f_j \geq f_i$ (4.84) and $y_{ji} = 0 \implies f_i + d_i \geq f_j + d_j$ (4.85). With, additional constraints (4.86), the overlap in frequency o_{ij} can be one of the following cases:

- $y_{ij} = 1$, $y_{ji} = 0$ and $o_{ij} \geq f_i + d_i - f_j$,
- or $y_{ij} = 0$, $y_{ji} = 1$ and $o_{ij} \geq f_j + d_j - f_i$,

- or $y_{ij} = 0$, $y_{ji} = 0$ and $o_{ij} \geq d_j$,

which are stated by constraints (4.87-4.89).

Constraints (4.92) and (4.93) impose that the user i should also be assigned a starting time whose duration set by constraints (4.94). Constraints (4.95) and (4.96) ensure contiguous time in the time resource assignment. The overlap in time is given by constraints (4.97) while the overall overlap Go_{ij} in both frequency and time is given in (4.98). Finally the cumulative interference constraints is given by (4.99).

5 Computational experiments

5.1 Binary interference

The scheduling models are coded in C++ and solved using IBM CP Optimizer [79]. The results were obtained on a 2.7GHz Intel Core i5 machine with 4GB RAM. The CPU times for the calculations were limited to 60s, unless stated otherwise.

For each of the test instances generated in the previous chapter and instances for 500 and 1,000 users generated solely for the maximal clique test in this chapter, a frequency demand vector having its length equal to the number of users is randomly generated from a set of $\{5, 10, 15, 20\}$ MHz values.

We started the experiments by studying the benefit of using maximal cliques. In order to do this, we solved for the minimum makespan and the number of scheduled tasks over 100 instances of 500 and 1,000 users by varying the maximal cliques usage in the model from 100% (using all maximal cliques) to 0% (using all constraint pairs). The number of optima was also counted. Note that maximal cliques were enumerated using Bron-Kerbosch algorithm [26], [131], [28] which is *Algorithm 457* in ACM collection. The algorithm is also coded in C++.

The results are shown in Figure 4.3 and 4.4. It can be seen that maximal cliques help improving the performance of the calculation. In Figure 4.3, Makespan is considerably lower in 100% maximal cliques usage compared to none. The trend is similar even in 1,000 user case which is considered hard to solve. The number of optima and the number of scheduled tasks are shown in Figure 4.4, tests on 500 users show consistent result that maximal cliques give us calculation benefit. Nonetheless, tests on 1,000 users show another interesting result. Note that, in this case, the number of optima is none even in the case of 100% maximal cliques usage. This means that the problem is much harder to solve. The number of scheduled tasks reduces as the percentage of maximal clique usage decreases from 100% to 80% then it increases afterwards, yet the difference between the minimum and the maximum value is marginal. This indicates that the benefit from maximal clique usage is limited for hard problem when the graph is too loaded.

Average numbers of maximal cliques and the average time needed to list all of them are shown in Table 4.1. The maximal clique listing time is low for 500 users but rather high for 1,000 users. Note that, for a large number of users, a faster depth-first search algorithm [151] which employs the same pruning method as Bron-Kerbosh algorithm could be used.

Table 4.1: Average number of maximal cliques.

	500 users	1,000 users
Average number of maximal cliques	1135.2	5241.52
Average time to list all maximal cliques (s)	1.03	22.95

In order to perform further tests, we need to find good binary interference matrices to base on. We added constraint violation check for all assigned users of each feasible solutions got from solving our makespan binary interference model. By varying the loading factor values, the total number of constraint violations and number of optima are provided in the Tables 4.2 and 4.3 .

Table 4.2: Number of constraint violations for different values of loading factors.

Number of violations	20	40	60	80	100	120	140	160	180	200
0.8	11	103	329	753	1063	1589	2007	2793	3170	4161
0.6	3	13	15	52	54	69	137	168	197	290
0.5	2	1	3	3	3	5	5	12	7	19
0.4	2	0	0	0	0	0	0	0	0	0

Reducing the LF causes more interference load (higher number of interference pairs) resulting in harder instances to solve as can be seen from reduction of the number of optima; however, the number of constraint violations decreases. We thus chose the interference matrices based on the loading factor 0.4 (no constraint violation) for further tests.

For minimizing makespan problem, we conducted a test comparing the results when using constraints from maximal cliques (MC) and constraints

Table 4.3: Number of optima for different values of loading factors.

Number of optima	20	40	60	80	100	120	140	160	180	200
0.8	100	100	100	100	99	97	97	99	96	96
0.6	100	95	98	95	93	84	77	66	46	34
0.5	100	98	94	95	87	76	55	49	21	13
0.4	100	97	95	92	82	64	47	21	10	5

directly from binary interference matrix (IM). Results are shown in Figure 4.5.

As expected, the model utilizing maximal cliques gives better results both in terms of makespan and the number of optima found. Nonetheless, the gap is small for makespan. The performance gap is small also when comparing the maximum number of scheduled tasks with a common due date (bandwidth) set to 60 MHz and 100 MHz, see Figure 4.6.

Both models show the consistent trends when the frequency resource is increased from 60 MHz to 100 MHz. The number of optima gap between both models is the same in 60MHz and 100 MHz bandwidth.

Maximal cliques gives slightly worse results in term of the number of scheduled tasks. But in terms of number of optima, it still performs better. Nonetheless; when looking into more details on the solver's parameter shown in Table 4.4, we can see that the models with constraints based directly on interference matrix (IM_1 to IM_3) yields almost twice the number of searching branches, requires higher number of variables and constraints. Thus, we could infer that if we allow longer solving time, the models with constraints based on maximal cliques (MC_1 to MC_3) have more potential to give better results.

5.2 Cumulative interference

Based on scheduling model, we solved the multiple carrier frequency assignment problem with cumulative interference in term of number of assigned users. Similar to the binary interference case, two bandwidth settings are used: 60 MHz and 100 MHz. Results are compared with the corresponded binary case with different loading factors, see Figure 4.7.

The number of assigned users saturate at 60 users for 60 MHz bandwidth case and 120 users for 100 MHz bandwidth case. Both are much lower than the results based on binary interference even that with loading factor (LF) of 0.4 with no constraint violation. This poses an interesting finding in that

Table 4.4: CP Optimizer solver status at 60MHz bandwidth and 60s solving time.

Instances	# branches	# fails	# Choice points	# variables	# constraints
MC_1	61464	19831	42707	2050	3501
MC_2	65981	24436	42225	2167	3735
MC_3	53513	19140	34592	2563	4527
IM_1	107787	38501	71892	2781	4963
IM_2	120988	43058	80548	2959	5319
IM_3	109413	38081	74319	3001	5403

solving the problem with the binary interference constructed from the actual interference matrix is easier and gives better results. This finding is consistent with a recent paper proposed by Graham et al. [67].

We would like to know how well this CP-based formulation perform compared to the ILP-based formulation. So we solved the single carrier cumulative interference problem presented in Chapter 3 using this CP formation (scheduling-based) by setting the demand of all users to 1. It turns out that the CP performs slighly worse than the ILP. Comparison is shown in Figure 4.8. Note that the calculation time is limited to 60 seconds in both cases.

Would this be the same for the multiple carrier case? We compare this CP-based formulation for multiple carrier case with three ILP formulations proposed in the chapter which are: ILP Model 1 (direct frequency model), ILP Model 2 (frequency-indexed model INDEXED1) and ILP Model 3 (frequency-indexed model INDEXED2). On the contrary, it can be clearly seen from the Figure 4.9 that, at more than 40 users, all of the ILPs perform much worse than the CP.

At 20 users, all the ILP models perform the same. After that, ILP Model 3 gives the worst performance. ILP Model 1 generally performs better than Model 2 except at 60 users. At 100 users or more, only Model 1 that can output feasible solutions. Note that the calculation is limited to 60 seconds.

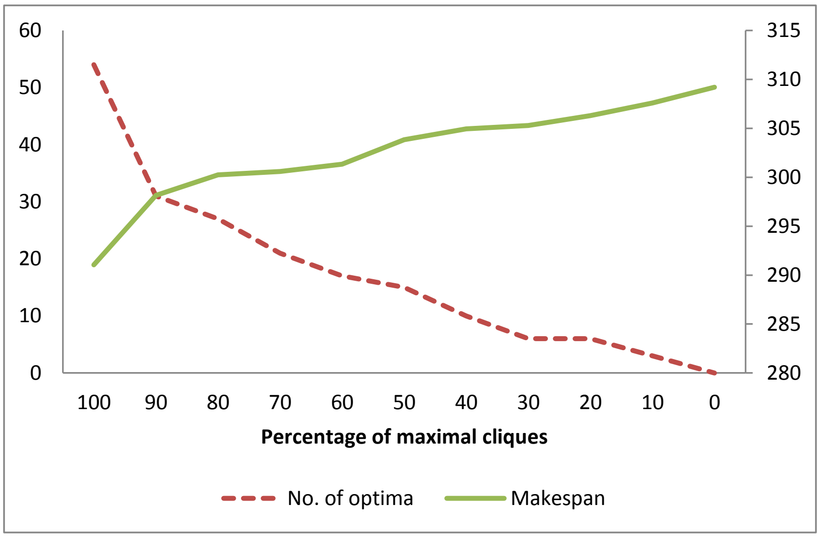
6 Conclusion

Frequency assignment can be treated as scheduling problem which can be solved efficiently by CP Optimizer. For binary interference environment, the concept of maximal clique can also be applied to improve the solving performance.

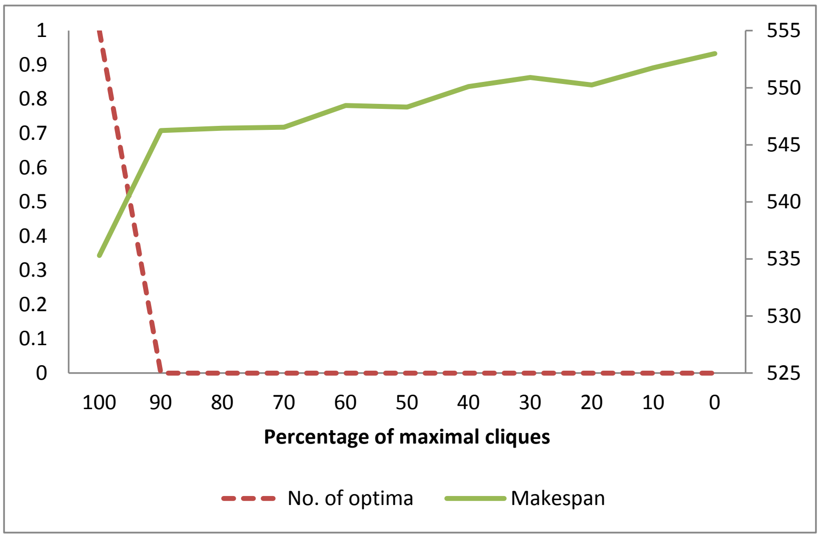
It is shown that for single carrier case, scheduling-based formulation performs as well as the ILP-based formulation while for multiple carrier case, this scheduling-based outperforms those of ILP, with a large performance gap. The performance is even better when we transform the cumulative interference to binary interference and solve the problem using the combined scheduling-based formulation and maximal clique concept (the solution should also cause no constraint violation based on the cumulative interference).

We have not tested all the proposed ILP formulations especially the pattern-based which lead to an interaction with column generation method, which could greatly improve the performance. Heuristic based on interval graph coloring, though not tested here, could also provide fast solving speed. Nonetheless, we have not extended the model to cope with cumulative interference. Both pattern-based ILP and interval graph coloring with cumulative interference would be interesting topics to explore further.

We also proposed an ILP formulation for 2-dimensional frequency assignment problem. Based on the 1D performance, this complex model would need long calculation time. Nonetheless, it would be interesting if we could combine this with a local search technique in order to improve it.

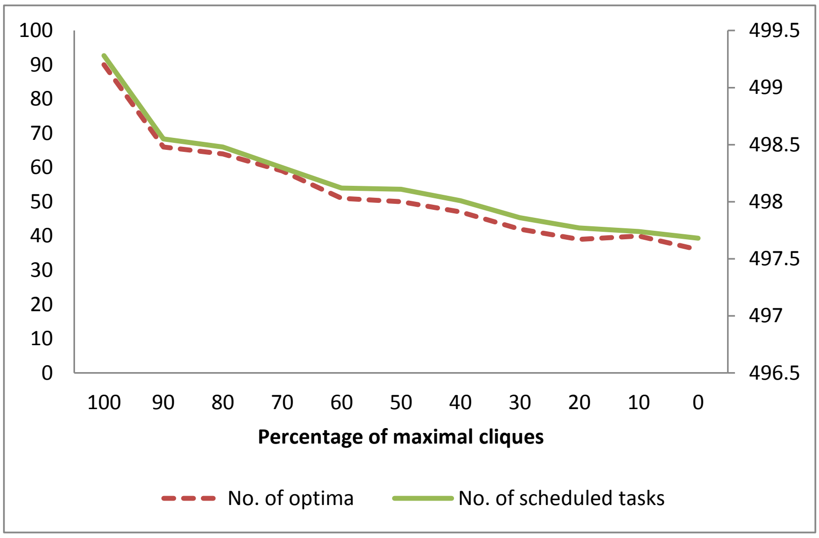


(a)

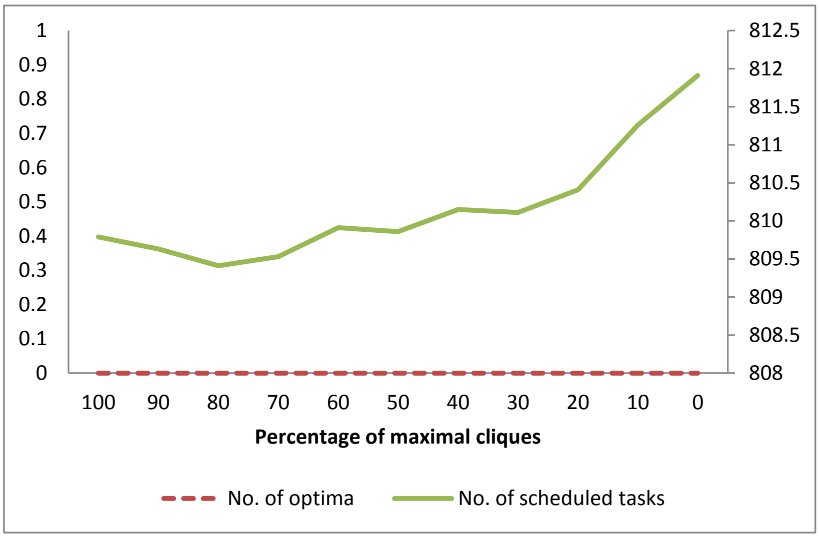


(b)

Figure 4.3: Number of optima and makespan versus percentage of maximal clique usage (a) 500 users (b) 1,000 users.

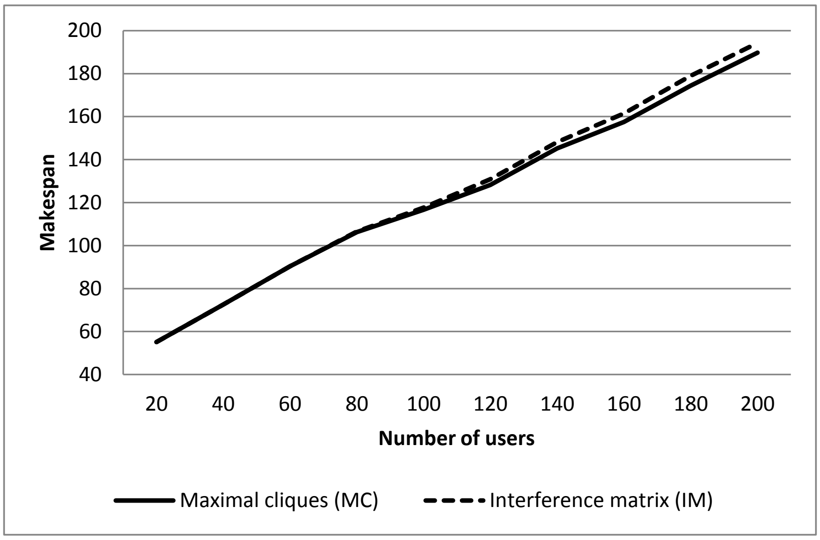


(a)

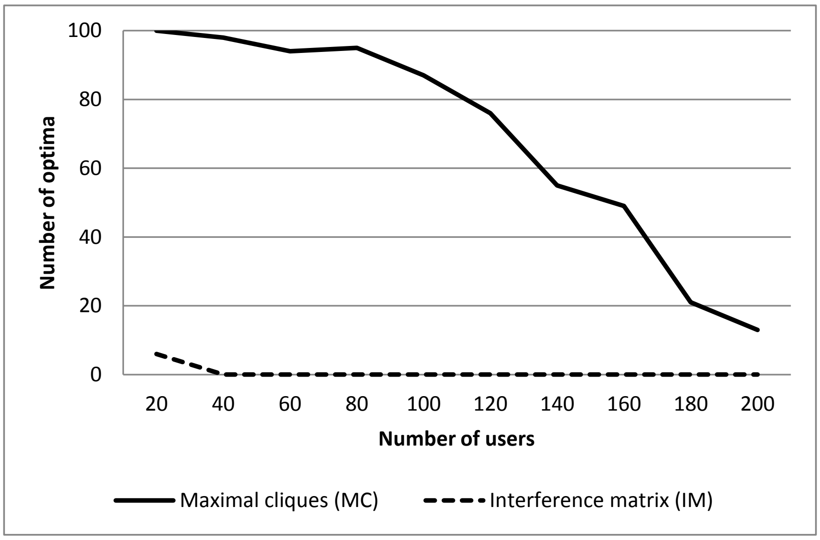


(b)

Figure 4.4: Number of optima and scheduled tasks versus percentage of maximal clique usage (a) 500 users (b) 1,000 users, both with 300MHz bandwidth.



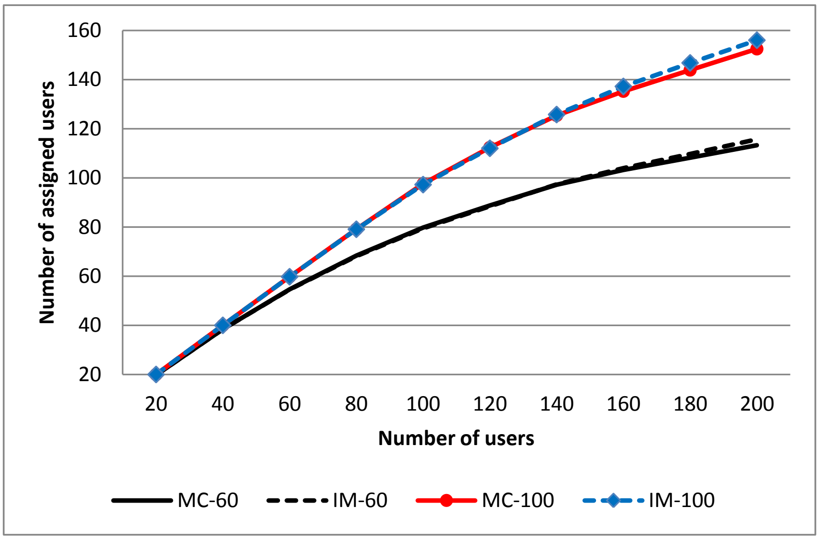
(a)



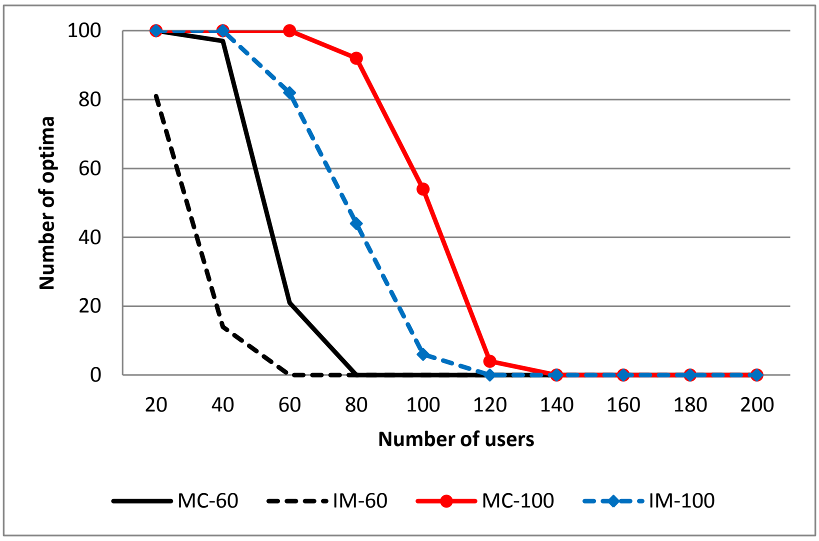
(b)

Figure 4.5: Makespan and number of optima by using constraints either from maximal cliques or binary interference matrix.

Chapter 4

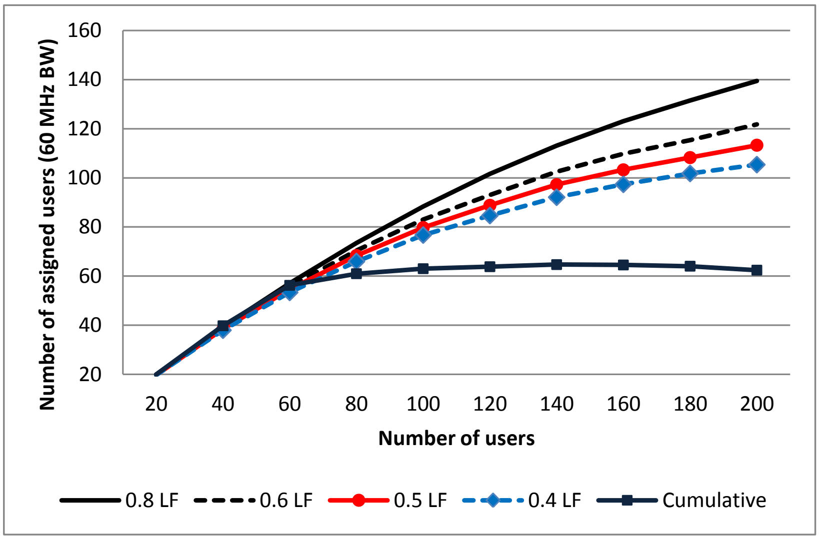


(a)

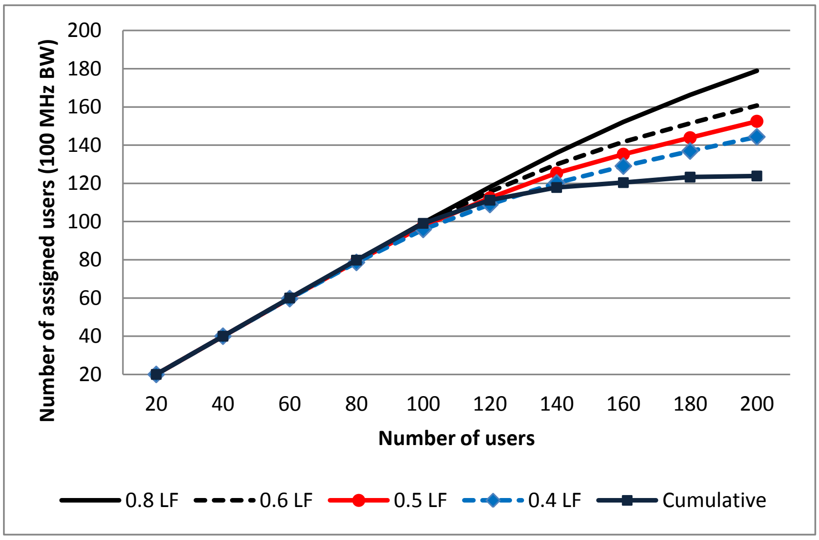


(b)

Figure 4.6: Number of scheduled tasks by using constraints either from maximal cliques (MC) or binary interference matrix (IM).



(a)



(b)

Figure 4.7: Comparison between the cumulative interference and binary interference with difference loading factors in term of number of assigned users (a) 60 MHz bandwidth (b) 100 MHz bandwidth.

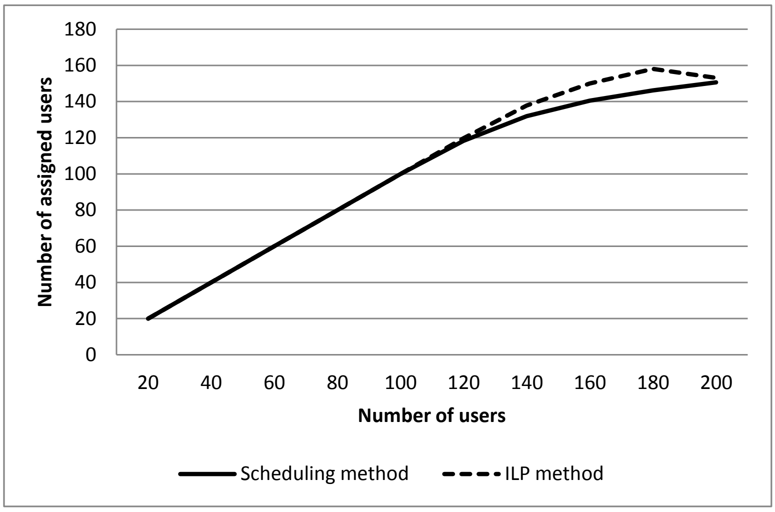


Figure 4.8: Comparison between CP and ILP for single carrier case.

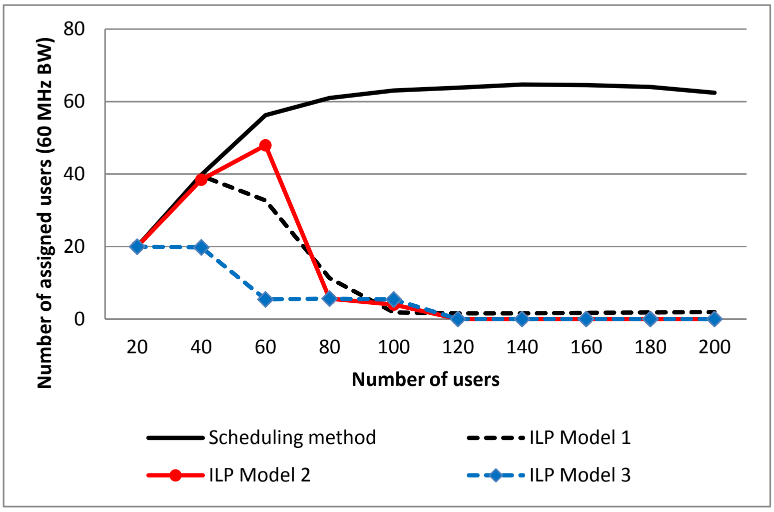


Figure 4.9: Comparison between CP and ILP for multiple carrier case.

Chapter 4

Industrial application

5

1 Introduction

This chapter presents greedy algorithms for frequency assignment in a SDMA satellite communication system based on parameters and requirements from the industry [33]. The study is a follow-up of a collaboration between Thales Alenia Space, LAAS-CNRS, and IRIT [11]. It is based on publication [91].

This frequency assignment problem requires assignment on both frequency and time. Unlike the single and multiple carrier models that the required signal to noise ratio is constant for all users, in this application, users can have different values which is depended on terminal type, traffic type, bitrate demand, and the selected RsModCod. Interference thresholds (α_i) and interference coefficients (δ_{ij}) also vary from an assignment to another. Moreover, user can be assigned a resource taken from a set of $frequency \times time$ rectangles having identical frequency-time product.

Additional requirements are provided in the section below. The objective is to serve as many number of users as possible according to the given constraints. The uplink is considered in the study.

In the industrial specifications, the solution method should be fast so as to be used in a dynamic context. Hence ILP and CP methods experienced in the previous chapters could not be selected. Therefore, two greedy algorithms are proposed and tested. Results are provided in the Computational experiments section.

2 Additional requirements

2.1 Resource optimization constraints

- For a given bitrate, the RsModCod configuration should be selected with the lowest possible symbol rate (see RsModCod section).
- The number of satellite beams should not exceed *NbBeamsMax*.
- The total system bandwidth which is sum of all frequency carriers used, should not exceed *BWMax*. A frequency carrier is considered utilized if any part of it is assigned to users. Same frequency carriers f_n are counted separately if they are from different superframes.
- The maximum number of frequency carriers that can be assigned to each user is provided by *FrameBWMax*.

2.2 User priority

A user is associated with a priority. There are four different priority classes ranging from 0 (the highest) to 3 (the lowest). These priority classes determine user ordering for the resource allocation. Users with higher priority are considered first. Nonetheless, if only following this rule, low priority users would never been treated. To avoid this, the Weighted Round-Robin (WRR) algorithm is applied. The algorithm selects (4-Priority) users from each of the class, see the WRR ordering (right to left) in Figure 5.1.

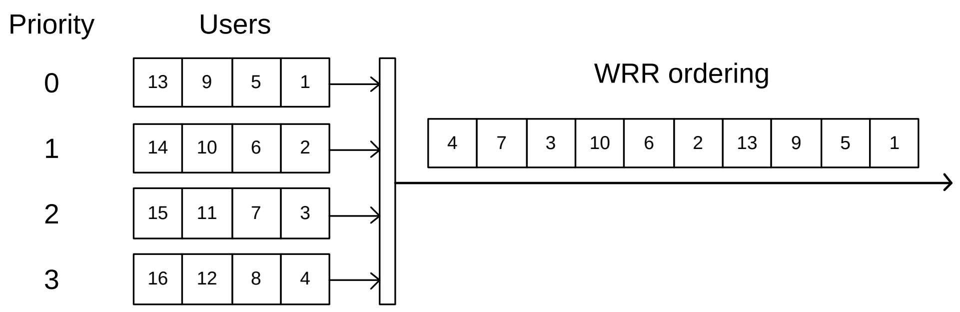


Figure 5.1: Weighted Round Robin user selection.

2.3 Terminal type

User's demand is treated in form of bitrate. A user can request for any bitrate not greater than a supported value based on the terminal type. Two terminal types are employed: Type 1 and 2. Type 1 supports up to 24 Mbps while Type 2 supports up to 12 Mbps.

These two terminal types have different output powers. Type 1's maximum effective isotropic radiated power (EIRPTerm) is 50 dBW and Type 2's maximum EIRPTerm is 45 dBW.

2.4 Traffic type

Traffic type separates users into guaranteed (Type 1) and non-guaranteed (Type 0). Guaranteed users will be rejected if the system could not assign resources corresponding to their bitrate demands whereas for non-guaranteed users, the system has possibility to consider providing them with bitrate at one step lower than their initial demands except that the current bitrate demand is already the lowest that the system can support.

2.5 Symbol rate, modulation and coding scheme (RsModCod)

Before proceeding to the resource allocation, a user should be associated with a symbol rate (RS), a modulation (Mod), and a coding scheme (Cod). For each of terminal type, there are 64 predefined combinations of RsModCod. These combinations are based on the following parameters:

- Modulation and coding scheme (ModCod): 16 configurations
- Required signal to noise ratio: 16 values, corresponding to each of ModCod configurations
- Bandwidth: 5, 10, 15 and 20 MHz
- Symbol rate: 4.167, 8.333, 12.5 and 16.677 Msymb/s
- Support bitrate
- Estimated signal to noise plus interference ratio

Signal in form of digital bit stream is coded and then modulated prior to transmission. Coding adds robustness to the original bit stream in order to combat with noise and interference. Different coding schemes provide different levels of protection. Generally, the better the protection, the more overhead is added and the less original message presents in the coded bit stream.

Modulation maps 2^M coded bits to M symbols (e.g. M = 2 for QPSK and 3 for 8PSK) which each corresponds to a waveform to be transmitted. One of these M possible waveforms is transmitted in a given time period T.

The communication rate R, in bits per second, is thus $log_2(M)/T$. The signal occupies a given bandwidth BW Hz so the normalized rate of communication is R/BW and is measured in bits/second/Hz. Larger M also requires higher signal to noise ratio for a successful communication.

A combination of a coding scheme and a modulation type (ModCod) provides unique communication rate (bits/second/Hz) and requires a specific signal to noise ratio. Total of 16 ModCod configurations are listed (see Appendix A). RsModCod is created by combining these 16 ModCod configurations with 4 different symbols rates which are 4.167, 8.333, 12.5 and 16.677 Msymb/s corresponding to 5, 10, 15 and 20 MHz signal bandwidth.

Each RsModCod thus provides different support bitrate and requires different signal to noise ratios. For a successful communication, the user's actual signal to noise plus interference ratio should not be lower than the signal to noise ratio requirement. Nonetheless, the actual signal to noise plus interference ratio will only be available after the connection is assigned with a RsModCod and the frequency $\times$ time resources and interference from other users is determined.

Since the actual signal to noise plus interference ratio is not available at the time of RsModCod selection, the estimated value of it is used instead. The estimated signal to noise plus interference ratio is calculated based on the required signal to noise ratio, a set of system and wave propagation parameters and a fixed interference level. Only the combinations having the estimated signal to noise plus interference ratio not less than the required signal to noise ratio will be considered valid.

For each user, the system searches the RsModCod table for the valid combinations having their support bitrates immediately better than the bitrate demand. According to the resource optimization constraints given above, the combination that has the lowest symbol rate should be chosen. If there is no RsModCod available and the user's traffic type is non-guaranteed, valid combinations having bitrates immediately lower than the bitrate demand will be considered and, among these, the one with the lowest symbol rate will be chosen. If none of this case is true, the user is rejected.

The chosen combination will be proceeded for the frequency and time assignment. If the assignment is failed and the user's traffic type is non-guaranteed, combinations having their support bitrates immediately lower than bitrate demand and their required signal to noise ratios lower than that of the previously chosen one will be considered. Again, the combination having the lowest symbol rate should be chosen.

Consider a user with Terminal Type 1 and Traffic Type 0 requesting for a 10 Mbps bitrate demand as an example. From the RsModCod table provided in Appendix A, Combinations 15, 22, 36, and 50 offer bitrate immediately better than 10 Mbps. Combination 15 cannot be chosen since its estimated signal to noise plus interference ratio is lower than the required signal to noise ratio. Combination 22 is chosen since it has the lowest symbol rate (requires 10 MHz bandwidth in this case). This combination is tried in the frequency and time resource assignment.

Suppose that the assignment is failed, since the user has Traffic Type 0, the system then lists other RsModCod combinations that support immediately lower bitrate than 10 Mbps and have their required signal to noise ratios lower than 3.7 dB. They are Combination 21, 35. and 49. Combination 21 is then chosen as it has the lowest symbol rate.

2.6 Beam positioning method

We consider two beam positioning methods which are fixed-beam and SDMA-beam. For fixed-beam configuration, the number of satellite beams is fixed at 40. Beams are positioned in the same manner as in the fixed-beam model provided in the Chapter 3. In this case, the users are assigned to the closest beam. Users within the same beam can only be assigned to the same frame.

For SDMA-beam, the number of beams should not be greater than the *NbBeamsMax*. In case that there are more users than beams, the beam assignment will be performed on the first-come, first-served basis. The first NbBeamsMax users will get their dedicated beams centered to them. Each of the following users will be assigned to the beam which is closest to it. Different beam users cannot share the same frame.

In both cases, there is no interference among users sharing the same superframe.

2.7 Uplink power control

Uplink power control function is employed in order to reduce the interference in the system. It provides possibility for the user terminals to reduce its transmission power to a level that is sufficient for their maintaining communication links.

In the resource assignment phase, all user terminals use their maximum output power. Then power control is performed, each user terminal's transmitted power (EIRPTerm) is evaluated if it can be reduced without impairing the communication link. The reduction is done in conjunction with a predefined power margin, PCMargin. This power control reduction should be made after the frequency assignment phase.

3 Greedy algorithms

Greedy algorithm is proposed as of its simplicity and speed. It can be tailored according to the given specifications. The drawback of the algorithm lies on no-look-back concept in that the already assigned users or rejected users will not be reconsidered again. We are trying to assign each user with its demand to a slot, a frame, and a superframe. Slot, frame and superframe are defined by an assignment of time and frequency. Slot can vary in size while one or more slots could be fit inside a frame. A frame is a slice of a superframe and can have different in size measured by number of frequencies. Assignment of slots inside a frame and a superframe follows constraints provided in Chapter 2 and additional requirements provided above.

Two greedy algorithms are provided namely Minimum Interference (MI) and Minimum Bandwidth (MB). Both share the same core concept but differ in the priority of the search for available slots and positions in that MB provides more possibility of utilizing lower bandwidth.

Input to the algorithm is a user profile consisting of a number of users with randomly generated demand, priority, terminal type, traffic type, and coordinates. These users are ranked first by their priority levels. From this ranked list, a user is then selected based on Weighted Round-Robin algorithm.

Before entering the assignment phase, each of the selected users will be assigned with an RsModCod. Note that a user might not get an RsModCod if there is no valid RsModCod corresponding to its demand. In this case, no further resource assignment will be performed.

For an RsModCod, the corresponding signal to noise plus interference radio and the bandwidth are provided. The former will be used for calculating the user's acceptable interference threshold (α_i) and interference coefficients towards other users (δ_{ij}). The latter will determine a set of valid combinations of slot size ($frequency \times time$) for the assignment.

Among all of the user's valid combinations of slot size, the one with lower bandwidth requirement is chosen first for an assignment try. This chosen slot size is tested in available positions (x_1, y_1 and x_2, y_2) of a superframe in which the X-axis represents time and Y-axis frequency. If there is no space left in a given superframe, a new superframe is created and tested. Nonetheless, the total number of superframe cannot be greater than the number of the satellite beams.

No overlapping both in time and frequency between slots is allowed within the same superframe; nonetheless, overlapping either in time or frequency or both could exist between users from different superframes. In this case, an interference between overlapping slots (or users) present. Interference between two users is mutual and the level of interference is depended on how large the overlapping area is. A user can get interference from more than one user and interference level adds up. This cumulative interference should not exceed the user's acceptable interference threshold.

3.1 Slot combinations

A slot size is defined by the resource demand in $frequency \times time$ which is an output of RsModCod selection and given in form of bandwidth. 5 MHz bandwidth corresponds to a slot size of $FrameDuration \times 5$ whereas $FrameDuration$ is the maximum possible slot length and 5 means 5 frequency carriers.

This slot size will be tested in the resource assignment. If it fails, other slot sizes, having the same frequency-time product are tested.

FrameBWMin and *FrameBWMax* define possible values of slot frequency. In this study, they are 1 and 30 MHz respectively: the possible values of slot width are $1, 2, \ldots, 30$.

SlotDuration and *FrameDuration* define possible values of slot time. In this study, they are 0.01 and 0.1 seconds respectively: the possible values of slot length are $1, 2, \ldots, 10$ per unit (of 0.01 seconds).

Slot combinations are listed based on these possible values. For example, slot combinations for a 5 MHz resource demand are $\{(5 \times 10), (10 \times 5), (25 \times 2)\}$. These combinations correspond to $n \times m$ in the algorithms where $n \in \{1, \ldots, 30\}$ and $m \in \{1, \ldots, 10\}$.

3.2 Minimum interference

In Minimum Interference, the algorithm searches, based on the slot combinations, for superframe and slot position that gives lowest interference. At the beginning of the assignment process, the first user is assigned at the bottom-most position in the first superframe. Next user's slot will be searched starting from free position on top of the first user. If there is no space available, the next superframe becomes active. With greedy algorithm concept, slot position will be assigned where it results in the lowest interference at the moment of the calculation. Figure 5.2 below presents an example of a slot assignment in the second superframe. The assigned position yields the lowest interference.

3.3 Minimum bandwidth

In general, minimum bandwidth is conflicting with minimum interference. Nonetheless, since the system is limited by interference, we choose to implement Minimum Bandwidth based on minimum interference concepts.

Instead of moving up in frequency within the same superframe to search for minimum interference position, Minimum Bandwidth searches first for an

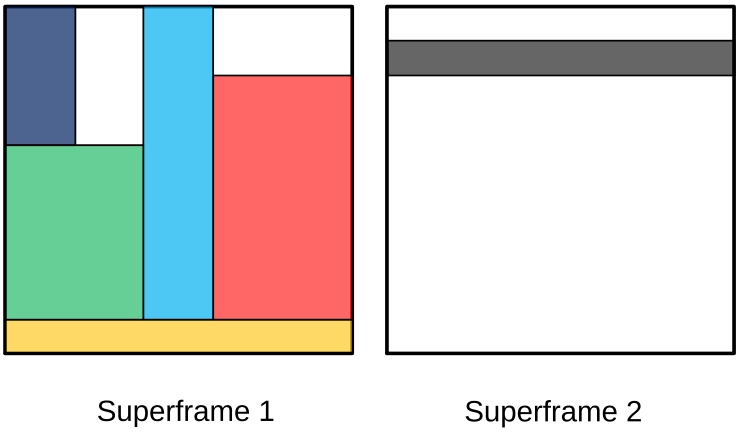


Figure 5.2: Minimum interference assignment.

unallocated area in another superframe and select the position which results in the lowest interference. Minimum bandwidth can thus be considered as minimum interference with modified search priority. Figure 5.3 below presents an example of a slot assignment in the second superframe. The assigned position yields the lowest interference. If it were to be Minimum interference algorithm, instead of choosing this position, a position in Superframe 1 at higher frequency (and no interference) will be chosen.

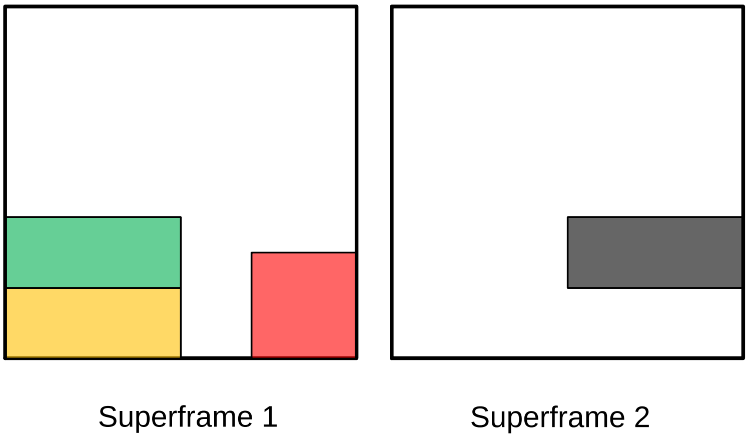


Figure 5.3: Minimum bandwidth assignment.

Minimum interference and minimum bandwidth algorithms are given in Algorithm (6) and (7) respectively.

3.4 Algorithm implementation details

Testing all slot sizes in all superframes and all available positions is time consuming and inefficient. Instead, we introduce five controlling parameters in order to limit the search space.

SpecificS provides a specific superframe that should be tested. It is used when assigning users served by the same satellite beam as the already assigned user(s).

CapacityS checks for the remaining capacity of a superframe whether it can support the given slot size ($frequency \times time$). If the superframe has not enough capacity, it will be skipped and the next superframe will be considered.

YLevel acts like a water level in that the test starts from this level and moves upwards, not every time from the bottom of each superframe.

OverlapOwnS determines if there is an overlap between the test position and the already assigned positions within the same superframe (*i.e.* no interference allowed for users residing in the same superframe). The overlapping case is skipped and test moves forward.

LastS provides the updated number of the active superframes and the search is performed at up to *LastS*$+1$ superframe not to the maximum number of superframe which is equal to NbBeamMax.

Frame structure constraints are verified by *c6check* variable.

totalSlack denotes interference gap between the acceptable interference threshold and the current cumulative interference level.

Greedy algorithm is given in Algorithm (4) for Assignment CASE 1 (fixed-beam) and 2 (SDMA-beam with the number of users not greater than the number of satellite beams). Assignment CASE 3 (SDMA-beam with the number of users greater than the number of satellite beams) is given as modification of Algorithm (4) in Algorithm (5).

3.5 C/(N+I) Calculation

Since the interference pattern changes after every user assignment; at the end of the user assignment phase, $C/(N+I)$ calculation is performed. The calculation result is used as input to the power control phase.

3.6 Power control

Uplink power control is performed in the final phase of the algorithm. Output power of each user is reduced to a lowest possible level but still maintaining *PCMargin* which is a gap between the calculated $C/(N+I)$ and the required C/N. The reduction is performed to all users at the same time. After the reduction, interference pattern changes, and since the change is non-linear, further power reduction is possible.

Thus power reduction is performed iteratively. After each reduction, $C/(N+I)$ is updated. If the power gap is lower than the *PCMargin*, no power reduction is performed. The iteration continues until there is no user terminal eligible for reduction.

4 Computational experiments

Greedy algorithms are coded in Matlab [119]. Simulations are performed on five different test environment: Env. 1 - Env. 5, each with 100 test instances. Each instance consists of 30 users, half of them are Terminal Type 1's and another half are Terminal Type 2's. Each user is associated with a bitrate demand, a priority class, a traffic type, and a pair of geographic coordinates. The bitrate demand is randomly generated from a predefined range $[a, b]$ inclusive corresponding to the terminal type and the test environment as shown below:

Table 5.1: Test instance characteristics.

Instance	Category	Bitrate demand (Mbps) Terminal Type 1	Terminal Type 2	BW
Env. 1	low demand	[1, 12]	[1, 5]	60
Env. 2	average demand	[1, 24]	[1, 10]	60
Env. 3	high demand	[12, 24]	[5, 10]	60
Env. 4	low bandwidth	10	10	60
Env. 5	high bandwidth	10	10	100

Fixed demand is applied in Env. 4 and Env. 5 in that the bitrate demand for all users is fixed to 10 Mbps. Priority, traffic type and geographic coordinates are randomly generated. The superframe bandwidth *BW* is set to 60 MHz for Env. 1 to Env. 4 and 100 MHz for Env. 5. The system bandwidth is set to 300 MHz.

The simulations were performed on an Intel Core2 Duo 2.4 GHz machine with 4 GB RAM over variations of algorithms and satellite beam configurations. Four indicators i.e. Assignment time, Number of rejected users, Total slack, and Frequency utilization are compared and presented in the sections below. The following abbreviations are used:

- MI: Minimum Interference
- MB: Minimum Bandwidth

- FB: Fixed Beams (40 beams)
- BC30: Beam-centered with 30 beams
- BC25: Beam-centered with 25 beams

4.1 Assignment time

Assignment time measures the duration of the resource assignment (RsModCod selection and frequency and time assignment) of all users. The average assignment time is calculated over 100 instances and listed in Table 5.2.

Table 5.2: Average assignment time (seconds).

Algorithm	Env. 1	Env. 2	Env. 3	Env. 4	Env. 5
MIFB	5.34	6.49	6.65	5.77	8.72
MBFB	1.03	4.77	6.53	5.77	7.52
MIBC30	8.50	10.64	10.15	6.95	14.96
MBBC30	2.55	10.49	10.11	8.19	13.58
MIBC25	6.32	10.49	10.40	6.95	14.00
MBBC25	1.89	9.54	9.97	9.23	13.41

Beam-centered configurations require longer assignment time than fixed-beams configuration for both MI and MB cases. This is based on the facts that (1) beam-centered configurations give higher acceptable interference thresholds granting the algorithm more calculation possibility and (2), in the fixed-beam configurations, frame constraints limit the search space for users assigned to the same beam.

For beam-centered configurations, lower number of beams requires less assignment time. This is also resulted from frame constraints imposition.

With average and high demands, MI takes about the same assignment time as MB. Nonetheless, with low demand, the former takes much longer. This indicates that the user demand impacts a lot on the algorithm performances.

High bandwidth requires longer time since there is more search space in each superframe. Note that the MATLAB environment was selected due to compatibility with the industrial requirements. The assignment times could be further reduced by recoding the algorithms in a more appropriate language, e.g. C++.

4.2 Number of rejected users

Rejected users are users that the algorithm fails to assign the resources, either the RsModCod or the frequency and time assignment. Average number of rejected users are shown in Table 5.3.

Table 5.3: Number of rejected users.

Algorithm	Env. 1	Env. 2	Env. 3	Env. 4	Env. 5
MIFB	5.79	11.95	15.29	15.61	10.51
MBFB	1.98	14.54	17.40	18.01	11.00
MIBC30	0.93	10.58	14.17	14.71	10.27
MBBC30	1.40	14.03	17.24	17.74	10.73
MIBC25	4.15	10.60	14.17	14.71	10.27
MBBC25	2.13	14.03	17.24	17.74	10.73

It is not surprising that the number of rejected users depends largely on the demand or bandwidth.

For Env. 1 (low demand), the results are highly satisfactory as only few users are rejected. The best method, for which almost no user is rejected in average, is the MI with beam centered and 30 beams (thus fully exploiting the SDMA technology). Surprisingly, the MB becomes the best choice for beam centering allocation when the number of beams reduces to 25. MB shows also a much better performance than MI when the beams are fixed.

When demand increases the MI algorithm uniformly performs better than the MB. However the number of rejected users dramatically increases. Comparison between Env. 4 and Env. 5 shows the high impact of the bandwidth availability when the demand is fixed to a high value. For these highly constrained scenarios the MI always performs better than the MB.

4.3 Total slack

Total slack for a user is initialized by its interference threshold (α). This gap is reduced when there is interference from other users. Larger interference gap means lower interference. Total interference gap is the summation of interference gap of all assigned users. The average total slacks are shown in Table 5.4.

As expected, MI gives higher total slack than MB; nonetheless the gap is wider in the low demand case. The fixed-beam configurations give slightly better total slack than that of the beam-centered, this is understandable if we

Table 5.4: Total slack (1E-18).

Algorithm	Env. 1	Env. 2	Env. 3	Env. 4	Env. 5
MIFB	3.66	1.66	0.82	1.19	1.49
MBFB	2.20	1.31	0.68	0.97	0.99
MIBC30	3.84	1.64	0.79	1.21	1.41
MBBC30	2.13	1.27	0.69	0.98	1.00
MIBC25	3.72	1.65	0.79	1.21	1.42
MBBC25	2.17	1.27	0.69	0.98	1.00

also consider the number of rejected users: higher number of rejected users, lower interference level they cause.

4.4 Frequency utilization

A frequency is considered utilized if any part of it is assigned to users. Same frequencies from different superframes are treated as different ones. Total number of utilized frequencies are counted, averaged over 100 instances and rounded up to the nearest integer values. This indicator generally portrays how much the frequency resource is utilized. In fact, this figure is considered as the total utilized bandwidth. Results are shown in Table 5.5.

Note that the satellite system supports a certain amount of bandwidth, which is 300 MHz in this case.

Table 5.5: Number of used frequency.

Algorithm	Env. 1	Env. 2	Env. 3	Env. 4	Env. 5
MIFB	146	195	205	217	293
MBFB	164	162	169	171	279
MIBC30	178	213	217	231	298
MBBC30	167	166	170	175	281
MIBC25	158	212	217	231	298
MBBC25	163	166	170	175	281

The MB configurations require lower number of frequencies than MI configurations. Nonetheless, a contradicting result can be found in a low demand case (MIFB vs. MBFB). This result also relates to the number of rejected users that MBFB yields much lower number of rejection than MIFB. This could be the case that since MB does not always assign the least interfered position and the user demands are low, more users could be packed in the same manner as shown in Figure 5.3.

5 Conclusion

Specifications and constraints provided by the industry render the resource allocation problem highly complex. This complexity and the fact that frequency assignment plans must be recomputed frequently in order to cope for user mobility yield classic optimization tool such as Integer Linear Programming impractical. Greedy algorithms have to be proposed for this problem. Two greedy algorithms are devised and tested.

When the user demand is reasonable, the proposed greedy algorithms obtain a user acceptance rate that has been judged as satisfactory by the industrial partner. Nevertheless, when the problem becomes highly constrained, especially when the demand increases or when the available bandwidth is limited the performance dramatically decrease. Future work could be focused on computing upper bounds on the number of accepted user to be able to estimate the possible performance gain in highly constrained environment. Then local search heuristics could be proposed to further improve the greedy algorithm.

Input: Input parameters, User profile, Beam positions
Output: *userAllocation*,*nbReject*,*totalSlack*,*Allocation time*,*numberFrequencyUsed*

$s_i, \mathit{c6check}_i \leftarrow 0, \forall i = 1, \ldots, \mathit{nbSuperframeMax}$
$\mathit{capacityS}_i \leftarrow$ size of $s, \forall i = 1, \ldots, \mathit{nbSuperframeMax}$
$\mathit{yLevel}_i \leftarrow$ height of $s, \forall i = 1, \ldots, \mathit{nbSuperframeMax}$
$\mathit{nbReject} \leftarrow 0$
$\mathit{allocationTry}_i, \mathit{allocationFail}_i, \alpha_i, \delta_{ij}, \mathit{RS}_i, \mathit{CsN}_i, \ldots \mathit{ModCodSelected}_i, \mathit{totalSlack}_i, \mathit{userAllocation}_i \leftarrow 0, \forall i = 1, \ldots, \mathit{nbUser}$
sort User profile (by priority and demand)
for $\mathit{allocationTry} = 1$ **to** nbUser **do**
    $i = \mathrm{WRR}(\mathit{allocationTry})$
    $(\mathit{RS}_i, \mathit{CsN}_i, \mathit{ModCodSelected}_i, d_i, \mathit{allocationFail}_i) = \mathit{RsModCodSelect}(i, \mathit{bitrate}_i, \mathit{terminalType}_i)$
    if $\mathit{allocationFail}_i = 0$ **then**
        calculate $\alpha_i, \delta_{ij}, \forall j | \mathit{RS}_j \neq 0$
        $\mathit{totalSlack}_i \leftarrow \mathit{totalSlack}_i + \alpha_i$
        $\mathit{specificS} \leftarrow k$ if i shares the same beam k of j and $\mathit{RS}_j \neq 0$
        $(\mathit{allocationResult}, \mathit{slackResult}, \mathit{c6Check}) =$ allocation_MI$(\mathit{nbUser}, \mathit{nbSuperframeMax}, i, d_i, S, \delta_{ij}, \ldots \mathit{c6check}, \mathit{totalSlack}, \mathit{beam}_i, \mathit{lastS}, \mathit{capacityS}, \mathit{yLevel}, \mathit{specificS})$
        if $\mathit{allocationResult} = 0$ **then**
            $\mathit{allocationFail}_i \leftarrow 1$
            $\mathit{nbReject} \leftarrow \mathit{nbReject} + 1$
            $\mathit{RS}_i, \mathit{CsN}_i, \mathit{ModCodSelected}_i, \alpha_i, \delta_{ij} \leftarrow 0$
            $\mathit{totalSlack}_i \leftarrow \mathit{totalSlack}_i - \alpha_i$
        else
            calculate $\mathit{numberFrequencyUsed}$
            if $\mathit{numberFrequencyUsed} \leq \mathit{BWMax}$ **then**
                $\mathit{userAllocation}_i \leftarrow \mathit{allocationResult}$
                $\mathit{totalSlack} \leftarrow \mathit{slackResult}$
                update $\mathit{capacityS}, \mathit{yLevel}, \mathit{lastS}$
            else
                $\mathit{allocationFail}_i \leftarrow 1$
                $\mathit{nbReject} \leftarrow \mathit{nbReject} + 1$
                $\mathit{RS}_i, \mathit{CsN}_i, \mathit{ModCodSelected}_i, \alpha_i, \delta_{ij} \leftarrow 0$
                $\mathit{totalSlack}_i \leftarrow \mathit{totalSlack}_i - \alpha_i$
            end
        end
    end
end

Algorithm 4: CASE 1 (fixed-beam), CASE 2 (SDMA-beam with the number of users not greater than the number of satellite beams)

```
Modify Algorithm for CASE 1,2 as following:
Line (4) to nbReject, beamCount ← 0
Insert if-else statement after Line (7)
if beamCount < nbBeams then
    beam_i ← user_i
    beamCount ← beamCount + 1
else
    beam_i ← beam_j | gain_{j→i} is maximum
end
Insert if statement after Line (19) and (30)
if beamCount < nbBeams then
    beamCount ← beamCount − 1
    beam_i ← 0
end
```

Algorithm 5: CASE 3 (SDMA-beam with the number of users greater than the number of satellite beams)

Input: $nbUser, nbSuperframeMax, i, d_i, S, \delta_{ij}, c6check, totalSlack, beam_i, lastS, capacityS, yLevel, specificS$

Output: $allocationResult, slackResult, c6check$

$maxSlack, allocationResult \leftarrow 0$

$M = \text{findM}(d_i)$

if $specificS = 0$ **then** $nbs \leftarrow \{1, \ldots, lastS + 1\}$ **else** $nbs \leftarrow specificS$

for $NBS = 1, \ldots, nbs$ **do**

  if $NBS > nbSuperframeMax$ **then** break

  if $capacityS_{NBS} \geq d_i$ **then**

    for $m = M_1, \ldots, M_n \in M$ **do**

      $n \leftarrow d_i/m$

      for $y_1 = yLevel_{NBS} - n + 1, yLevel_{NBS} - n, \ldots, 1$ **do**

        $y_2 \leftarrow y_1 + n - 1$

        for $x_1 = 1, \ldots, FrameDuration - m + 1$ **do**

          $x_2 \leftarrow x_1 + m - 1$

          $overlapOwnS = \text{overlap}(nbUser, S_{NBS}, x_1, x_2, y_1, y_2)$

          if $overlapOwnS = 0$ **then** $slack \leftarrow totalSlack$

          for $otherS = 1, \ldots, lastS$ **do**

            if $otherS \neq NBS$ **then**

              $overlapOtherS = \text{overlap}(nbUser, S_{\text{otherS}}, x_1, x_2, y_1, y_2)$

              $addInterference = \text{interference}(nbUser, i, d_i, overlapOtherS, \delta_{ij})$

              $slack \leftarrow slack - addInterference$

            end

          end

          if $slack > 0$ **then**

            if $slack_i > maxSlack$ **then**

              if $c6check\ PASS$ **then**

                update $allocationResult$

                $slackResult \leftarrow slack$

                $maxSlack \leftarrow slack_i$

              end

            end

          end

        end

      end

    end

  end

end

if $allotionResult \neq 0$ **then** update $c6check$ **else** $slackResult \leftarrow totalSlack$

Algorithm 6: Minimum interference

Input: $\mathit{nbUser}, \mathit{nbSuperframeMax}, i, d_i, S, \delta_{ij}, \mathit{c6check}, \mathit{totalSlack}, \mathit{beam}_i, \mathit{lastS}, \mathit{capacityS}, \mathit{yLevel}, \mathit{specificS}$
Output: $\mathit{allocationResult}, \mathit{slackResult}, \mathit{c6check}$

$\mathit{maxSlack}, \mathit{allocationResult} \leftarrow 0$
$M = \text{findM}(d_i)$
if $\mathit{specificS} = 0$ **then** $\mathit{nbs} \leftarrow \{1, \ldots, \mathit{lastS} + 1\}$ **else** $\mathit{nbs} \leftarrow \mathit{specificS}$
for $m = M_1, \ldots, M_n \in M$ **do**
$n \leftarrow d_i/m$
for $y_1 = \mathit{BWAvial} - n + 1, \mathit{BWAvail} - n, \ldots, 1$ **do**
$y_2 \leftarrow y_1 + n - 1$
for $x_1 = 1, \ldots, \mathit{FrameDuration} - m + 1$ **do**
$x_2 \leftarrow x_1 + m - 1$
for $\mathit{NBS} = 1, \ldots, \mathit{nbs}$ **do**
if $\mathit{NBS} > \mathit{nbSuperframeMax}$ **then** break
if $\mathit{capacityS}_{\mathit{NBS}} \geq d_i$ **then**
$\mathit{overlapOwnS} = \text{overlap}(\mathit{nbUser}, S_{\mathit{NBS}}, x_1, x_2, y_1, y_2)$
if $\mathit{overlapOwnS} = 0$ **then** $\mathit{slack} \leftarrow \mathit{totalSlack}$
for $\mathit{otherS} = 1, \ldots, \mathit{lastS}$ **do**
if $\mathit{otherS} \neq \mathit{NBS}$ **then**
$\mathit{overlapOtherS} = \text{overlap}(\mathit{nbUser}, S_{\mathit{otherS}}, x_1, x_2, y_1, y_2)$
$\mathit{addInterference} = \text{interference}(\mathit{nbUser}, i, d_i, \mathit{overlapOtherS}, \delta_{ij})$
$\mathit{slack} \leftarrow \mathit{slack} - \mathit{addInterference}$
end
end
if $\mathit{slack} > 0$ **then**
if $\mathit{slack}_i > \mathit{maxSlack}$ **then**
if $\mathit{c6check\ PASS}$ **then**
update $\mathit{allocationResult}$
$\mathit{slackResult} \leftarrow \mathit{slack}$
$\mathit{maxSlack} \leftarrow \mathit{slack}_i$
end
end
end
end
end
end
if $\mathit{allocationResult} \neq 0$ **then** break
end
if $\mathit{allocationResult} \neq 0$ **then** break
end
if $\mathit{allotionResult} \neq 0$ **then** update $\mathit{c6check}$ **else** $\mathit{slackResult} \leftarrow \mathit{totalSlack}$

Algorithm 7: Minimum bandwidth

Chapter 5

Input: *nbUser*, $S_{NBS}, x_1, x_2, y_1, y_2$
Output: *overlapResult*

for $i = y_1, \ldots, y_2$ **do**
  for $j = x_1, \ldots, x_2$ **do**
    if $S(i, j) \neq 0$ **then**
      $\textit{overlapResult}(S(i, j), 1) = 1$
      $\textit{overlapResult}(S(i, j), 2) \leftarrow \textit{overlapResult}(S(i, j), 2) + 1$
    end
  end
end
if *there is at least an overlap* **then**
  for $i = 1, \ldots,$ *height of* S **do**
    for $j = 1, \ldots,$ *width of* S **do**
      $\textit{overlapResult}(S(i, j), 3) \leftarrow \textit{overlapResult}(S(i, j), 3) + 1$
    end
  end
end

Algorithm 8: Overlap function

Input: *nbUser*, i, d_i, overlap, δ_{ij}
Output: *add_interference*

add_interference $\leftarrow 0$
for $j = 1, \ldots,$ *nbUser* **do**
  if $\textit{overlap}(j, 1) = 1$ **then**
    $\textit{add\_interference}(j) \leftarrow \delta(j, i) \cdot \mathrm{overlap}(j, 2)/\textit{overlap}(j, 3)$
    $\textit{add\_interference}(i) \leftarrow \textit{add\_interference}(i) + \delta(i, j) \cdot \textit{overlap}(j, 2)/d_i$
  end
end

Algorithm 9: Interference function

Input: *userAlloation*, *CsN*, *EIRPTerm*, *PCMargin*
Output: *powerMargin*, *powerAdjustment*, α_i, δ_{ij}, *finalTotalSlack*

powerMargin, *powerAdjustment* $\leftarrow 0, \forall i = 1, \ldots,$ *nbUser*
calculate *CsNplusI_dB*, *CsN_dB*
powerMargin $\leftarrow$ *CsNplusI_dB* $-$ *CsN_dB*
powerAdjustment $\leftarrow max(0,$floor$($*CsNplusI_dB* $-$ *CsN_dB*$-$*PCMargin*$)$
if *powerAdjustment* $\neq 0$ **then**
    while *powerAdjustment* $\neq 0$ **do**
        EIRPTerm $\leftarrow$ *EIRPTerm* $-$ *powerAdjustment*
        update *CsNplusI_dB*
        powerMargin $\leftarrow$ *CsNplusI_dB* $-$ *CsN_dB*
        powerAdjustment $\leftarrow max(0,$floor$($*CsNplusI_dB* $-$ *CsN_dB*$-$*PCMargin*$)$
    end
end
calculate α_i, δ_{ij}, *finalTotalSlack*

Algorithm 10: Uplink power control

Part III

Conclusion

Conclusion and perspectives

We have solved frequency assignment problems in an SDMA satellite communication system by considering intra-system co-channel interference between users. Within this interference-limited environment, our objective is to assign as many users as possible using a short calculation time. Two types of interference are considered i.e. binary and cumulative interference. The latter, though more complicated, yields a more realistic representation. For each of them, single carrier and multiple carrier frequency assignment models are taken into account. We also dealt with 2-dimensional frequency $\times$ time assignments which is more complicated and harder to solve.

Single carrier FAPs are solved by greedy algorithms and ILP. The proposed greedy algorithm is efficient and can provide good quality solutions. We have solved a non-linear algorithm called Beam Moving problem to further improve the solutions from both ILP and greedy algorithm; nonetheless, its calculation time is long. To improve these results, a fast heuristic to solve the continuous optimization problem could be designed. Furthermore, an integrated approach where frequency assignment and beam position are determined simultaneously could be proposed. This yields highly complex mixed non-linear integer programming formulations. More improvements could be achieved by allowing temporary decrease of the objective functions via metaheuristics framework such as tabu search. Better upper bounding techniques could also be helpful to stop the search earlier.

In this thesis, we have also established connections between FAP and scheduling theory. More precisely, multiple carrier FAPs are modelled as scheduling problem and ILPs. We have shown that the scheduling model offers superior performance than the proposed ILP. It is worth noting that, by transforming the cumulative interference into binary interference, scheduling method together with clique-induced constraints yields much better results.

This cumulative-to-binary transformation yields easier problem and will open us to more solving possibilities. In the thesis, we propose a simple transformation based on the average value and a loading factor. This simple transformation tends to overload the graph. A better transformation could help improve the overall performance. Instead of relying only on the average value, statistical analysis could be applied.

We have not tested all the proposed ILP formulations especially the pattern-based one which lead to an interaction with column generation method, which could greatly improve the performance. Heuristic based on interval graph coloring, though not covered here, could also provide fast solving speed. It would be interesting to extended the graph coloring model to cope with cumulative interference or to apply the graph coloring methods to the cumulative-transformed binary interference. Thus, both pattern-based ILP and interval graph coloring would be interesting topics to explore further.

The problem could be also treated as 2-dimensional bin packing problem with constraints on overlapping between users represented by rectangular boxes.

We have also proposed an ILP formulation for 2-dimensional frequency assignment problem. This complex model would need long calculation time. Nonetheless, it would be interesting if we could combine this with a local search technique in order to improve it.

The recently proposed hyper-heuristic [30] method could also be applied to this problem. The key is to define a common local search that can be used by different heuristics. Starting from a solution that could be constructed by a simple greedy algorithm, one could deal simultaneously with both frequency and time or by fixing the time and explore the frequency domain or vice versa.

We also consider the frequency assignment problem which incorporates the specifications and constraints provided by the industry. These requirements render the resource allocation problem highly complex. This complexity and the fact that frequency assignment plans must be recomputed frequently in order to cope for user mobility yield classic optimization tool such as Integer Linear Programming impractical. According to this, we thus proposed greedy algorithms, although it does not provide an optimal solution. Two greedy algorithms are devised and tested. These algorithms are based on two different concepts: minimizing interference or minimizing bandwidth.

When the user demand is reasonable, the proposed greedy algorithms obtain a user acceptance rate that has been judged as satisfactory by the industrial partner. Nevertheless, when the problem becomes highly constrained, especially when the demand increases or when the available bandwidth is limited the performance dramatically decrease. Future work could be focused on computing upper bounds on the number of accepted user to be able to estimate the possible performance gain in highly constrained environment. Then local search heuristics could be proposed to further improve the greedy algorithm.

So far we have not considered the feature that supports temporal demand changes or adding/removing of users. Unlike in the cellular network that considers temporal traffic volume changes to support a number of users either by adding more frequency (each supports multiple users) or considering the worst case interference, change of traffic to the satellite users (ground stations) mean adding or removing frequencies or modifying the assigned time which, in all cases, should strictly respect the resulted signal-to-noise ratio of each connection. In other words, demand change of a satellite user impact other users and resource reassignment would be inevitable. Regarding to this, one would consider a fast local optimization algorithm which could provide resource reassignment to the impacted users. The local optimization could also be used to support mobile users.

Part IV

Appendix

A RsModCod tables

Table A.1: RsModCod Table for Terminal Type 1

No.	ModCod	Required C/N (dB)	BW (MHz)	Symbol Rate (Msymb/s)	Bitrate (Mb/s)	Estimated C/(N+I) (dB)
1.	S2-QPSK1s4	-1.32	5	4.167	1.994	9.065
2.	S2-QPSK1s3	-0.7	5	4.167	2.670	9.065
3.	S2-QPSK2s5	0.25	5	4.167	3.211	9.065
4.	S2-QPSK1s2	1.56	5	4.167	4.022	9.065
5.	S2-QPSK3s5	2.81	5	4.167	4.833	9.065
6.	S2-QPSK2s3	3.7	5	4.167	5.378	9.065
7.	S2-QPSK3s4	4.65	5	4.167	6.050	9.065
8.	S2-QPSK4s5	5.32	5	4.167	6.456	9.065
9.	S2-QPSK5s6	5.84	5	4.167	6.730	9.065
10.	S2-QPSK8s9	6.95	5	4.167	7.185	9.065
11.	S2-QPSK9s10	7.32	5	4.167	7.275	9.065
12.	S2-8PSK3s5	7.57	5	4.167	7.248	9.065
13.	S2-8PSK2s3	7.84	5	4.167	8.065	9.065
14.	S2-8PSK3s4	9.21	5	4.167	9.073	9.065
15.	S2-8PSK5s6	10.78	5	4.167	10.093	9.065
16.	S2-8PSK8s9	12.28	5	4.167	10.775	9.065
17.	S2-QPSK1s4	-1.32	10	8.333	3.988	7.068
18.	S2-QPSK1s3	-0.7	10	8.333	5.340	7.068
19.	S2-QPSK2s5	0.25	10	8.333	6.422	7.068
20.	S2-QPSK1s2	1.56	10	8.333	8.044	7.068
21.	S2-QPSK3s5	2.81	10	8.333	9.667	7.068

Table A.1: RsModCod Table for Terminal Type 1 (continued)

No.	ModCod	Required C/N (dB)	BW (MHz)	Symbol Rate (Msymb/s)	Bitrate (Mb/s)	Estimated C/(N+I) (dB)
22.	S2-QPSK2s3	3.7	10	8.333	10.757	7.068
23.	S2-QPSK3s4	4.65	10	8.333	12.101	7.068
24.	S2-QPSK4s5	5.32	10	8.333	12.912	7.068
25.	S2-QPSK5s6	5.84	10	8.333	13.461	7.068
26.	S2-QPSK8s9	6.95	10	8.333	14.370	7.068
27.	S2-QPSK9s10	7.32	10	8.333	14.550	7.068
28.	S2-8PSK3s5	7.57	10	8.333	14.496	7.068
29.	S2-8PSK2s3	7.84	10	8.333	16.130	7.068
30.	S2-8PSK3s4	9.21	10	8.333	18.146	7.068
31.	S2-8PSK5s6	10.78	10	8.333	20.186	7.068
32.	S2-8PSK8s9	12.28	10	8.333	21.549	7.068
33.	S2-QPSK1s4	-1.32	15	12.500	5.982	5.705
34.	S2-QPSK1s3	-0.7	15	12.500	8.010	5.705
35.	S2-QPSK2s5	0.25	15	12.500	9.633	5.705
36.	S2-QPSK1s2	1.56	15	12.500	12.067	5.705
37.	S2-QPSK3s5	2.81	15	12.500	14.500	5.705
38.	S2-QPSK2s3	3.7	15	12.500	16.135	5.705
39.	S2-QPSK3s4	4.65	15	12.500	18.151	5.705
40.	S2-QPSK4s5	5.32	15	12.500	19.368	5.705
41.	S2-QPSK5s6	5.84	15	12.500	20.191	5.705
42.	S2-QPSK8s9	6.95	15	12.500	21.555	5.705
43.	S2-QPSK9s10	7.32	15	12.500	21.826	5.705
44.	S2-8PSK3s5	7.57	15	12.500	21.745	5.705
45.	S2-8PSK2s3	7.84	15	12.500	24.196	5.705
46.	S2-8PSK3s4	9.21	15	12.500	27.219	5.705
47.	S2-8PSK5s6	10.78	15	12.500	30.278	5.705
48.	S2-8PSK8s9	12.28	15	12.500	32.324	5.705
49.	S2-QPSK1s4	-1.32	20	16.667	7.976	4.669
50.	S2-QPSK1s3	-0.7	20	16.667	10.680	4.669
51.	S2-QPSK2s5	0.25	20	16.667	12.844	4.669
52.	S2-QPSK1s2	1.56	20	16.667	16.089	4.669
53.	S2-QPSK3s5	2.81	20	16.667	19.334	4.669
54.	S2-QPSK2s3	3.7	20	16.667	21.513	4.669
55.	S2-QPSK3s4	4.65	20	16.667	24.201	4.669
56.	S2-QPSK4s5	5.32	20	16.667	25.824	4.669
57.	S2-QPSK5s6	5.84	20	16.667	26.921	4.669

Table A.1: RsModCod Table for Terminal Type 1 (continued)

No.	ModCod	Required C/N (dB)	BW (MHz)	Symbol Rate (Msymb/s)	Bitrate (Mb/s)	Estimated C/(N+I) (dB)
58.	S2-QPSK8s9	6.95	20	16.667	28.740	4.669
59.	S2-QPSK9s10	7.32	20	16.667	29.101	4.669
60.	S2-8PSK3s5	7.57	20	16.667	28.993	4.669
61.	S2-8PSK2s3	7.84	20	16.667	32.261	4.669
62.	S2-8PSK3s4	9.21	20	16.667	36.292	4.669
63.	S2-8PSK5s6	10.78	20	16.667	40.371	4.669
64.	S2-8PSK8s9	12.28	20	16.667	43.099	4.669

Table A.2: RsModCod Table for Terminal Type 2

No.	ModCod	Required C/N (dB)	BW (MHz)	Symbol Rate (Msymb/s)	Bitrate (Mb/s)	Estimated C/(N+I) (dB)
1.	S2-QPSK1s4	-1.32	5	4.167	1.994	5.519
2.	S2-QPSK1s3	-0.7	5	4.167	2.670	5.519
3.	S2-QPSK2s5	0.25	5	4.167	3.211	5.519
4.	S2-QPSK1s2	1.56	5	4.167	4.022	5.519
5.	S2-QPSK3s5	2.81	5	4.167	4.833	5.519
6.	S2-QPSK2s3	3.7	5	4.167	5.378	5.519
7.	S2-QPSK3s4	4.65	5	4.167	6.050	5.519
8.	S2-QPSK4s5	5.32	5	4.167	6.456	5.519
9.	S2-QPSK5s6	5.84	5	4.167	6.730	5.519
10.	S2-QPSK8s9	6.95	5	4.167	7.185	5.519
11.	S2-QPSK9s10	7.32	5	4.167	7.275	5.519
12.	S2-8PSK3s5	7.57	5	4.167	7.248	5.519
13.	S2-8PSK2s3	7.84	5	4.167	8.065	5.519
14.	S2-8PSK3s4	9.21	5	4.167	9.073	5.519
15.	S2-8PSK5s6	10.78	5	4.167	10.093	5.519
16.	S2-8PSK8s9	12.28	5	4.167	10.775	5.519
17.	S2-QPSK1s4	-1.32	10	8.333	3.988	2.928
18.	S2-QPSK1s3	-0.7	10	8.333	5.340	2.928
19.	S2-QPSK2s5	0.25	10	8.333	6.422	2.928
20.	S2-QPSK1s2	1.56	10	8.333	8.044	2.928
21.	S2-QPSK3s5	2.81	10	8.333	9.667	2.928
22.	S2-QPSK2s3	3.7	10	8.333	10.757	2.928

Table A.2: RsModCod Table for Terminal Type 2 (continued)

No.	ModCod	Required C/N (dB)	BW (MHz)	Symbol Rate (Msymb/s)	Bitrate (Mb/s)	Estimated C/(N+I) (dB)
23.	S2-QPSK3s4	4.65	10	8.333	12.101	2.928
24.	S2-QPSK4s5	5.32	10	8.333	12.912	2.928
25.	S2-QPSK5s6	5.84	10	8.333	13.461	2.928
26.	S2-QPSK8s9	6.95	10	8.333	14.370	2.928
27.	S2-QPSK9s10	7.32	10	8.333	14.550	2.928
28.	S2-8PSK3s5	7.57	10	8.333	14.496	2.928
29.	S2-8PSK2s3	7.84	10	8.333	16.130	2.928
30.	S2-8PSK3s4	9.21	10	8.333	18.146	2.928
31.	S2-8PSK5s6	10.78	10	8.333	20.186	2.928
32.	S2-8PSK8s9	12.28	10	8.333	21.549	2.928
33.	S2-QPSK1s4	-1.32	15	12.500	5.982	1.316
34.	S2-QPSK1s3	-0.7	15	12.500	8.010	1.316
35.	S2-QPSK2s5	0.25	15	12.500	9.633	1.316
36.	S2-QPSK1s2	1.56	15	12.500	12.067	1.316
37.	S2-QPSK3s5	2.81	15	12.500	14.500	1.316
38.	S2-QPSK2s3	3.7	15	12.500	16.135	1.316
39.	S2-QPSK3s4	4.65	15	12.500	18.151	1.316
40.	S2-QPSK4s5	5.32	15	12.500	19.368	1.316
41.	S2-QPSK5s6	5.84	15	12.500	20.191	1.316
42.	S2-QPSK8s9	6.95	15	12.500	21.555	1.316
43.	S2-QPSK9s10	7.32	15	12.500	21.826	1.316
44.	S2-8PSK3s5	7.57	15	12.500	21.745	1.316
45.	S2-8PSK2s3	7.84	15	12.500	24.196	1.316
46.	S2-8PSK3s4	9.21	15	12.500	27.219	1.316
47.	S2-8PSK5s6	10.78	15	12.500	30.278	1.316
48.	S2-8PSK8s9	12.28	15	12.500	32.324	1.316
49.	S2-QPSK1s4	-1.32	20	16.667	7.976	0.143
50.	S2-QPSK1s3	-0.7	20	16.667	10.680	0.143
51.	S2-QPSK2s5	0.25	20	16.667	12.844	0.143
52.	S2-QPSK1s2	1.56	20	16.667	16.089	0.143
53.	S2-QPSK3s5	2.81	20	16.667	19.334	0.143
54.	S2-QPSK2s3	3.7	20	16.667	21.513	0.143
55.	S2-QPSK3s4	4.65	20	16.667	24.201	0.143
56.	S2-QPSK4s5	5.32	20	16.667	25.824	0.143
57.	S2-QPSK5s6	5.84	20	16.667	26.921	0.143
58.	S2-QPSK8s9	6.95	20	16.667	28.740	0.143

Table A.2: RsModCod Table for Terminal Type 2 (continued)

No.	ModCod	Required C/N (dB)	BW (MHz)	Symbol Rate (Msymb/s)	Bitrate (Mb/s)	Estimated C/(N+I) (dB)
59.	S2-QPSK9s10	7.32	20	16.667	29.101	0.143
60.	S2-8PSK3s5	7.57	20	16.667	28.993	0.143
61.	S2-8PSK2s3	7.84	20	16.667	32.261	0.143
62.	S2-8PSK3s4	9.21	20	16.667	36.292	0.143
63.	S2-8PSK5s6	10.78	20	16.667	40.371	0.143
64.	S2-8PSK8s9	12.28	20	16.667	43.099	0.143

System and Link parameters

The system and link parameters used in the experiment are listed below:

Effective Isotropic Radiated Power (EIRPTerm) of Terminal Type 1 and 2	50dBW, 45dBW
Antenna equivalent temperature (TA)	300K
Repeater equivalent temperature (TRep)	500K
Uplink frequency (FUp)	8.4GHz
Uplink free space loss (LFSLUp)	203dB
Uplink atmospheric loss (LAtmoUp)	3.2dB
Intermodulation product signal to noise ratio (CsIM)	20dB
Feeder link signal to interference ratio (CsIFeeder)	100dB
Feeder link signal to noise ratio (CsNFeeder)	20dB
LDepTerm	1
Antenna efficiency (eta)	0.8
Antenna diameter (D)	1m
Primary source diameter (d)	0.13m
Maximum number of satellite beams (NbBeamsMax)	30
Superframe bandwidth (BWAvail)	60MHz, 100MHz
System bandwidth (BWMax)	300MHz
Frame's minimum frequency size (FrameBWMin)	1MHz
Frame's maximum frequency size (FrameBWMax)	30MHz
Frame duration	0.1s
Slot duration	0.01s
Power control margin (PCMargin)	2dB

C++ code for scheduling using CP Optimizer

B

1 Multiple carrier with binary interference

```
/* Find maximum profit (number of assigned operations)
 *
 * This is a scheduling problem equivalent to a frequency assignment problem.
 *
 * Inputs are an interference coefficient matrix (delta) and
 * a demand vector of each user.
 *
 * The problem is to assign each user a range of frequency in that the interfering users
 * do not overlap and the overall frequency usage is minimum.
 *
 * The problem is reduced to scheduling problem by representing each user as
 * an operation, demand as a duration and frequency as time.
 * Each maximal clique corresponds to a set of disjunctive operations in which overlapping
 * of users are not allowed.
 *
 */

#include <ilcp/cp.h>
#include <stdlib.h>
#include <stdlib.h>
#include <iostream>
#include <fstream>
using namespace std;
#include <vector>
#include <string>
#include <algorithm>

class FileError: public IloException {
  public:
  FileError() : IloException("Cannot open data file") {}
};

int main(int argc, const char* argv[]) {
  IloEnv env;
  try {
    IloInt nbUser;
    IloInt instNb;
    IloNum loadingFactor;
    IloInt timeLimit = 60;
    IloInt bandwidth;

    if (argc > 1)
      nbUser = atoi(argv[1]);
    if (argc > 2)
      instNb = atoi(argv[2]);
    if (argc > 3)
      loadingFactor = atof(argv[3]);
    if (argc > 4)
      bandwidth = atoi(argv[4]);
    if (argc > 5)
      timeLimit = atoi(argv[5]);
```

```
// open the delta file
char fileDelta[80];
sprintf(fileDelta,"PL_data/delta_scenar3_%d_%d.txt",nbUser,instNb);
ifstream fileA(fileDelta);
IloNum sumDelta = 0.0;
IloNum avgDelta = 0.0;
IloInt countDelta = 0;
IloNumArray2 delta(env, nbUser);
IloIntArray2 newDelta(env, nbUser);
for (IloInt i = 0; i < nbUser; i++) {
  delta[i] = IloNumArray(env, nbUser);
  newDelta[i] = IloIntArray(env, nbUser);
  for (IloInt j = 0; j < nbUser; j++) {
    fileA >> delta[i][j];
    delta[i][j] *= 1e20;
    sumDelta += delta[i][j];
    if (delta[i][j] > 0)
      countDelta ++;
  }
}
fileA.close();
avgDelta = (loadingFactor * sumDelta) / countDelta;
for (IloInt i = 0; i < nbUser; i++)
  for (IloInt j = 0; j < nbUser; j++) {
    if (i == j) {
      newDelta[i][j] = 0;
      newDelta[j][i] = 0;
    }
    else if ( (j > i) && (delta[i][j] < avgDelta) ) {
      newDelta[i][j] = 0;
      newDelta[j][i] = 0;
    }
    else if ( (j > i) && (delta[i][j] >= avgDelta) ) {
      newDelta[i][j] = 1;
      newDelta[j][i] = 1;
   }
}

if (!fileA) {
  env.out() << "usage: " << argv[0] << " <fileA>" << endl;
  throw FileError();
}

// open the demand file
char fileDemand[80];
sprintf(fileDemand,"PL_data/Demand_%d_%d.txt",nbUser,instNb);
ifstream fileB(fileDemand);
IloNumArray demand(env, nbUser);
for (IloInt i = 0; i < nbUser; i++)
  fileB >> demand[i];
fileB.close();

if (!fileB) {
  env.out() << "usage: " << argv[0] << " <fileB>" << endl;
  throw FileError();
}

// open the alpha file
char fileAlpha[80];
sprintf(fileAlpha,"PL_data/alpha_scenar3_%d_%d.txt",nbUser,instNb);
ifstream fileD(fileAlpha);
IloNumArray alpha(env, nbUser);
IloNum max_alpha = -IloInfinity;
IloNum min_alpha = IloInfinity;
for (IloInt i=0; i < nbUser; i++) {
  fileD >> alpha[i];
  alpha[i] *= 1e20;
  if (alpha[i] < min_alpha)
    min_alpha = alpha[i];
  if (alpha[i] > max_alpha)
    max_alpha = alpha[i];
}
fileD.close();

if (!fileD) {
  env.out() << "usage: " << argv[0] << " <fileD>" << endl;
  throw FileError();
}

// start modeling
IloModel model(env);

// define operations
IloIntervalVarArray operations(env, nbUser);
IloIntExpr profit(env);
for (IloInt i = 0; i < nbUser; i++) {
```

```
    operations[i] = IloIntervalVar(env, demand[i]);
    operations[i].setOptional();
    operations[i].setEndMax(bandwidth);
    profit += IloPresenceOf(env, operations[i]);
  }

  // create non-overlapping constraints
  for (IloInt i = 0; i < nbUser; i++) {
    for (IloInt j = 0; j < nbUser; j++) {
      if ( (j > i) && (newDelta[i][j] == 1) ) {
        IloIntervalVarArray disjunctive(env);
        disjunctive.add(operations[i]);
        disjunctive.add(operations[j]);
        model.add(IloNoOverlap(env, disjunctive));
      }
    }
  }

  // create model objective
  IloObjective objective = IloMaximize(env, profit);
  model.add(objective);

  // solve the model and output the solution
  IloCP cp(model);
  cp.setParameter(IloCP::LogVerbosity, IloCP::Quiet);
  cp.setParameter(IloCP::TimeLimit, timeLimit);

  if (cp.solve()) {
    char namefileout[80];
    sprintf(namefileout,"Output6_%d_%d.txt",nbUser,instNb);
    ofstream fileout(namefileout);
    fileout << "Number of assigned operations \t: " << cp.getValue(objective) << std::endl;
    int opt;
    if (cp.getStatus() == IloAlgorithm::Optimal)
      opt = 1;
    else if (cp.getStatus() == IloAlgorithm::Feasible)
      opt = 2;
    else if (cp.getStatus() == IloAlgorithm::Infeasible)
      opt = 3;
    else if (cp.getStatus() == IloAlgorithm::Unbounded)
      opt = 4;
    else if (cp.getStatus() == IloAlgorithm::Unknown)
      opt = 5;
    else
      opt = 0;

    fileout << "Opt \t: " << opt << std::endl;
    cp.printInformation(fileout);
    for (IloInt i = 0; i < nbUser; i++) {
      if (cp.isPresent(operations[i])) {
        fileout << "Operation " << i << "\t: " << cp.getStart(operations[i]) << "\t" << cp.getEnd(operations[i]) << endl;
      }
    }

   // check constraints
   IloBool check = true;
   IloInt countViolation = 0;
   fileout << "checking constraints" << endl;
   for (IloInt i = 0; i < nbUser; i++) {
     if (cp.isPresent(operations[i])) {
       fileout << "constr " << i << ": " ;
       IloNum slack = 0.0;
       for (IloInt j = 0; j < nbUser; j++) if (i!=j) {
         if (cp.isPresent(operations[j])) {
           int overlap = 0;
           if (cp.getStart(operations[i]) <= cp.getEnd(operations[j]) -1 &&
               cp.getStart(operations[j]) <= cp.getEnd(operations[i]) -1)
             overlap = min(cp.getEnd(operations[i]),cp.getEnd(operations[j]))
                       -max(cp.getStart(operations[i]),cp.getStart(operations[j]));
           slack += overlap * delta[i][j] / demand[i];
         }
       }
       fileout << "slack = " << slack << "<=" << alpha[i];
       if (slack <= alpha[i])
         fileout << " OK" << endl;
       else {
         check = false;
         fileout << " VIOLATED" << endl;
         countViolation ++;
       }
     }
   }
   fileout.close();
   out << nbUser << "\t" << instNb << "\t" << cp.getValue(objective) << "\t" << opt << "\t"
   << cp.getInfo(IloCP::SolveTime) << "\t" << countViolation << endl;
 }
 else {
```

```
     cp.out() << "No solution found."  << std::endl;
   }
 }

 catch(IloException& e) {
  env.out() << " ERROR: " << e << std::endl;
 }
 env.end();
 return 0;
}
```

2 Multiple carrier with cumulative interference

```
/* Find maximum number of assigned operations
 *
 * This is a scheduling problem reduced from a frequency assignment problem.
 *
 * Inputs are an interference matrix delta, a threshold matrix alpha and
 * a demand vector of each user.
 *
 * The problem is to assign each user a range of frequency in that the interfering users are allowed in that
 * the interference does not exceed the threshold and the overall frequency usage is minimum.
 *
 * The problem is reduced to scheduling problem by representing each user as an operation,
 * the demand as a time interval, the frequency resource as time.
 * Each maximal clique corresponds to a set of disjunctive operations in which overlapping of users are not allowed.
 *
 */

#include <ilcp/cp.h>
#include <stdlib.h>
#include <stdlib.h>
#include <iostream>
#include <fstream>
using namespace std;
#include <vector>
#include <string>
#include <algorithm>

class FileError: public IloException {
  public:
  FileError() : IloException("Cannot open data file") {}
};

int main(int argc, const char* argv[]) {
  IloEnv env;
    try {
      IloInt nbUser;
      IloInt instNb;
      IloInt timeLimit = 60;
      IloInt bandwidth;

      if (argc > 1)
        nbUser = atoi(argv[1]);
      if (argc > 2)
        instNb = atoi(argv[2]);
      if (argc > 3)
        bandwidth = atoi(argv[3]);
      if (argc > 4)
        timeLimit = atoi(argv[4]);

      // open the alpha file
      char fileAlpha[80];
      sprintf(fileAlpha,"PL_data/alpha_scenar3_%d_%d.txt",nbUser,instNb);
      ifstream fileA(fileAlpha);
      IloNumArray alpha(env, nbUser);
      IloNum max_alpha = -IloInfinity;
      IloNum min_alpha = IloInfinity;
      for (IloInt i=0; i < nbUser; i++) {
        fileA >> alpha[i];
        alpha[i] *= 1e20;
          if (alpha[i] < min_alpha)
            min_alpha = alpha[i];
          if (alpha[i] > max_alpha)
            max_alpha = alpha[i];
      }
      fileA.close();

      if (!fileA) {
        env.out() << "usage: " << argv[0] << " <fileAlpha>" << endl;
        throw FileError();
```

```
}

// open the delta file
char fileDelta[80];
sprintf(fileDelta,"PL_data/delta_scenar3_%d_%d.txt",nbUser,instNb);
ifstream fileB(fileDelta);

// read data from fileB to delta
IloNum max_delta = -IloInfinity;
IloNum min_delta = IloInfinity;
IloNumArray2 delta(env, nbUser); // IloNumArray<IloNumArray> delta(env);
for (IloInt i = 0; i < nbUser; i++) {
  delta[i] = IloNumArray(env, nbUser); // delta.add(IloNumArray(env, nbUser);
  for (IloInt j = 0; j < nbUser; j++) {
    fileB >> delta[i][j];
    delta[i][j] *= 1e20;
    if (delta[i][j] < min_delta)
      min_delta = delta[i][j];
    if (delta[i][j] > max_delta)
      max_delta = delta[i][j];
  }
}
fileB.close();

if (!fileB) {
  env.out() << "usage: " << argv[0] << " <fileDelta>" << endl;
  throw FileError();
}

// open the demand file
char fileDemand[80];
sprintf(fileDemand,"PL_data/Demand_%d_%d.txt",nbUser,instNb);
ifstream fileC(fileDemand);

// read data from fileC to demand
IloNumArray demand(env, nbUser);
for (IloInt i = 0; i < nbUser; i++)
  fileC >> demand[i];
fileC.close();

if (!fileC) {
  env.out() << "usage: " << argv[0] << " <fileDemand>" << endl;
  throw FileError();
}

// start modeling
IloModel model(env);

// define operations
IloIntervalVarArray operations(env, nbUser);
IloIntExpr profit(env);
IloInt sum_demand = 0;
for (IloInt i = 0; i < nbUser; i++) {
  operations[i] = IloIntervalVar(env, demand[i]);
  operations[i].setOptional();
  operations[i].setEndMax(bandwidth);
  profit += IloPresenceOf(env, operations[i]);
  sum_demand += demand[i];
}

// create overlapping constraints
IloExprArray slack(env, nbUser);
for (IloInt i = 0; i < nbUser; i++) {
  slack[i] = IloExpr(env);
  for (IloInt j = 0; j < nbUser; j++) {
    if (j != i) {
      if (delta[i][j]>0)
        slack[i] += IloOverlapLength(env, operations[i], operations[j]) * ( delta[i][j] / demand[i] );
    }
  }
  model.add(slack[i] <= alpha[i]);
}

// create model objective
IloObjective objective = IloMaximize(env, profit);
model.add(objective);

// solve the model and output the solution
IloCP cp(model);
cp.setParameter(IloCP::LogVerbosity, IloCP::Quiet);
cp.setParameter(IloCP::TimeLimit, timeLimit);
if (cp.solve()) {
  char namefileout[80];
  sprintf(namefileout,"Output9_%d_%d.txt",nbUser,instNb);
  ofstream fileout(namefileout);
  fileout << "Number of assigned operations \t: " << cp.getValue(objective) << std::endl;
  int opt;
```

```
        if (cp.getStatus() == IloAlgorithm::Optimal)
          opt = 1;
        else if (cp.getStatus() == IloAlgorithm::Feasible)
          opt = 2;
        else if (cp.getStatus() == IloAlgorithm::Infeasible)
          opt = 3;
        else if (cp.getStatus() == IloAlgorithm::Unbounded)
          opt = 4;
        else if (cp.getStatus() == IloAlgorithm::Unknown)
          opt = 5;
        else
          opt = 0;

        fileout << "Opt \t: " << opt << std::endl;
        cp.printInformation(fileout);

        for (IloInt i = 0; i < nbUser; i++) {
          if (cp.isPresent(operations[i])) {
            fileout << "Operation " << i << "\t: " << cp.getStart(operations[i]) << "\t" << cp.getEnd(operations[i]) << endl;
          }
        }

        // check constraints
        IloBool check = true;
        fileout << "checking constraints" << endl;
        for (IloInt i = 0; i < nbUser; i++) {
          if (cp.isPresent(operations[i])) {
            fileout << "constr " << i << ": ";
            IloNum slack = 0.0;
            for (IloInt j = 0; j < nbUser; j++) if (i!=j) {
              if (cp.isPresent(operations[j])) {
                int overlap = 0;
                if (cp.getStart(operations[i]) <= cp.getEnd(operations[j]) -1 &&
                    cp.getStart(operations[j]) <= cp.getEnd(operations[i]) -1)
                  overlap = min(cp.getEnd(operations[i]),cp.getEnd(operations[j]))
                          -max(cp.getStart(operations[i]),cp.getStart(operations[j]));
                slack += overlap*delta[i][j] / demand[i];
              }
            }
            fileout << "slack=" << slack << "<=" << alpha[i];
            if (slack <= alpha[i])
              fileout << " OK" << endl;
            else {
              check = false;
              fileout << " VIOLATED" << endl;
            }
          }
        }
        fileout.close();
        cout << nbUser << "\t" << instNb << "\t" << cp.getValue(objective) << "\t" << opt << "\t"
        << cp.getInfo(IloCP::SolveTime) << "\t" << cp.getInfo(IloCP::NumberOfFails) << "\t" << check << endl;
      }
      else {
        cp.out() << "No solution found."  << std::endl;
      }
    }
    catch(IloException& e) {
      env.out() << " ERROR: " << e << std::endl;
    }
    env.end();
    return 0;
}
```

C

Flow charts for industrial application greedy algorithms

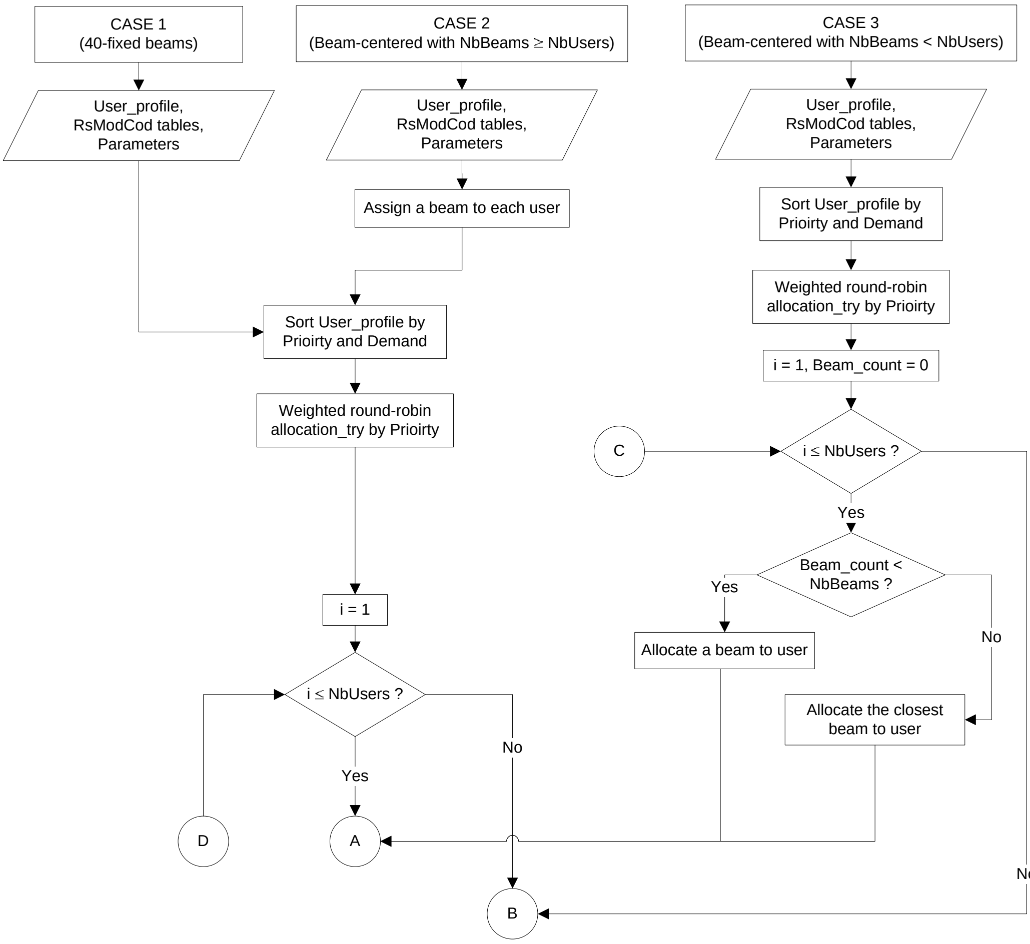


Figure C.1: Allocation cases part 1.

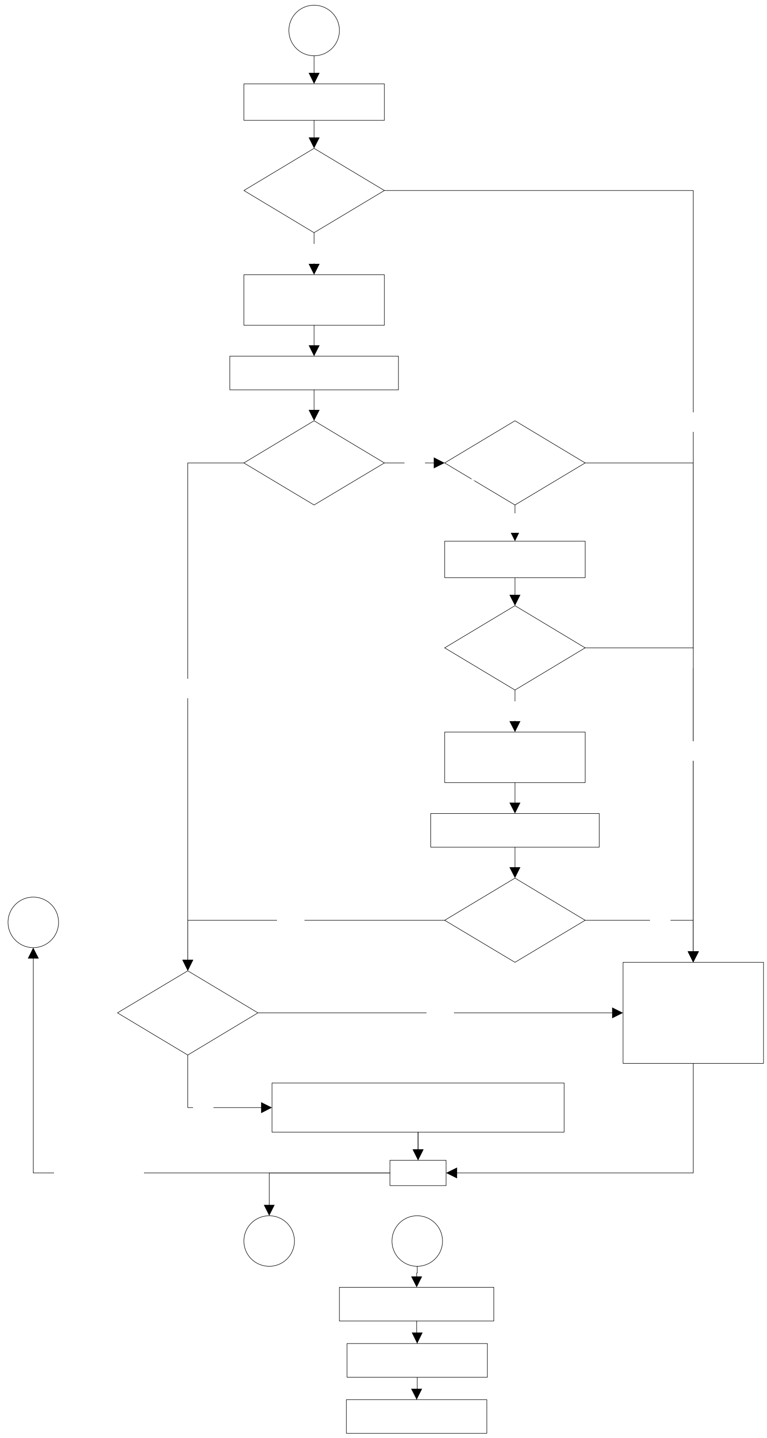
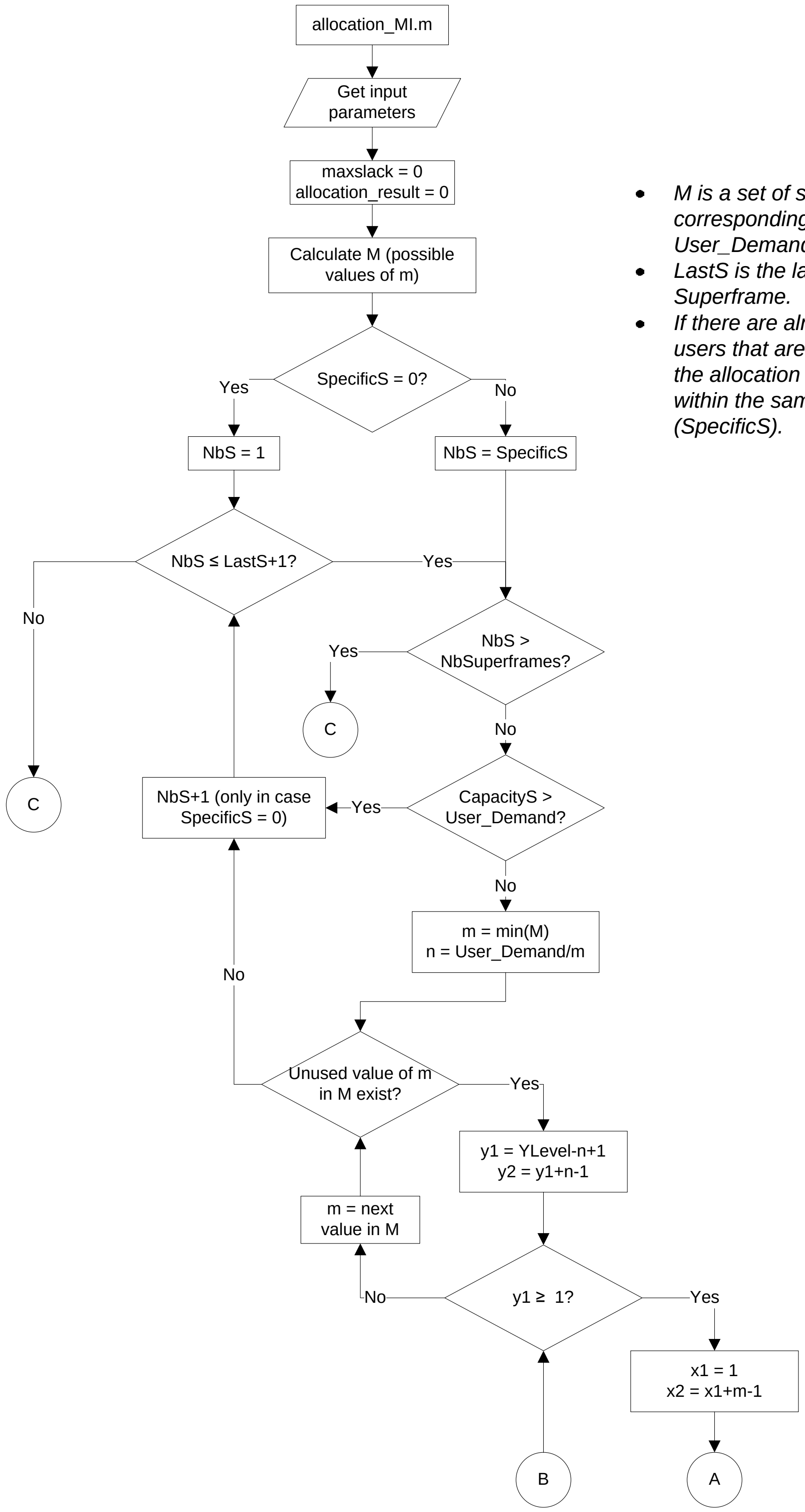


Figure C.2: Allocation cases part 2.



- *M is a set of slot lengths corresponding to the User_Demand*
- *LastS is the lastly allocated Superframe.*
- *If there are already allocated users that are in the same beam, the allocation should only be tried within the same Superframe (SpecificS).*

Figure C.3: Minimum interference part 1.

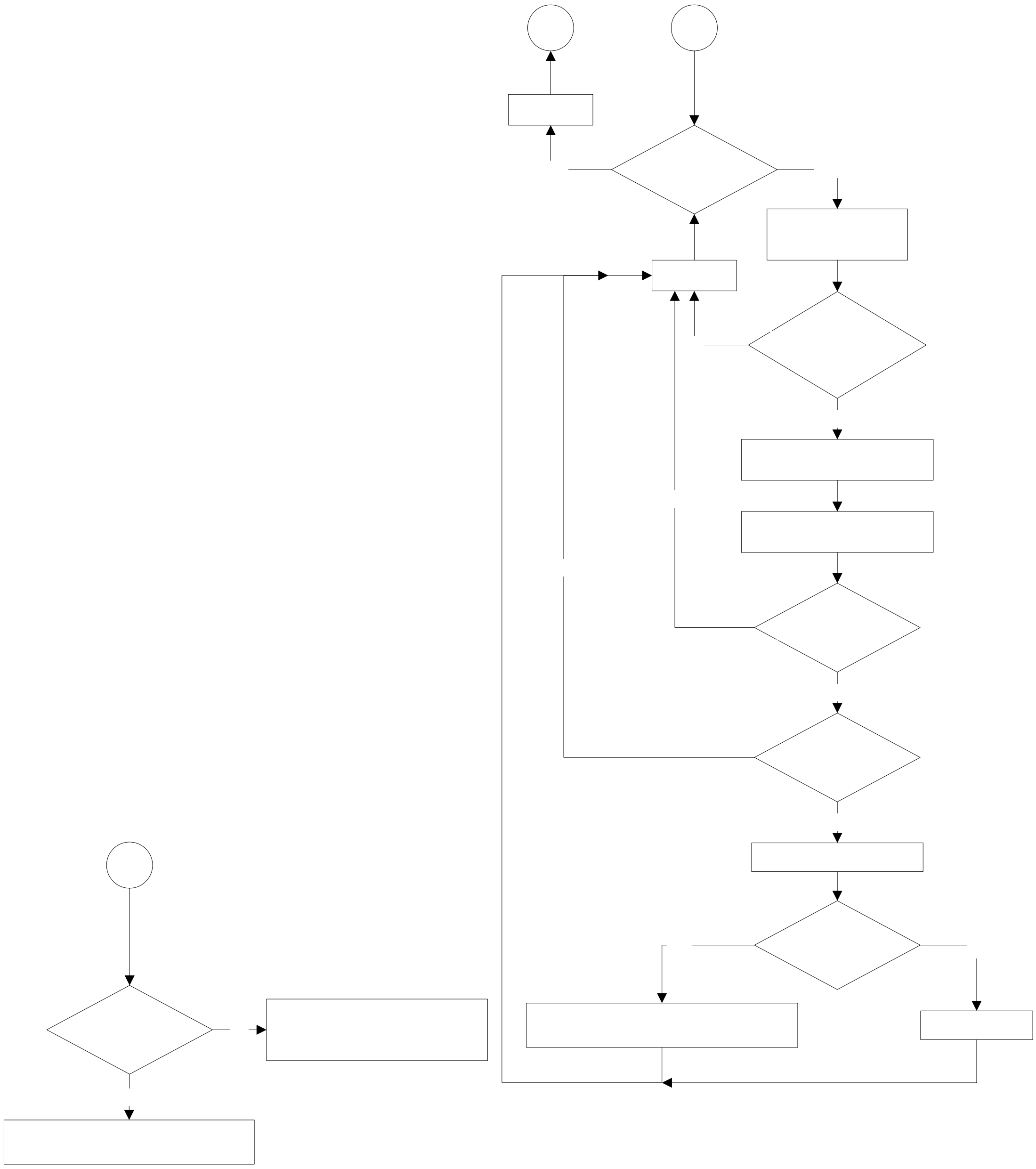


Figure C.4: Minimum interference part 2.

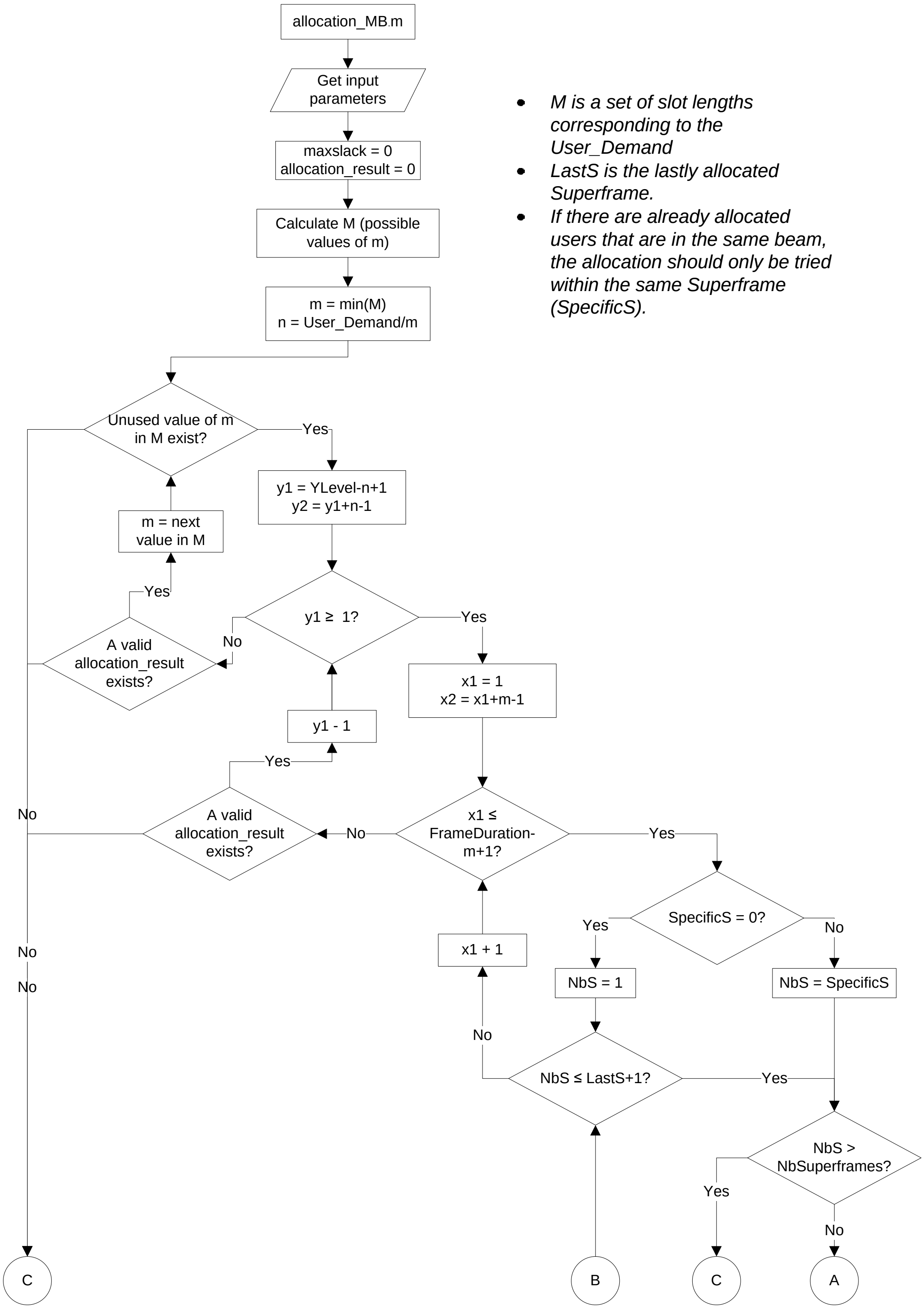


Figure C.5: Minimum bandwidth part 1.

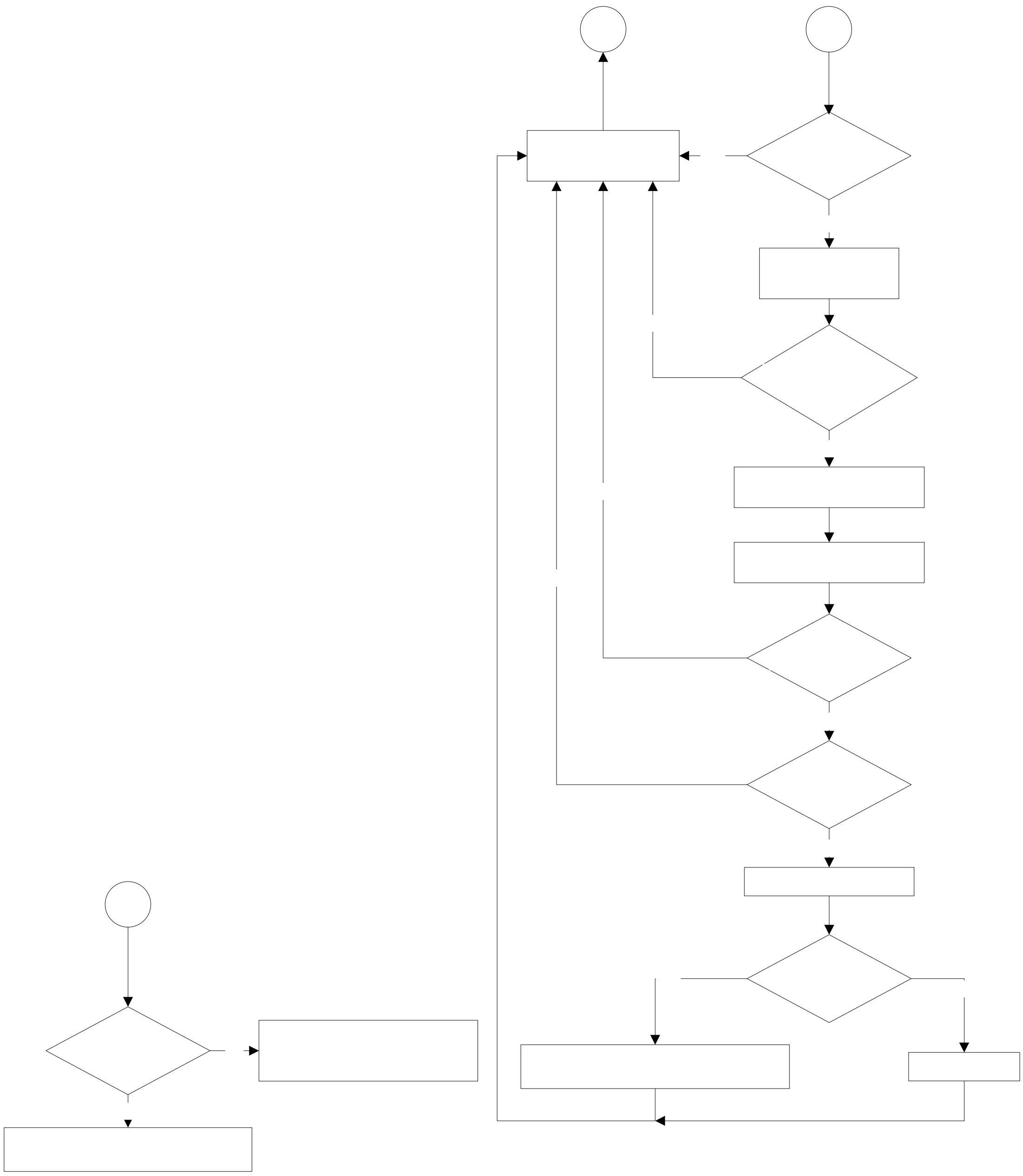


Figure C.6: Minimum bandwidth part 2.

Bibliography

Bibliography

[1] IEEE 802.11b: Higher-Speed Physical Layer (PHY) Extension in the 2.4 Ghz Band, 2001.

[2] ETSI EN 301 790: Digital Video Broadcasting (DVB); Interaction channel for satellite distribution systems, 03 2003.

[3] ETSI EN 302 307: Digital Video Broadcasting (DVB); second generation framing structure, channel coding and modulation systems for broadcasting, interactive services, news gathering and other broadband satellite applications (DVB-S2), 08 2009.

[4] K. I. 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.

[5] Karen Aardal, Stan van Hoesel, Arie Koster, Carlo Mannino, and Antonio Sassano. Models and solution techniques for frequency assignment problems. *Annals of Operations Research*, 153:79–129, 2007. 10.1007/s10479-007-0178-0.

[6] 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, 2000.

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

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

[9] David Allouche, Simon De Givry, and Thomas Schiex. Towards parallel non serial dynamic programming for solving hard weighted csp. In *Proceedings of the 16th international conference on Principles and practice of constraint programming*, CP'10, pages 53–60, Berlin, Heidelberg, 2010. Springer-Verlag.

[10] Sara Alouf, Eitan Altman, Jerome Galtier, Jérôme Galtier, Jean-Francois Lalande, and Corinne Touati. Quasi-optimal bandwidth allocation for multi-spot MFTDMA satellites. In *Proc. of IEEE Infocom 2005*, 2005.

[11] C. Artigues, L. Houssin, and F. Messine. Greedy algorithms for SDMA satellite communication system frequency allocation. Technical Report 12092, LAAS-CNRS, 2012. Rapport de Contrat Thales Alenia Space, INPT, LAAS-CNRS, No. 73688.

[12] A. Avenali, C. Mannino, and A. Sassano. Minimizing the span of d-walks to compute optimum frequency assignments. *Mathematical Programming*, 91(2):357–374, 2002.

[13] Constantine A. Balanis. *Antenna Theory: Analysis and Design*. Wiley-Interscience, 2005.

[14] Shumeet Baluja. Population-based incremental learning: A method for integrating genetic search based function optimization and competitive learning. Technical report, 1994.

[15] P. Baptiste, C. Le Pape, and W. Nuijten. *Constraint-Based Scheduling*. Kluwer Academic Publishers, 2001.

[16] Holger Boche and Andreas Eisenblätter. Mathematics in wireless communications. In Martin Grötschel, Klaus Lucas, and Volker Mehrmann, editors, *Production Factor Mathematics*, pages 157–178. Springer Berlin Heidelberg, 2010.

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

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

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

[20] Sandro Bosio and Di Yuan. Modeling and solving ap location and frequency assignment for maximizing access efficiency in wi-fi networks. Technical Report arXiv:1101.5027, Jan 2011. Comments: In proceedings of International Network Optimization Conference 2009 (INOC 2009), Pisa, Italy, 2009.

[21] Mathieu Bouchard, Mirjana Čangalović, and Alain Hertz. About equivalent interval colorings of weighted graphs. *Discrete Applied Mathematics*, 157(17):3615 – 3624, 2009. Sixth International Conference on Graphs and Optimization 2007.

[22] Mathieu Bouchard, Mirjana Čangalović, and Alain Hertz. On a reduction of the interval coloring problem to a series of bandwidth coloring problems. *Journal of Scheduling*, 13:583–595, 2010. 10.1007/s10951-009-0149-1.

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

[24] A. Bouju, J. F. Boyce, C. H. D. Dimitropoulos, G. Vom Scheidt, J. G. Taylor, A. Likas, G. Papageorgiou, and A. Stafylopatis. Intelligent search for the radio links frequency assignment problem. In *Int. Conf. For Digital Signal Processing (DSP9́5)*, Limassol, Cypres, 1995.

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

[26] Coen Bron and Joep Kerbosch. Algorithm 457: finding all cliques of an undirected graph. *Commun. ACM*, 16(9):575–577, September 1973.

[27] Yves Caseau and François Laburthe. Disjunctive scheduling with task intervals. pages 363–377. MIT Press, 1995.

[28] F. Cazals and C. Karande. A note on the problem of reporting maximal cliques. *Theoretical Computer Science*, 407(1 - 3):564 – 568, 2008.

[29] K.-N. Chang and S. Kim. Channel allocation in cellular radio networks. *Computers and Operations Research*, 24(9):849–860, 1997.

[30] José M. Chaves-González, Miguel A. Vega-Rodríguez, Juan A. Gómez-Pulido, and Juan M. Sánchez-Pérez. Optimizing a realistic large-scale frequency assignment problem using a new parallel evolutionary approach. *Engineering Optimization*, 43(8):813–842, 2011.

[31] Z.N. Chen and K.M. Luk. *Antennas for base stations in wireless communications*. Communication engineering. McGraw-Hill, 2009.

[32] A. T. Clementson and C. H. Elphick. Approximate colouring algorithms for composite graphs. *The Journal of the Operational Research Society*, 34(6):pp. 503–509, 1983.

[33] E. Corbel. Gestion de la ressources radio dans un système utilisant le SDMA. Technical Report RT2E-TAS-MO-641, Thales Alenia Space, 2010.

[34] Thomas H. Cormen, Clifford Stein, Ronald L. Rivest, and Charles E. Leiserson. *Introduction to Algorithms*. McGraw-Hill Higher Education, 2nd edition, 2001.

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

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

[37] C. Crisan and H. Mühlenbein. The frequency assginment problem: A look at the performance of evolutionary search. *Lecture Notes in Computer Science*, 1363:263–274, 1998.

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

[39] D. de Werra and Y. Gay. Chromatic scheduling and frequency assignment. *Discrete Applied Mathematics*, 49(1 - 3):165 – 174, 1994. Special Volume Viewpoints on Optimization.

[40] D. de Werra and A. Hertz. Consecutive colorings of graphs. *Mathematical Methods of Operations Research*, 32:1–8, 1988. 10.1007/BF01920567.

[41] Mohammad Dib, Alexandre Caminada, and Hakim Mabed. A fast algorithm to solve the frequency assignment problem. In Willem-Jan van Hoeve and John Hooker, editors, *Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems*, volume 5547 of *Lecture Notes in Computer Science*, pages 316–318. Springer Berlin / Heidelberg, 2009.

[42] Mohammad Dib, Alexandre Caminada, and Hakim Mabed. Nogood recording with tabu search for CSP (application to FAP). In *Proceedings of the 2009 Third Asia International Conference on Modelling & Simulation*, AMS '09, pages 218–223, Washington, DC, USA, 2009. IEEE Computer Society.

[43] Mohammad Dib, Hakim Mabed, and Alexandre Caminada. Constraint propagation with tabu list for min-span frequency assignment problem. In Hoai An Le Thi, Pascal Bouvry, and Tao Pham Dinh, editors, *Modelling, Computation and Optimization in Information Systems and Management Sciences*, volume 14 of *Communications in Computer and Information Science*, pages 97–106. Springer Berlin Heidelberg, 2008.

[44] Marco Dorigo and Thomas Stützle. *Ant Colony Optimization*. Bradford Company, Scituate, MA, USA, 2004.

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

[46] 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.

[47] N. Dunkin, J. Bater, P. Jeavons, 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.

[48] Audrey Dupont, Andréa Carneiro Linhares, Christian Artigues, Dominique Feillet, Philippe Michelon, and Michel Vasquez. The dynamic frequency assignment problem. *European Journal of Operational Research*, 195(1):75 – 88, 2009.

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

[50] A. Eisenblätter, M. Grötschel, and A. M. C. A. Koster. Frequency assignment and ramifications of coloring. *Discussiones Mathematicae Graph Theory*, 22:51–88, 2002.

[51] J. Elhachimi and Z. Guennoun. Frequency assignment for cellular mobile systems using a hybrid tabu search with an adaptive constraint satisfaction technique. In *Multimedia Computing and Systems (ICMCS), 2011 International Conference on*, pages 1 –5, April 2011.

[52] Jamal Elhachimi and Zouhair Guennoun. Evolutionary neural networks algorithm for the dynamic frequency assignment problem. *International Journal of Computer Science & Information Technology*, 3(3):49–61, 2011.

[53] EUCLID CALMA. CALMA Frequency Assignment Problems. `http://fap.zib.de/problems/CALMA/`.

[54] Thomas A. Feo and Mauricio G. C. Resende. Greedy randomized adaptive search procedures. *Journal of Global Optimization*, 6:109–133, 1995.

[55] M. Fischetti, C. Lepschy, G. Minerva, G. Romanin-Jacur, and E. Toto. Frequency assignment in mobile radio systems using branch-and-cut techniques. *European Journal of Operational Research*, 123:241–255, 2000. previously published as technical report of the Universita di Padova.

[56] Stéphane Foldes and Peter L. Hammer. Split graphs. In *Proceedings of the Eighth Southeastern Conference on Combinatorics, Graph Theory and Computing (Louisiana State Univ., Baton Rouge, La., 1977)*, pages 311–315. Congressus Numerantium, No. XIX, Winnipeg, Man., 1977. Utilitas Math.

[57] N. Funabiki and S. Nishikawa. A gradual neural-network approach for frequency assignment in satellite communication systems. *Neural Networks, IEEE Transactions on*, 8(6):1359 –1370, Nov 1997.

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

[59] Michael R. Garey and David S. Johnson. *Computers and Intractability; A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, NY, USA, 1990.

[60] G. Giambene. *Resource management in satellite networks: optimization and cross-layer design*. Springer, 2007.

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

[62] F. Glover and M. Laguna. *Tabu Search*. Kluwer Academic, Dordrecht, 1997.

[63] Fred Glover, Manuel Laguna, and Rafael Martí. Scatter search. In *Advances in Evolutionary Computing: Theory and Applications*, pages 519–537. Springer-Verlag, 2003.

[64] Martin C. Golumbic. *Algorithmic graph theory and perfect graphs*. Academic Press, 1980.

[65] A. Gondran, O. Baala, A. Caminada, and H. Mabed. Interference management in IEEE 802.11 frequency assignment. In *Vehicular Technology Conference, 2008. VTC Spring 2008. IEEE*, pages 2238 –2242, May 2008.

[66] A. Gondran, O. Baala, H. Mabed, and A. Caminada. Hypergraph t-coloring for automatic frequency planning problem in wireless lan. In *Personal, Indoor and Mobile Radio Communications, 2008. PIMRC 2008. IEEE 19th International Symposium on*, pages 1 –5, sept. 2008.

[67] J. S. Graham, R. Montemanni, J. N. J. Moon, and D. H. Smith. Frequency assignment, multiple interference and binary constraints. *Wireless Networks*, 14(4):449–464, Auguest 2008.

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

[69] Pierre Hansen and Nenad Mladenović. Variable neighborhood search. In Fred Glover and Gary Kochenberger, editors, *Handbook of Metaheuristics*, volume 57 of *International Series in Operations Research & Management Science*, pages 145–184. Springer New York, 2003.

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

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

[72] J.-K. Hao and L. Perrier. Tabu search for the frequency assignment problem in cellular radio networks. Technical Report LGI2P, EMA-EERIE, Parc Scientifique Georges Besse, Nimes, France, 1999.

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

[74] John H. Holland. *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence*. MIT Press, Cambridge, MA, USA, 1992.

[75] D. Hong and S.S. Rappaport. Heuristic channel assignments for cellular land mobile radio systems. In *Proceedings of the IEEE GLOBECOM*, pages 997–1001, 1985.

[76] J.J. Hopfield and D.W. Tank. Neural computation of decisions in optimization problems. *Biological Cybernetics*, 52:141–152, 1985.

[77] L. Houssin and C.Artigues. Modèles mathématiques et algorithmes d'allocation de fréquences de type SDMA appliqué à un système de télécommunications par satellite. Technical Report 08802, LAAS-CNRS, 2008. Rapport de contrat LAAS/Thales Alenia Space, No. 33986, 47p.

[78] Laurent Houssin, Christian Artigues, and Erwan Corbel. Frequency allocation problem in a SDMA satellite communication system. *Computers and Industrial Engineering*, 61(2):346–351, 2011.

[79] IBM Corp., Armonk, New York. *IBM ILOG CPLEX Optimization Studio V12.4 documentation*, 2012.

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

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

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

[83] B. Jaumard, O. Marcotte, C. Meyer, and T. Vovor. Comparison of column generation models for channel assignment in cellular networks. *Discrete Applied Mathematics*, 118:299–322, 2002. previously published as technical report of Ecole Polytechnique de Montréal, November 1998.

[84] B. Jaumard and T. Vovor. A column generation approach for the exact solution of channel assignment problems. Technical Report G-98–39, Ecole Polytechnique de Montréal, July 1998.

[85] A. Kapsalis, P. Chardaire, V. Rayward-Smith, and G. Smith. The radio link frequency assignment problem: A case study using genetic algorithms. In Terence Fogarty, editor, *Evolutionary Computing*, volume 993 of *Lecture Notes in Computer Science*, pages 117–131. Springer Berlin / Heidelberg, 1995.

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

[87] Dervis Karaboga and Bahriye Basturk. A powerful and efficient algorithm for numerical function optimization: artificial bee colony (abc) algorithm. *Journal of Global Optimization*, 39:459–471, 2007.

[88] I. Katzela and M. Naghshineh. Channel assignment schemes for cellular mobile telecommunication systems. *Personal Communications Magazine*, 3(3):10–31, 1996.

[89] K. Kiatmanaroj, C. Artigues, L. Houssin, and F. Messine. Frequency assignment in a SDMA satellite communication system with beam decentring feature. submitted to Computational Optimization and Applications.

[90] K. Kiatmanaroj, C. Artigues, L. Houssin, and F. Messine. Frequency allocation in a SDMA satellite communication system with beam moving. In *IEEE International Conference on Communications (ICC), Ottawa*, 2012.

[91] K. Kiatmanaroj, C. Artigues, L. Houssin, and F. Messine. Greedy algorithms for time-frequency allocation in a SDMA satellite communication system. In *9th Int. Conf. on Modeling, Optimization and SIMulation (MOSIM), Bordeaux*, 2012.

[92] K. Kiatmanaroj, C. Artigues, L. Houssin, and F. Messine. Hybrid discrete-continuous optimization for the frequency assignment problem in satellite communication system. In *14th IFAC symposium on Information Control in Manufacturing, (INCOM), Bucharest*, 2012.

[93] J.-S. Kim, S. Park, P. Dowd, and N. Nasrabadi. Cellular radio channel assignment using a modified hopfield network. *IEEE transactions on vehicular technology*, 46(4):957–967, 1996.

[94] S. Kim and K.-N. Chang. Optimal channel allocation for cellular mobile systems with nonuniform traffic distribution. *INFOR*, 32:203–213, 1994.

[95] S. Kirkpatrick, C. D. Gelatt, Jr., and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220:671–680, 1983.

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

[97] Joshua Knowles, Richard Watson, and David Corne. Reducing Local Optima in Single-Objective Problems by Multi-objectivization. pages 269–283. 2001.

[98] A. W. J. Kolen. A genetic algorithm for frequency assignment. *Statistica Neerlandica*, 61(1):4–15, 2007.

[99] A. M. C. A. Koster, C. P. M. van Hoesel, and A. W. J. Kolen. The partial constraint satisfaction problem: Facets and lifting theorems. *Operations Research Letters*, 23(3–5):89–97, 1998.

[100] A. M. C. A. Koster, C. P. M. van Hoesel, and A. W. J. Kolen. Optimal solutions for a frequency assignment problem via tree-decomposition. *Lecture Notes in Computer Science*, 1665:338–349, 1999.

[101] A. M. C. A. Koster, C. P. M. van Hoesel, and A. W. J. Kolen. Solving frequency assignment problems via tree-decomposition. *Electronic Notes on Discrete Mathematics*, 3, 1999.

[102] A. M. C. A. Koster, C. P. M. van Hoesel, and A. W. J. Kolen. Solving partial constraint satisfaction problems with tree decomposition. *Networks*, 40(3):170–180, 2002.

[103] Arie M. C. A. Koster, Stan P. M. van Hoesel, and Antoon W. J. Kolen. Optimal solutions for frequency assignment problems via tree decomposition. In *Proceedings of the 25th International Workshop on Graph-Theoretic Concepts in Computer Science*, WG '99, pages 338–349, London, UK, UK, 1999. Springer-Verlag.

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

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

[106] R. Leese and S. Hurley. *Methods and algorithms for radio channel assignment*. Oxford lecture series in mathematics and its applications. Oxford University Press, 2002.

[107] Joseph C. Liberti and Theodore S. Rappaport. *Smart Antennas for Wireless Communications: IS-95 and Third Generation CDMA Applications*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 1999.

[108] Andréa Carneiro Linhares, Philippe Michelon, and Dominique Feillet. An exact site availability approach to modeling the d-fap. *Electronic Notes in Discrete Mathematics*, 36(0):1 – 8, 2010.

[109] Jing Liu, Jinshu Li, Weicai Zhong, Li Zhang, and Ruochen Liu. Minimum span frequency assignment based on a multiagent evolutionary algorithm. *International Journal of Swarm Intelligence Research (IJSIR)*, 2(3):29–42, 2011.

[110] F. Luna, E. Alba, A. J. Nebro, and S. Pedraza. Evolutionary algorithms for real-world instances of the automatic frequency planning problem in GSM networks. In *Seventh European Conference on Evolutionary Computation in Combinatorial Optimization (EVOCOP 2007)*, volume 4446 of *Lecture Notes in Computer Science*, pages 108–120. 2007.

[111] Francisco Luna, César Estébanez, Coromoto León, José M. Chaves-González, Enrique Alba, Ricardo Aler, Carlos Segura, Miguel A. Vega-Rodríguez, Antonio J. Nebro, José M. Valls, Gara Miranda, and Juan A. Gómez-Pulido. Metaheuristics for solving a real-world frequency assignment problem in gsm networks. In *Proceedings of the 10th annual conference on Genetic and evolutionary computation*, GECCO '08, pages 1579–1586, New York, NY, USA, 2008. ACM.

[112] Hakim Mabed, Alexandre Caminada, and Jin-Kao Hao. Multi-period channel assignment. In Marco Conti, Silvia Giordano, Enrico Gregori, and Stephan Olariu, editors, *Personal Wireless Communications*, volume 2775 of *Lecture Notes in Computer Science*, pages 541–554. Springer Berlin / Heidelberg, 2003.

[113] Hakim Mabed, Alexandre Caminada, and Jin-Kao Hao. Genetic tabu search for robust fixed channel assignment under dynamic traffic data. *Computational Optimization and Applications*, 50:483–506, 2011.

[114] Hakim Mabed, Alexandre Caminada, Jin-Kao Hao, and Denis Renaud. A dynamic traffic model for frequency assignment. In *Proceedings of the 7th International Conference on Parallel Problem Solving from Nature*, PPSN VII, pages 779–788, London, UK, UK, 2002. Springer-Verlag.

[115] V. Maniezzo and A. Carbonaro. An ants heuristic for the frequency assignment problem. *Future Generation Computer Systems*, 16:927 – 935, 2000.

[116] Z.Á. Mann. *Linear Programming: New Frontiers in Theory and Applications*. Mathematics Research Developments. NOVA SCIENCE PUB, 2012.

[117] C. Mannino and A. Sassano. An enumerative algorithm for the frequency assignment problem. *Discrete Applied Mathematics*,

129(1):155–169, 2003. Previously published as technical report 1096, Università di Roma La Sapienza, 1998.

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

[119] MATLAB. *version 7.7.0 (R2008b)*. The MathWorks Inc., Natick, Massachusetts, 2008.

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

[121] A. Mishra, S. Banerjee, and W. Arbaugh. Weighted coloring based channel assignment for wlans. *ACM SIGMOBILE Mobile Computing and Communications Review*, 9(3):19–31, 2005.

[122] R. Montemanni, J. N. J. Moon, and D. H. Smith. An improved tabu search algorithm for the fixed spectrum frequency assignment problem. *IEEE Transactions on Vehicular Technology*, 52(4):891–901, 2003.

[123] R. Montemanni, D. H. Smith, and S. M. Allen. An ANTS algorithm for the minimum-span frequency-assignment problem with multiple interference. *IEEE Transactions on Vehicular Technology*, 15(5):949–953, 2002.

[124] R. Montemanni, D.H. Smith, and S.M. Allen. Lower bounds for fixed spectrum frequency assignment. *Annals of Operations Research*, 107:237–250, 2001.

[125] R. Montemanni, D.H. Smith, and S.M. Allen. An improved algorithm to determine lower bounds for the fixed spectrum frequency assignment problem. *European Journal of Operational Research*, 156(3):736 – 751, 2004.

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

[127] Pablo and San Segundo. A new dsatur-based algorithm for exact vertex coloring. *Computers & Operations Research*, 39(7):1724 – 1733, 2012.

[128] Mireille Palpant, Cristian Oliva, Christian Artigues, Philippe Michelon, and Mohamed Didi Biha. Models and methods for frequency assignment with cumulative interference constraints. *International Transactions in Operational Research*, 15(3):307–324, 2008.

[129] Mireille Palpant, Cristian Oliva, Christian Artigues, Philippe Michelon, and Mohamed Didi Biha. Models and methods for frequency assignment with cumulative interference constraints. *International Transactions in Operational Research*, 15(3):307–324, 2008.

[130] Christos H. Papadimitriou and Kenneth Steiglitz. *Combinatorial optimization: algorithms and complexity*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1982.

[131] Panos M. Pardalos and Jue Xue. The maximum clique problem. *Journal of Global Optimization*, 4:301–328, 1994. 10.1007/BF01098364.

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

[133] Michael L. Pinedo. *Scheduling: Theory, Algorithms, and Systems*. Springer, 4th edition, 2012.

[134] A. Punter. *Systems for timetabling by computer based on graph colouring*. PhD thesis, C.N.A.A., Hatfield Polytechnic, 1976.

[135] Helena Ramalhinho-Lourenço, Olivier C. Martin, and Thomas Stützle. Iterated local search. Economics Working Papers 513, Department of Economics and Business, Universitat Pompeu Fabra, November 2000.

[136] A. Raychaudhuri. Further results on t-coloring and frequency assignment problems. *SIAM Journal on Discrete Mathematics*, 7(4):605–613, 1994.

[137] Fred S. Roberts. T-colorings of graphs: recent results and open problems. *Discrete Mathematics*, 93(2 - 3):229 – 245, 1991.

[138] R.H. Roy. Spatial division multiple access technology and its application to wireless communication systems. In *Vehicular Technology Conference, 1997 IEEE 47th*, volume 2, pages 730–734 vol.2, May 1997.

[139] R.H. Roy. An overview of smart antenna technology: the next wave in wireless communications. In *Aerospace Conference, 1998. Proceedings., IEEE*, volume 3, pages 339–345 vol.3, Mar 1998.

[140] S. Salcedo-Sanz and C. Bousoño Calzón. A hybrid neural-genetic algorithm for the frequency assignment problem in satellite communications. *Applied Intelligence*, 22(3):207–217, May 2005.

[141] S. Salcedo-Sanz, R. Santiago-Mozos, and C. Bousono-Calzon. A hybrid hopfield network-simulated annealing approach for frequency assignment in satellite communications systems. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 34(2):1108 – 1116, April 2004.

[142] Ayed A. Salman, Imtiaz Ahmad, Mahamed G.H. Omran, and Mohammad Gh. Mohammad. Frequency assignment problem in satellite communications using differential evolution. *Computers & Operations Research*, 37(12):2152 – 2163, 2010.

[143] M. Sanchez, D. Allouche, S. De Givry, and T. Schiex. Russian doll search with tree decomposition. In *Proceedings of the 21st international jont conference on Artifical intelligence*, IJCAI'09, pages 603–608, San Francisco, CA, USA, 2009. Morgan Kaufmann Publishers Inc.

[144] E. Segredo, C. Segura, and C. Leon. A multiobjectivised memetic algorithm for the frequency assignment problem. In *Evolutionary Computation (CEC), 2011 IEEE Congress on*, pages 1132 –1139, June 2011.

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

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

[147] Peter Stadler and Santa Institute. Towards a theory of landscapes. In *Complex Systems and Binary Networks*, volume 461-461 of *Lecture Notes in Physics*, pages 78–163. Springer Berlin / Heidelberg, 1995.

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

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

[150] S. R. Tiourine, C. A. J. Hurkens, and J. K. Lenstra. Local search algorithms for the radio link frequency assignment problem. *Telecommunication Systems*, 13:293–314, 2000.

[151] Etsuji Tomita, Akira Tanaka, and Haruhisa Takahashi. The worst-case time complexity for generating all maximal cliques and computational experiments. *Theoretical Computer Science*, 363(1):28 – 42, 2006.

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

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

[154] Michel Vasquez, Audrey Dupont, and Djamal Habet. Consistent neighbourhood in a tabu search. In Toshihide Ibaraki, Koji Nonobe, Mutsunori Yagiura, Ramesh Sharda, and Stefan Voß, editors, *Metaheuristics: Progress as Real Problem Solvers*, volume 32 of *Operations Research/Computer Science Interfaces Series*, pages 369–388. Springer US, 2005.

[155] Mirjana Čangalović and Jan A.M. Schreuder. Exact colouring algorithm for weighted graphs applied to timetabling problems with lectures of different lengths. *European Journal of Operational Research*, 51(2):248 – 258, 1991.

[156] Černý V. Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm. *Journal of Optimization Theory and Applications*, 45:41–51, 1985.

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

[158] L. Wang and W. Gu. Genetic algorithms with stochastic ranking for optimal channel assignment in mobile communications. *Lecture Notes in Computer Science*, 3314:154–159, 2004.

[159] Lipo Wang, Wen Liu, and Haixiang Shi. Noisy chaotic neural networks with variable thresholds for the frequency assignment problem in satellite communications. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 38(2):209 –217, March 2008.

[160] Wei Wang and C.K. Rushforth. An adaptive local-search algorithm for the channel-assignment problem (cap). *Vehicular Technology, IEEE Transactions on*, 45(3):459 –466, Aug 1996.

[161] L. Darrell Whitley, Francisco Chicano, Enrique Alba, and Francisco Luna. Elementary landscapes of frequency assignment problems. In *Proceedings of the 12th annual conference on Genetic and evolutionary computation*, GECCO '10, pages 1409–1416, New York, NY, USA, 2010. ACM.

[162] K.L. Yeung and T.S. Yum. The optimization of nominal channel allocation in cellular mobile systems. In *Proceedings of the IEEE ICC*, pages 915–919, 1993.

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

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

AUTHOR: Kata KIATMANAROJ

TITLE: Frequency assignment for satellite communication systems

SUPERVISORS: Christian ARTIGUES, Laurent HOUSSIN

PLACE AND DEFENSE DATE: LAAS-CNRS, June 2012

ABSTRACT:

In this thesis, we consider frequency assignment problems arising from an SDMA satellite communication system which consists of a satellite and a number of users distributed inside a fixed sized service area. The objective is to assign a given number of frequency carriers to as many users as possible. This assignment should not violate the incurred interference constraints. Two types of interference are considered i.e. binary and cumulative interference. For each of them, single carrier and multiple carrier frequency assignment models are taken into account. We also propose an Integer Linear Programming (ILP) formation to deal with 2-dimensional frequency-time assignments which is more complicated and harder to solve. Single carrier FAPs are solved by greedy algorithms and ILP. A Beam Moving algorithm is devised to further improve the solutions by solving a non-linear optimization problem. Multiple carrier FAPs are modeled as scheduling problem and ILPs. We show that the scheduling model solved through constraint programming methods offers superior performance than the proposed ILP. It is worth noting that, by transforming the cumulative interference into binary interference, scheduling method together with clique-induced constraints yields much better results.
A frequency assignment problem that incorporates the specifications and constraints provided by the industry is also considered. These requirements render the resource allocation problem highly complex. This complexity and the fact that frequency assignment plans must be recomputed frequently in order to cope for user mobility yield classic optimization tool such as ILP impractical. According to this, two greedy algorithms are devised and tested.

KEYWORDS: frequency allocation, combinatorial optimization, telecommunications