Definition of a Common Formulation of Military Frequency Assignment Problems and the Application of Meta-Heuristic Algorithms

James Graham

Division of Mathematics and Statistics
University of Glamorgan

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

April 2005

Acknowledgements

I would like to thank all of the staff at QinetiQ who provided the relevant information needed to complete this project, with special thanks to Dr. Leslie D. Westbrook and Mr. Ray Bradbeer.

A special thanks goes to Professor Derek H. Smith for all his guidance, support and supervision throughout this project. I would also like to thank my second supervisor Dr. Jim Moon for all of his help and input.

Abstract

The purpose of this research is to define a common formulation for military frequency assignment problems, in a form which allows the application of meta-heuristic algorithms. Currently different frequency assignment systems are in use in the four areas of the military, in the categories of land, air, maritime and satellite. The tasks are to define the common aspects of the relevant frequency assignment problems in the four areas and then to implement a piece of software that may tackle all of these problems with little or no loss in performance when compared to existing frequency assignment problem solving systems. Much work has been done over the years in developing new and more efficient techniques to solve military frequency assignment problems. It has never been attempted to create a generic system to handle all types of problems.

This thesis contains a general background to military and terrestrial frequency assignment problems, with relevant literature referenced. The majority of military frequency assignment systems rely heavily on the solving of binary constraints as they have been proved to be extremely efficient. This research uses a combination of binary constraints and signal-to-interference calculations to solve frequency assignment problems. The signal-to-interference techniques are more accurate than the use of binary constraints only, as signal-to-interference considers interference from multiple sources while the binary constraints only handle single sources of interference. Although signal-to-interference techniques are slower than the use of binary constraints, this research has shown that, if used in the correct context, signal-to-interference techniques substantially improve network performance with acceptable performance. Appropriate propagation models have been used to generate test data; these models are presented in the thesis.

The model presented in this research addresses the assignment of multiple carrier types. These carrier types consist of homogeneous narrow band carrier types, aggregated channel carriers, heterogeneous wide band carrier types, CDMA carriers, unsynchronised frequency hopping carriers and synchronised frequency hopping carriers. Test data has been generated that includes all of these carrier types in a frequency assignment problem. The model incorporates a weighted linear cost function consisting of binary constraint violations, signal-to-interference based cost, frequency hopping list costs, and costs associated with spurious emissions, spurious responses, and intermodulation products. The research focused on the use of the simulated annealing meta-heuristic algorithm to solve the frequency assignment problems presented here. The algorithm can use any linear combination of the first three components of the cost function as listed above, and could optionally include the last three.

The experiments performed and presented in the thesis have shown that the assignment of multiple carrier types in a single frequency assignment problem solving system is possible. Signal-to-interference techniques can be employed to further improve network performance when combined with binary constraints and that, most importantly, a model of the common formulation of military frequency assignment can be implemented with acceptable performance. The experiments also show that signal-to-interference techniques are best used in conjunction with binary constraints. The binary constraints should be used to create an initial assignment which can then be improved by signal-to-interference techniques. For this purpose it appears that binary constraints are best generated at a required signal-to-interference threshold 2-4dB higher than the threshold required for the network.

This research, as described above, represents an original contribution in the area of military radio frequency assignment through the pursuit of a common formulation of the four military areas of frequency assignment. The merit of this approach is that advances in frequency assignment algorithms can be applied directly to all of the problems encompassed. This has clear advantages over the separate development of techniques for individual problem areas.

Contents

List of Figures

List of Tables

Chapter 1

An Introduction to Frequency Assignment

1.1 Specific Aims of the Work

The purpose of this research is to define a common formulation for military fixed spectrum frequency assignment problems. The objective of this common formulation is to provide a framework for the improvement of frequency planning in all areas of military frequency assignment. This formulation must be modelled in a form which allows the application of meta-heuristic algorithms. Currently different frequency assignment systems are in use for the four areas of the military, in the categories of land, air, maritime and satellite. The common aspects of these relevant frequency assignment areas must be defined. Once defined a model can be created and implemented in software. The model of the formulation must be capable of being implemented with little or no loss in performance when compared to existing frequency assignment problem solving systems. Successful completion of these aims would allow research on frequency assignment algorithms for all areas of the military to be combined in a single formulation.

1.2 Achievements of the Work

A generic model has been designed and implemented in software. The model incorporates a weighted linear cost function encompassing binary constraint violations, signal-to-interference (SIR) based cost, frequency hopping list costs, spurious emissions, spurious responses, and intermodulation products. A hill climbing algorithm and a simulated annealing algorithm have been implemented. The algorithms can use any linear combination of the first three components of the cost function, and could optionally include the last three. Valid frequencies can be assigned to narrow band homogeneous carrier types, aggregated channel carrier types, wide band heterogeneous carrier types and CDMA carrier types. Valid lists of frequencies can be assigned to individual unsynchronised hopping carriers and groups of synchronised hopping carriers. Experiments have been performed and results have been presented that show that the software has excellent performance in many circumstances and acceptable performance in all circumstances. The experiments also clarify the use of binary constraints in aiding the solution of problems formulated with an SIR based cost function.

1.3 Overview

The efficient assignment of frequencies to the transmitters of radio systems has been an active research area for many years now. A large variety of literature has appeared during this time. The task of assigning frequencies effectively is generally known as the *Frequency Assignment Problem (FAP)*, or *Channel Assignment Problem (CAP).* The main impetus for this work has been the rapid implementation of wireless telephone networks (e.g. GSM networks) and other terrestrial and satellite communication systems. At the same time related developments in military communication systems have also inspired new research. There is a continuing and growing demand for fast and efficient deployment of these systems during military operations.

The rapid development of all the radio services mentioned has resulted in a scarcity of the most important resource, the radio spectrum. As with all scarce resources, there is a need for economic use of the available frequencies. Reuse of frequencies within a wireless system can offer economies that are critical to the operation of a radio network. However, reuse of frequencies may also lead to a loss in quality in the communication links. The use of the same (or almost the same) frequency for multiple wireless connections can cause unacceptable interference between signals. Frequencies should be selected in such a way that interference is avoided or minimised. The FAP aims to balance the economies of reuse of frequencies with the loss of quality in the network.

1.4 History of Frequency Assignment Problems

FAPs first appeared in the 1960's [29]. The development of new wireless services such as the first cellular phone networks, led to a scarcity of useable frequencies in the radio spectrum. Frequencies were licensed by governments, who charged operators for the use of each single frequency band separately. This introduced the need for operators to develop frequency plans that not only avoid high interference levels, but also maximise the capacity for a given licence cost. It turned out that it was far from obvious how to find such a plan. At this point techniques from operational research and graph theory were introduced.

Until the early 1980s, most contributions on frequency assignment used heuristics based on the related graph colouring problem. The first lower bounds were derived by Gamst and Rave [20] in 1982. For a more recent paper on lower bounds see [32]. The widespread development of a common digital cellular phone standard, General System for Mobile Communication (GSM), in the late 1980s and 1990s led to a rapidly increasing interest in frequency assignment.

1.5 The Need for Solutions of Frequency Assignment Problems

Two points can establish a wireless connection with the use of a transmitter and receiver. Electromagnetic oscillations at a radio frequency can be amplitude, phase or frequency modulated. Receivers can pick up these oscillations and demodulate them to recover the original data. Two transmitters using the same or a very close frequency may interfere. Interference can also be caused by other mechanisms such as spurious emissions, spurious responses and intermodulation. The level of this interference can depend on many aspects such as distance between the transmitters and receivers, the geographical position of the transmitters, the power of the signal, antenna directivity, and the weather conditions. With high levels of interference, the received signal may drop below the required signal-to-noise ratio, causing an unacceptable loss of quality. However, due to the limited availability of frequencies, they must be reused by many transmitters within the same network.

As a consequence, an operator should carefully choose the frequencies on which each station transmits to avoid unacceptable interference-levels, hence the need for the solutions of FAPs. Depending on the application, the conditions that should be satisfied by the frequency plan may vary. Furthermore, the interference may be handled in a number of different ways and the available set of frequencies may differ among applications.

1.6 Military Frequency Assignment Problems

A European research programme was set up in the early 1990s called "European Cooperation on the Long term In Defence" or EUCLID [44]. Through this program came the "Combinatorial Algorithms for Military Applications" or CALMA project. This project involved six research groups from three European countries. Each group contributed its expertise in a particular area of combinatorial optimisation to the project. The groups involved were as follows: Centre d'Etudes et de Recherches de Toulouse, France; Delft University of Technology, The Netherlands; Kings College London, United Kingdom; Limburg University, Maastricht, The Netherlands; University of East Anglia, United Kingdom. The groups' work consisted of looking at many different algorithmic approaches to solving military radio links FAPs [8], [44].

The use of terrestrial battlefield communication systems or airborne radios leads to FAPs that are dynamic in time and space. The receivers have a certain amount of selectivity, allowing them to ignore nearby transmitters. Thus the main requirement is a sufficient frequency separation between pairs of transmitters. However, an important part of these problems is the fact that each connection consists of two mobile terminals. Therefore two frequencies must be assigned for each connection (one for each direction of communication) and in many radio links systems these must be a fixed number of channels apart to avoid unacceptable levels of interference. Thus, all frequencies are given in pairs with this fixed number of channels between them.

In satellite communications both the transmitters and receivers are ground, air or maritime terminals. They communicate with each other with the help of one or more satellites. Each signal is first transmitted via an uplink to the satellite and then transmitted by the satellite via a downlink to the receiving terminal. A fixed number of channels that is much larger than the signal bandwidth is typically used to separate the uplink and downlink frequency, and it is always specified which has the greater frequency. There are four frequencies involved: mobile station to satellite, satellite to mobile station, satellite to base station and base station to satellite. Thus there are two uplink/downlink pairs, so two independent frequencies are required. This implies that frequencies need only be assigned to the uplink. This is a simpler situation than is described for radio links above, where it is not specified which direction must have the greater frequency.

1.7 Omissions

The initial part of the work involved the development of an understanding of the role of frequency assignment in the wide variety of military communication systems in use and planned. An understanding of the requirements for frequency assignment was vital if any useful generic formulation was to be attempted. Much useful co-operation was obtained from staff of QinetiQ. Inevitably the level of co-operation was variable across the range of activity in QinetiQ. Thus issues of staff availability and security led to a much more detailed understanding of satellite and land military communications than of sea and air communications. Thus some assumptions were required to create a generic system covering all areas of military communications.

One specialised area of frequency assignment concerns techniques for handling constraints which avoid intermodulation effects, spurious emission effects and spurious response effects. These have previously been developed specifically for combat net radio in [41]. This capability should be included in a way which causes no loss of efficiency when the capability is not required. As these techniques were well developed in [41] but are time consuming to implement, they were not included in the work described here. However, there appears to be no theoretical difficulty in adding this capability if required.

One further area not considered was the inclusion of intermodulation effects in signal-to-interference (SIR) calculations. In satellite systems [47] in particular, the avoidance of intermodulation products altogether is not feasible, and so their effects must be minimised. The efficient incorporation of these effects into the cost function of a meta-heuristic algorithm would appear to be particularly challenging. This has been demonstrated in the past for a single satellite beam with carriers of various bandwidths [38], but even this limited optimisation has proved to be very time consuming. Thus this work appears to require a substantial study itself before it could be considered for inclusion in a generic system.

1.8 Structure of Thesis

The material in this thesis will be presented in an order consistent with the creation of the generic model outlined in section 1.1.

Chapter 2 gives detailed descriptions of the algorithms that may be used in the generic model. These algorithms will be used to solve the frequency assignment problems that are presented in this document.

Chapter 3 specifies the different aspects of military communications. The chapter outlines the components of air, land, maritime and satellite communications. Common factors may be derived from these components. The chapter also gives a brief introduction into propagation models that are used to create data for frequency assignment problems.

Chapter 4, chapter 6, chapter 7, chapter 10 and chapter 11 give detailed descriptions of the various carrier types that are included in the model. These carrier types consist of homogeneous carrier types (chapter 4), aggregated channel carriers (chapter 6), heterogeneous wide band carriers (chapter 7), CDMA carriers (chapter 7) and hopping carriers (chapter 10 and chapter 11). Data structures, cost functions and file formats are defined for the carrier types.

Chapter 4 and chapter 5 give detailed descriptions of binary constraints, intermodulation constraints and spurious constraints. Signal to interference calculations and propagation models are outlined in chapter 8. Spectral overlap functions used for wide band carriers are discussed in chapter 9.

Chapter 12 presents the generic model designed to handle all of the aspects of military frequency assignment problems discussed in chapter 2 through to chapter 11. The model consists of models of interference, carrier types, algorithms, problem representations, cost function and the solution. Chapter 13 outlines the software that is designed and implemented to show that the model is feasible.

Chapter 14 presents the experiments performed and their results. Chapter 15 discusses the conclusions derived from this research.

1.9 Literature Review

The literature on frequency assignment has grown substantially over the past 30 years. The impetus for this research has been the widespread implementation of wireless networks (e.g. GSM networks) and of satellite communication systems. The first research on frequency assignment was published by Metzger in [29]. The use of sequential heuristics to solve frequency assignment problems were first shown by Hale in [23] and Gamst and Rave in [20]. Hale presented an overview of the frequency assignment problems of that time, focussing on modelling the problems. Hale [22] also used the relationship of the frequency assignment problem with graph colouring, which has already been mentioned by Metzger. Gamst and Rave's research introduced the first use of lower bounds in frequency assignment. More recent use of lower bounds can be found in [3], [4], [32] and [34].

In recent years meta-heuristic algorithms have been shown to be particularly effective for the frequency assignment problem. The simulated annealing algorithm was derived from thermodynamics and was first published by Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller and E. Teller in [28]. Thirty years later Kirkpatrick, Gelatt and Vecchi in [27] published further use of the simulated annealing algorithm in optimisation. The first application of the simulated annealing algorithm in frequency assignment can be seen in the work of Duque-Anton, Kunz and Ruber in [16]. The simulated annealing algorithm adopted for this work is from [26]. The use of the tabu search algorithm to solve frequency assignments problems was first demonstrated by Bouju, Boyce, Dimitropoulos, vom Scheidt and Taylor in [7]. Since, there has been much research performed on the use of the tabu search algorithm, see [10], [12], [24], [26], [31] and [39]. Genetic algorithms have been shown to be effective when applied to FAPs [42]. More recently an ANTS algorithm [33] has been developed and has been shown to work well when solving binary constraint problems, signal-to-interference problems and combinations of the two [21]. Other meta-heuristic have been used, e.g. threshold accepting [13]. For recent surveys on the frequency assignment problem see [1], [8], [18] and [44]. Much consideration went into deciding on the algorithms that would be applied to the common formulation presented here, chapter 2 outlines the decisions.

The majority of the literature published is concerned with algorithms that use binary constraints. The EUCLID CALMA project [8], [19] and [42] presented some early benchmarks and also introduced the well establish CELAR format for representing frequency assignment problems with binary constraints. The CELAR format is adopted for this work as it allows the transfer of test data with ease. Binary constraint algorithms were applied to the "Philadelphia" benchmarks for wireless networks, the optimal solutions presented in [40] were obtained using the algorithms described in [26]. A research project (COST 259) on mobile telephone networks is presented in [11]. Important benchmarks for weighted frequency assignment were presented in the COST 259 project. Further benchmarks (including transmitter locations) can be found in [14], the 458 transmitter benchmark from [14] will be used in this work. Much of the proposals presented in the literature for the solving of binary constraints will be used here, forming the core of this work.

Relevant information about specific military communication systems (i.e. land, air, satellite and maritime) can be obtained from a number of sources [43], [17], [2] and [30]. Although much information about military systems is confidential, these sources are all public websites and contain general information for military radio systems.

The idea of minimising a cost function using signal-to-interference calculations was first shown in [46]. In [46] an algorithm is defined that minimises deficits in the signal-to-interference ratio at a defined set of "reception points". Algorithms for fixed spectrum problems based on the same idea were later published by Capone and Trubian in [9] and independently by Hurley, Whitaker and Smith in [25]. It was demonstrated in these papers that such algorithms gave better results than the simple binary constraint approach to frequency assignment. The techniques developed for solving signal-to-interference problems that have been presented in the literature have been adopted for this work, but have been taken further by considering multiple carrier types and combining the solution of binary constraints with signal-to-interference techniques. A recent paper [21] presents the novel method that combines the solution of binary constraints with these signal-to-interference techniques, derived from the work presented here.

There is limited literature in the area of intermodulation products and spurious emissions. Research presented in [41] introduced a method for reducing the effects of intermodulation products and spurious emissions and responses in radio networks with co-sited radios. Intermodulation effects must also be minimised in satellite systems. A method is presented by Pujante and de Haro in [38] that attempts to address the problem for a single satellite beam. The issue of intermodulation in satellite communications is also mentioned in [47].

Recent research has been completed to enhance the capacity of GSM networks using the technique known as frequency hopping. A detailed account of frequency hopping in GSM networks can be found in [45]. The application of meta-heuristic algorithms, namely the simulated annealing and tabu search algorithms, to the assignment of frequency lists in GSM networks have been presented by Björklund, Värbrand and Yuan in [5] and [6]. Research presented by Moon, Hughes and Smith in [35] creates benchmarks for frequency hopping problems by modifying the COST 259 problem and shows that the hopping lists should not be pre-generated but modified as the algorithm proceeds. Further reference to frequency hopping can be found in [11]. The GSM proposals presented in the literature have been adapted for this work and combined with novel military specific requirements.

The current work includes the assignment of stacks of synchronised Code Division Multiple Access (CDMA) carriers. A description of CDMA can be found in [37].

The literature stated shows the feasibility of finding good solutions to various types of frequency assignment problems in many realistic circumstances. The question addressed here is whether a single system can be derived that can cope with all the circumstances of military relevance and retain efficiency.

Chapter 2

Algorithms for Frequency Assignment Problems

This chapter gives an overview of the main algorithms used in the solution of frequency assignment problems.

2.1 Sequential and Meta-Heuristic Algorithms

Most early approaches to solving FAPs used sequential (or greedy) algorithms to create an assignment one transmitter at a time and once a transmitter has been assigned a frequency, the frequency will not change later. The work of Hale in 1980 [23] led sequential orderings to some degree of sophistication, but they are known to be less effective than meta-heuristics and will not be used or discussed further in any part of this project.

Meta-heuristic algorithms are local search algorithms which take a given solution and proceed to make small changes to create some sample neighbourhood of solutions of the current solution. The meta-heuristic can then decide on the acceptance of the new solutions chosen from the neighbourhood. Ideally the meta-heuristic will have the ability to escape from local minima of their cost function in the search space. Effective meta-heuristic are able to do this, causing a much-needed improvement in performance. A detailed description of some meta-heuristic algorithms applied to frequency assignment can be seen in [12], [16], [26] and [40]. For recent surveys of frequency assignment techniques see [1], [18] and [40]. FAP web [18] is a web-site devoted to frequency assignment problems in wireless communication networks and has many available resources. This report will focus on the use of simulated annealing to solve specific FAPs.

Many algorithms, and variations of these algorithms have been presented in the above literature. These algorithms include Genetic Algorithms [42] and more recently an ANTS algorithm [33]. However, the aim of this work is to apply algorithms to a wide range of military FAPs, rather than design specific new algorithms. Hence only the hill climb and simulated annealing algorithm will be used; the tabu search algorithm will not be implemented. The simulated annealing algorithm has been shown in the past to be generally effective when used on FAPs [26]. The algorithm is straightforward to implement and has a very good capability of being expanded to handle more complex FAPs. The main advantage that the simulated annealing algorithm has over tabu search is when SIR calculations are considered. The tabu search algorithm [26] looks at the potential of many moves before making one, but due to the complexity of SIR calculations it is possible that the tabu search algorithm would struggle to make progress within an acceptable time limit. In [31] for example, very long runtimes are required for the best results. However, the simulated annealing algorithm simply selects one random transmitter and one random frequency and tests the potential of this one move, and has been shown to perform well for SIR [25]. Therefore, given the uncertainties that arise as more military systems are included, simulated annealing is the most sensible choice due to its simplistic but effective nature.

2.2 Hill Climbing

Hill climbing is a basic local search algorithm. The nature of hill climbing is to start with a given solution and replace it with a better one selected from a restricted subset of the solution set. The starting solution is usually a random one giving initial high cost, but a start found using a sequential algorithm may also be used. If no better (or improving) solution exists in the restricted subset then the algorithm would stop. Thus the algorithm terminates at a local minimum of the cost function, but this is not necessarily a global minimum.

2.3 Simulated Annealing

Simulated annealing is a stochastic computational technique derived from statistical mechanics for finding near-global-minimum-cost solutions to large optimisation problems [27] and [28]. The method is derived from thermodynamics, specifically from the way that liquids freeze or that metals cool and anneal. At high temperatures the molecules of a liquid move freely with respect to one another. Thermal mobility is restricted if the liquid is cooled slowly. The atoms are often able to line themselves up and form a completely regular crystal. This crystal represents the state of minimum energy for the system. However, if a liquid is cooled quickly it does not reach a minimum energy state but instead solidifies at a higher energy state. This higher state would correspond to a sub optimal solution.

Specifically, the algorithm will normally begin with a random start giving a high initial cost. The best solution in the neighbourhood is accepted as a new current solution if it is better than the old one, or otherwise with a probability that depends on its value. This probability increases as the difference between the new value and current value decreases. The acceptance probability is also adjusted by the temperature. The temperature decreases (the system cools) as the number of iterations increases[1]. The lower the temperature the lower the acceptance probability, hence when the temperature is very low, non-improving solutions will almost never be accepted and the algorithm will begin to imitate hill climbing.

The main aspects of the simulated annealing algorithm are as follows:

- Solution representation: each feasible solution of the problem must have a unique representation within the search space;
- Cost Function: a function that may represent each feasible solution as a numeric value. The goal of the algorithm is to find a solution that would minimise the cost;
- Neighbourhood: each time the algorithm has to consider a new solution it is chosen randomly among those in the neighbourhood of the current solution (i.e. the new solution differs from the current solution in the frequency assigned to precisely one transmitter);
- Temperature *t*: a control parameter similar to that used in the physical annealing process. Initially the *temperature* is set to a high value that decreases as the algorithm progresses. The parameter is used in the decision whether or not to accept a new solution (i.e. where ΔE<0 or random<$e^{\frac{-\Delta E}{t}}$);
- Annealing schedule: this is the process that simulates the cooling, i.e. how the *temperature* is lowered from a high value to low values. Different cooling techniques are described in some detail in [26], although for this project only one cooling technique shall be used (i.e. t:=βt, where 0<β<1). This was shown in [26] to be a simple but effective cooling technique;

---

[1] There are a number of ways in which the temperature may be decreased [26].

- Termination criterion: once the termination criterion is satisfied the algorithm will stop. The criterion is usually represented by a minimum value for the *temperature*; however, it would be sensible for the algorithm to stop if a minimum cost is found with no further improvement in a given further number of iterations.

The simulated annealing algorithm may be expressed as pseudocode as shown in Figure 2.1.

```
Initialise $t$
Generate random configuration $X_{old}$ of cost $E_{old}$
$X_{best} \Leftarrow X_{old}$
$E_{best} \Leftarrow E_{old}$
WHILE $t > t_{min}$ DO
    FOR $i$=1 to $NUM_{loop}$ DO
        generate new configuration, $X_{new}$, from $X_{old}$
        calculate new cost, $E_{new}$
        ΔE $\Leftarrow$ $E_{new}$ - $E_{old}$
        generate $random$ such that $random \in [0,1)$
        IF ΔE $< 0$ or $random <$ prob $= e^{\frac{-\Delta E}{t}}$
        THEN
            $X_{old} \Leftarrow X_{new}$
            $E_{old} \Leftarrow E_{new}$
            IF $E_{new} < E_{best}$
            THEN
                $X_{best} \Leftarrow X_{new}$
                $E_{best} \Leftarrow E_{best}$
            END IF
        END IF
    END FOR
    t:=$\beta$t, (0<$\beta$<1)
END WHILE
RETURN $E_{best}$, $X_{best}$
```

Figure 2.1: Simulated annealing algorithm.

2.3.1 Threshold Accepting

Threshold accepting is a variation of simulated annealing. New assignments are accepted if the cost reduces or if it increases by no more than some specified threshold. The threshold may reduce over time in a similar way to the temperature in simulated annealing [16].

2.4 Tabu Search

Tabu search is a meta-heuristic algorithm that partially explores the search space of all feasible solutions by a sequence of moves. In order for the algorithm to escape local minima, to prevent it from cycling over the same set of moves and to expand the assignments examined, a system of long-term and short-term memory is used. At each iteration the move that is accepted is the most promising of those available. Those that are available have their associated cost stored along with the definitions of moves that are forbidden. A short term memory condition would cause a move to be forbidden or *tabu*. Normally this would specify that a move specified cannot be repeated until a certain number of iterations have been completed after its last occurrence. A sequence of previous moves are stored in a recency list; this is then checked to determine if a move is *tabu*. The second source of *tabu* moves is the long-term memory. This is used to ensure that a request (i.e. a request for a frequency for a transmitter) does not change its frequency too many times. The long-term memory usually has as many entries as there are requests in the problem. However, long-term memory has been found to be inefficient for frequency assignment problems and is not normally used. The aim of the algorithm is to produce a minimum cost solution or near minimum cost solution. This algorithm has been used with much success in the solution of FAPs [7], [10], [12], [24], [26] and [39].

The main elements of the algorithm are:

- Solution representation: each feasible solution of the problem must have a unique representation within the search space;
- Cost Function: a function that represents each feasible solution as a numerical value. The goal of the algorithm is to find a solution that would minimise the cost;
- Neighbourhood: moves are chosen from a neighbourhood of the current solution. All neighbourhoods used have the property that the assignments they contain differ in the frequency assigned to just one transmitter. There are various choices of neighbourhood. A violating neighbourhood contains moves where the transmitter to be re-assigned is currently involved in a constraint violation. A random violating neighbourhood is a subset of a violating neighbourhood. A full violating neighbourhood can be used, but requires a technique known as a cost change table for efficient cost function updating;
- Tabu list: a list containing the moves that are forbidden. A new solution that is generated from the current neighbourhood cannot contain a move that is in the tabu list;

- Aspiration criterion: if a move that is contained within the tabu list produces a solution that satisfies this criterion then it may be accepted. The usual criterion is that the move produces the best solution obtained so far. Many tabu search algorithms for FAPs use an aspiration criterion, but Montemanni, Moon and Smith [31] recommend that no aspiration criterion be used;
- Termination criterion: once the termination criterion is satisfied the algorithm will stop.

The tabu search algorithm may be expressed as pseudocode as shown in Figure 2.2.

```
Generate random configuration X of cost E
$X_{best}$ ⇐ X
$E_{best}$ ⇐ E
WHILE (termination criterion not met) DO
    X ⇐ best non-tabu or aspirate solution in the neighbourhood of X
    Calculate cost, E
    IF E < $E_{best}$
    THEN
        $X_{best}$ ⇐ X
        $E_{best}$ ⇐ E
    END IF
    update tabu list
END WHILE
RETURN $E_{best}$, $X_{best}$
```

Figure 2.2: Tabu search algorithm.

Chapter 3

Areas of Military Frequency Assignment

3.1 Overview

This chapter contains a brief overview of the areas of military communications in which frequency assignment problems arise. Specific information on various aspects of the military was acquired from the Ministry of Defence public website [30] and also from meetings with personnel from QinetiQ plc.

3.2 Satellite

Satellite communication or SATCOM is used at the strategic, operational and tactical levels. Satellites link all three areas of the military, i.e. land, air and maritime. Headquarter bases in the field must have contact with the UK or other areas via SATCOM. Ships must also use the same method to communicate within certain areas of the world. For tactical military communication it is essential that ships and ground troops have contact with positions of authority. In the air the Airborne Warning and Control System (AWACS) maintains contact with the UK via SATCOM. AWACS provides superior radar information and a “big picture” view of the field of operations. This picture is a valuable piece of information for military intelligence as it provides a safer environment for military personnel. With the use of SATCOM key military personnel may have detailed information about the field of operations from the safety of the UK. In the future the ‘Nimrod’ maritime reconnaissance aircraft (used primarily in the roles of maritime surface surveillance, anti-submarine warfare, and search and rescue) will be able to pass surveillance data to the UK via SATCOM in real time. The Airborne Stand-Off Radar (ASTOR) surveillance aircraft will also be able to pass surveillance data to the UK via SATCOM in the future.

Much work has been done in the area of military SATCOM [47]. It is important that there is a flexible or dynamic balance between the following areas:

- For maximum data throughput at minimum cost the efficient assignment of frequencies is required;
- For robustness and graceful degradation when under attack a great deal of resilience is required.

3.3 Land

Land tactical military wireless communication systems connect troops, mobile vehicles and headquarters all stationed in theatres of operation. VHF/UHF combat net radio (e.g. Bowman, a system that also incorporates HF capability) allows individual soldiers to communicate with each other and with their headquarters. The Bowman system is due to replace the current Clansman combat net radio, offering modern digital communications. PTARMIGAN radio links will connect these networks. These trunk links are also used for the communication between stationary and mobile military vehicles. The CORMORANT system provides microwave radio relay, satellite communications, commercial cable bearers and a new tropo scatter ‘over the horizon’ capability. FALCON is a replacement communications system for PTARMIGAN still in the early process of project definition.

Combat net radio is probably the most complex FAP. New sets of frequencies are required to be assigned daily or more frequently and these should show no correlation with previous sets assigned. This unpredictability aspect of the problem is essential, as any military force would be at great disadvantage if an enemy force were able to “listen in” on ground communications. An ongoing issue with ground tactical military communications (and, of course any other communication system) is the possible interference with equipment from other nations as well as civil communication systems.

3.4 Air

Military aircraft may communicate with each other via VHF or UHF. Individual fighter jets may communicate with the AWACS aircraft for tactical or strategic information; this communication may consist of a data link directly to the fighter jet’s onboard computer that displays this information to the pilot. These jets may also communicate with the UK via HF. Fighter jets are equipped with radar to locate air or ground targets and these may interfere or conflict with radio communications.

Unmanned Airborne Vehicles (UAVs)[1] have dedicated Microwave links to the ground or in some cases satellite. A UAV is used mainly for surveillance purposes and is equipped with many sensors. Due to the fact that it is unmanned it can be placed in extremely hostile conditions without the cost of human lives. The UAV may also act as a repeater for over the horizon VHF, UHF or microwave communications.

3.5 Maritime

The UK's maritime tactical military craft consist of aircraft carriers, Type 23 frigates, Type 22 frigates, Type 43 destroyers, Type 45 anti-air warfare destroyers, assault ships and submarines. These craft communicate with each other via VHF, UHF or microwave tactical data links, similar to those used for ground communications. They may also communicate with the UK and each other via HF communication or satellite. Most of these craft are equipped with numerous items of radar and sonar equipment, giving them the ability to detect threats from great distances, which may interfere or conflict with radio communications.

3.6 Discussion on Propagation Models

The propagation models discussed in this report include a d^{-4} model (see section 8.3.1) and a $|sin(x)/x|^2$ model (see section 8.3.2). The d^{-4} propagation model is used for land based communication networks while the $|sin(x)/x|^2$ propagation model is used for satellite communications. In the areas of air and maritime communications information about the types of propagation models used is not as freely available. It may be assumed that for air and maritime communications free space loss or a modified free space loss is used. However for this research, propagation models are used to form either binary constraints or test data for SIR problems. With the use of an intermediate file (see section 8.8.3), test data produced from any propagation model may be used to generate a frequency assignment problem.

---

[1] currently UAVs have no spectrum specifically allocated for their use.

Chapter 4

Binary Constraints and Homogeneous Carrier Types

4.1 Overview

In all four areas of the military (land, satellite, maritime and air) the majority of frequency assignment systems currently use binary constraints to represent problems. The simplest binary constraint problems involve a set of transmitters for which the carriers are homogeneous, i.e. they all have the same bandwidths. These transmitters are assigned channels from sets of channels known as frequency domains. These sets of available channels may vary from transmitter to transmitter. The channels are obtained by partitioning the available spectrum into equally sized blocks of frequencies. The transmitter is assigned to the centre frequency. Channel separation constraints exist for pairs of transmitters. A cost function calculates the total number of violated constraints and this number is then minimised using an algorithm. Binary constraints are of the form $|f_i\text{-}f_j|>c$ where f_i is the channel number assigned to transmitter i.

The use of binary constraints defines a core model for frequency assignment, which can be extended to give more accurate models.

4.2 Data Structures

This section contains descriptions of the internal data structures and the way in which they are maintained. Some are fixed from the start and simply accessed from time to time, while others are variable and constantly accessed and updated. The structures used are very simple array based structures, but are efficient in operation.

For the types of data required to be stored in FAPs, two-dimensional arrays are necessary. However, two-dimensional arrays tend to be slower to access than one-dimensional arrays. Therefore, a data structure for a one-dimensional array has been created that can be used as a two-dimensional array. This is done by creating a one-dimensional array but accessing two-dimensional information from it. If the two-dimensional array is of size |R|x|C|, where |R| is the number of rows and |C| the number of columns, then position (r,c) is stored in position c + (r x |C|) of the one-dimensional array. Figure 4.1 gives an example of a two-dimensional array of size |2x3|, while figure 4.2 shows how the array is actually stored by the software. The two-dimensional data structure is generic and is used for many of the data structures in the software.

1	3	11
2	1	4

Figure 4.1: 2D Array

1
3
11
2
1
4

Figure 4.2: 1D Array

4.2.1 Variables Structure

The variables structure contains the transmitter information; this is made up of the transmitter identity and the frequency domain identity assigned to the transmitter. This information is stored in an array in which each element contains the transmitter and the frequency domain identity. The transmitter identity and the domain identity are both integer values. Figure 4.3 gives an example of the variables structure. The nature of the frequency domains will be described in section 4.2.3.

The variables data structure is fixed, i.e. once the array has been filled its content will not change throughout the running of the algorithm. The data contained in the structure is accessed from time

Transmitter	Domain
1	2
2	0
3	1
4	1
5	0
6	0
7	2
8	0
9	0

Figure 4.3: Variables data structure.

to time when required. The domain identity is required when a transmitter is moved, while the transmitter identity is required for assignment and cost function evaluation.

4.2.2 Constraint Structure

The binary constraints are made up of two transmitter identities and a channel separation value. This information is stored in a two dimensional array. The dimensions are equal to the number of transmitters. The channel separations for pairs of transmitters are then stored in the array, hence the array is symmetric. To represent the case where there is no channel separation constraint between a pair of transmitters a -1 is used. Figure 4.5 gives an example of how a set of binary constraints for a 12 transmitter problem, given in figure 4.4 would be stored in the data structure.

The binary constraints for some problems may become sparse, i.e. large numbers of -1's occur in the structure. This could be handled with the use of pointers that indicate the next channel separation constraint for a transmitter. Then the algorithm need not run through all the -1 values, but simply skip to the next constraint. This can add great efficiency to large sparse binary constraint problems, but wastes some memory when a problem is not sparse. As the relevant military problems are not all necessarily sparse this option has not been implemented. The algorithm with binary constraints is much faster than with other cost functions (see chapter 14).

In a similar way to the variables data structure the constraints data structure remains fixed throughout runtime and is not updated. The constraint information is only accessed during cost function calculations.

0	11	>	0
1	2	>	4
1	3	>	4
1	4	>	4
1	5	>	4
1	6	>	2
1	7	>	2
1	8	>	2
1	9	>	2
2	3	>	4
2	4	>	4
2	5	>	4
2	6	>	2
2	7	>	1
2	8	>	2
2	9	>	1
3	4	>	4
3	5	>	4
3	6	>	2
3	7	>	2
3	8	>	2
3	9	>	2
4	5	>	4
4	6	>	2
4	7	>	1
4	8	>	2
4	9	>	1
5	6	>	2
5	7	>	2
5	8	>	2
5	9	>	2
6	7	>	4
6	8	>	4
6	9	>	4
7	8	>	4
7	9	>	4
8	9	>	4

Figure 4.4: Binary constraints for 12 transmitter problem.

4.2.3 Frequency Domain Structure

The frequency domain data structure is used only when a domain file is in use. Otherwise the only information required is the number of channels available as the domain then consists of a set of consecutive integer channel numbers.

-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	0
-1	-1	4	4	4	4	2	2	2	2	-1	-1
-1	4	-1	4	4	4	2	1	2	1	-1	-1
-1	4	4	-1	4	4	2	2	2	2	-1	-1
-1	4	4	4	-1	4	2	1	2	1	-1	-1
-1	4	4	4	4	-1	2	2	2	2	-1	-1
-1	2	2	2	2	2	-1	4	4	4	-1	-1
-1	2	1	2	1	2	4	-1	4	4	-1	-1
-1	2	2	2	2	2	4	4	-1	4	-1	-1
-1	2	1	2	1	2	4	4	4	-1	-1	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
0	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1

Figure 4.5: Constraint data structure.

When the structure is in use it is pointed to by the domain identity in the variables data structure, see figure 4.3. A frequency domain structure is made up of a one-dimensional array where each element of the array contains the domain identity, number of channels in the domain and an array containing the set of channels belonging to the domain. The domain identity and domain size values are both integers. Figure 4.6 gives an example of how the domain data in figure 4.14 would be stored.

Domain identity	Domain size	Channels
0	15	1 2 3 4 5 6 7 8 9 20 21 22 23 24 25
1	5	4 5 6 7 8
2	3	20 21 22
3	7	2 3 4 5 23 24 25

Figure 4.6: Frequency domain data structure.

The structure is used when a transmitter's channel is selected at random. The software checks that a domain file is in use, if true then a channel is selected from the respective frequency domain array. Domain file data is not updated during runtime.

4.2.4 Assignment Structure

The frequency assignment is stored in a one-dimensional array that is indexed by transmitter identity. Therefore, this consists of a list of the transmitters with a channel number in the respective array position. Figure 4.7 gives an example of the structure using the assignment presented in figure 4.15.

Transmitter	Channel
1	4
2	9
3	2
4	1
5	10
6	10
7	1
8	4
9	11
10	4
11	13

Figure 4.7: Assignment data structure.

The frequency assignment structure is constantly accessed and updated during runtime.

4.3 Cost Function For Binary Constraint Based Problems

The cost function looks at the current assignment and simply sums each transmitter's violated constraints. If a constraint is violated (i.e. the number of channels separating two transmitters is equal to or less than that stated in the constraint) then a penalty of 1 is added to the total cost. The software uses an updating cost function technique, this only involves evaluating the change in cost made by the current move.

4.3.1 Cost Function Data Structure

The violation data structure is used to store all the constraint violations in a way that can be accessed quickly and efficiently by the updating cost function. The data structure consists of two arrays, one two-dimensional and one one-dimensional. The two-dimensional array is the cost table containing the constraint violations for all pairs of transmitters. The size of the array is defined as |T|x|T|, where |T| is the number of transmitters. It is therefore symmetrical as constraint violations are counted for both directions. For example, if transmitter 1 had its channel separation constraint with transmitter 2 violated then transmitter 2 would have its channel separation constraint with transmitter 1 violated. The array contains two possible values:

- 0: No constraint or no violation;

- 1: Constraint violation.

When a transmitter has changed channel the cost function only evaluates the constraints that affect that transmitter. To do this the algorithm looks through the relevant row for the transmitter in question and updates its violation costs, if an update is made to position (i,j) the position (j,i) is also updated. This process is illustrated in figure 4.8 with a transmitter's row and column being outlined.

$$\begin{array}{ccc|c|cccc}
0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \\
1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\
0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\
\hline
1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\
\hline
0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\
1 & 1 & 0 & 0 & 0 & 1 & 1 & 0
\end{array}$$

Figure 4.8: Cost table.

The one-dimensional array contains the total constraint violations for each row, indexed by transmitter identity. The values in this array are initially filled by summing each row of the cost table. When updating, each element in the cost array is modified by subtracting the change in cost from the current value. Figure 4.9 gives the cost array for the cost table given in figure 4.8.

Transmitter	Row total
1	3
2	4
3	3
4	4
5	2
6	4
7	2
8	4

Figure 4.9: Cost array.

It can be seen that the total number of violated constraints for this problem is 26.

Figure 4.10 presents some pseudocode of the cost function data structures updating process. When a new move is made by the algorithm the cost is evaluated and the cost evaluation process updates the data structures. If the new cost is an improvement the move is accepted and the data structures remain the same. If there is no improvement in the cost however, the data structures must be changed back to the state they were before the move was made. This is done by retaining copies of both the assignment array and the cost array that were generated before the new move was made. The structures reverts to these copies. The cost table is handled slightly differently, with the assignment array and cost array reverting to their original configuration, the cost table only needs to be updated where it was changed, i.e. the transmitter that had its channel changed. Therefore a process is run that evaluates the original values in the cost table for the respective transmitter. This process is more efficient than copying the entire array.

4.4 File Formats

Input files are used to represent the specific frequency assignment problem to be solved. They are used to establish the number of transmitters, receivers, channels and constraints. Further data files may be used to specify the frequency domain and the characteristics of the types of carriers. Constraint and Variable files follow the established CELAR format [8]. This allows existing test data to be used and also permits comparison with existing software for some data. The efficiency of these file formats is important and has been well established (see [8]), therefore the formats will remain the same for this work. For many problems some of the files are quite large so good internal memory management is essential to ensure satisfactory performance.

The file formats discussed in the following sub-sections are required for constraint based approaches. Other file formats will arise as more areas of military FAP's are considered and will be described as they are required.

4.4.1 Constraint File

The constraint file contains the necessary number of channels separation for pairs of transmitters. These constraints model the various forms of interference that were outlined in 1.5. The format is first transmitter, second transmitter, operator and channel separation. A typical constraint file contains two types of constraints:

- Co-site constraints: representing constraints between channels assigned to transmitters that are

```
Select Random Transmitter, T
IF (Domain File Used)
THEN
    Get Domain, $D_T$ from figure 4.3
    Select Random channel, F from figure 4.6
ELSE
    Select Random channel F
END IF
Create Backup of Assignment, copy figure 4.7
Create Backup of CostArray, copy figure 4.9
Assign F to T, $F_T$, store in figure 4.7
RowTotal = 0
LOOP through Transmitters, i
    Get Channel Separation Constraint(T,i), $C_{T,i}$ from figure 4.5
    IF ($C_{T,i} \neq$ -1)
    THEN
        Get channel, $F_i$ from figure 4.7
        channel Difference, $\delta$F = $|F_T$-$F_i|$
        IF ($\delta$F $\leq C_{T,i}$)
        THEN
            CostTable(T,i) = 1, update figure 4.8
            temp = CostTable(i,T)
            CostTable(i,T) = 1, update figure 4.8
            $\Delta$ = 1 - temp
            CostArray(i) = CostArray(i) + $\Delta$, update figure 4.9
            RowTotal = RowTotal + 1
        ELSE
            CostTable(T,i) = 0, update figure 4.8
            temp = CostTable(i,T)
            CostTable(i,T) = 0, update figure 4.8
            $\Delta$ = 0 - temp
            CostArray(i) = CostArray(i) + $\Delta$, update figure 4.9
        END IF
    END IF
END LOOP
CostArray(T) = RowTotal, update figure 4.9
```

Figure 4.10: Pseudocode of cost updating process.

close;[1]

- Far-site constraints: representing constraints between channels assigned to transmitters that are distant.

The most important constraints for far-site interference are co-channel constraints. These represent the situation where a pair of transmitters must not be assigned the same channel. If f_i and f_j are the

---

[1] 10's of metres for combat net radio.

channels assigned to transmitters i and j then this gives rise to constraints of the form $|f_i - f_j| > 0$.

Adjacent channel constraints are used when two transmitters may not be tuned to close channels, otherwise there is still the potential for interference. Constraints arise of the form $|f_i - f_j| > m$, for some value of m, the required number of channels separation.[2]

Co-site constraints involve any pair of channels assigned to transmitters at the same site (or assigned to one transmitter and the transmitter serving a receiver at the same site). The channels must be separated by a certain fixed number of channels. This separation is usually significantly larger than the adjacent channel constraints. These constraints can be written in the form $|f_i - f_j| \geq m$ or in the form $|f_i - f_j| > m-1$.

Figure 4.11 gives an example of a small set of binary constraints in CELAR format.

1	2	>	4
1	3	>	4
1	4	>	4
1	5	>	4
1	6	>	2
1	7	>	2
1	8	>	2
1	9	>	2
2	3	>	4
2	4	>	4
2	5	>	4

Figure 4.11: Binary constraints in CELAR format.

4.4.2 Variable File

The variable file contains the transmitter number and its corresponding domain identity. The domain identity is used in conjunction with the domain file, see section 4.4.3, and identifies the set of channels available for the particular transmitter. The CELAR variable file can be extended to include more variables such as starting channels, fixed channels, or priorities. The extensions are currently not used in the software. The format is transmitter number, domain identity.

---

[2] For far-site interference m is typically 1 or 2 for combat net radio.

Figure 4.12 gives a small example of a variable file.

```
1  0
2  0
3  1
4  1
5  1
6  0
7  0
8  0
9  2
```

Figure 4.12: Variable file.

4.4.3 Domain File

The domain file is used to represent the channels that are available for assignment. The domain information takes into account the carrier tuning step, carrier tuning limits, carrier bandwidth and coordination agreements. A single domain file consists of a single set of channels. Multiple domain files consist of more than one set of channels. A multiple domain file has a set of channels in the first domain and the other domains are subsets of this.

Figure 4.13 gives an example of a domain file, while figure 4.14 gives an example of a multiple domain file. In each row the first element is the domain identity and the second element gives the number of channels in the domain. The remainder of the row is the set of available channel numbers.

```
0 15 1 2 3 4 5 6 7 8 9 20 21 22 23 24 25
```

Figure 4.13: Single domain file

```
0 15 1 2 3 4 5 6 7 8 9 20 21 22 23 24 25
1 5 4 5 6 7 8
2 3 20 21 22
3 7 2 3 4 5 23 24 25
```

Figure 4.14: Multiple domain file

Alternatively, the user interface allows the specification of a fixed number of consecutive available channels, see section 13.2, in which case no domain file is necessary.

4.4.4 Assignment File

The assignment file contains the frequency assignment solution produced by FAPSolver that shares a common format with FASoft [26]. The format of this file consists of a channel column and a transmitter column with headings. The file simply shows what transmitters are assigned to each channel.

The assignment file may be used as a start file for a problem. This may be done without any adjustments to the format. The software is capable of reading this file and using the assignment obtained from the file as an alternative to a random start. This has the benefit of being able to apply different algorithms to a partially optimised solution as well as applying different parameters to a previously used algorithm that has not produced the desired solution.

Figure 4.15 gives an example of the format of the assignment file for a small problem with 11 transmitters and 9 channels. The format is channel number, number of transmitters assigned to channel, separator (i.e. |) and a list of transmitter identities assigned to the channel.

```
frequency        |  transmitters
    1       2    |  4 7
    2       1    |  3
    3       0    |
    4       3    |  1 8 10
    9       1    |  2
   10       2    |  5 6
   11       1    |  9
   12       0    |
   13       1    |  11
```

Figure 4.15: Assignment file

Chapter 5

Intermodulation and Spurious Constraints

5.1 Overview

This chapter discusses the use of complex co-site constraints to handle intermodulation and spurious emissions and responses. The most common co-site constraint is a binary frequency separation constraint with a larger separation required than is typical for far-site constraints. However, other constraints do arise for co-sited transmitters. Intermodulation and spurious emissions and responses arise in co-sited groups of mobile or stationary ground or maritime terminals, such as tanks, military personnel, trucks, ships or headquarters. Research performed in [41] addressed the avoidance of interference arising from intermodulation products and spurious emissions and responses for co-sited radios in a network. The methods derived from this research form an integral part of the generic model presented here, but it was regarded as unnecessary for them to be implemented. It has already been shown in [41] that the complex co-site constraints provide only an acceptable loss in performance in frequency assignment systems.

5.2 Co-Site Spurious Emission and Response Constraints

Various spurious emissions on frequencies other than the frequency assigned to a transmitter may occur. These emissions may deny frequencies to other radios, although they are generally not powerful enough to affect far-sited radios. The nature and strength of the emissions may depend on the type and power of the radio. A receiver can also respond spuriously to signals on frequencies other than the frequency on which it is attempting to receive. If interference is to be avoided, the responses will deny frequencies to other radios. Different radio types will be susceptible to different incoming frequencies, and some spurious receptions can also depend on the tuned frequency of the radio.

The constraints generated by consideration of the spurious emissions and responses can be taken to be of the form:

$$\mathrm{a}f(t_i) + \mathrm{b} \neq f(t_j) \tag{5.1}$$

for some constants a and b that depend on the radio type and power. Each such formula only applies if there is at least one pair of radios, with one transmitter t_i and a receiver of the other transmitter t_j being co-sited. Note that $\neq$ in equation (5.1) needs to allow some margin, and can be taken to read "differs by at least 10kHz," for example.

Figure 5.1 presents an example of a and b values for a given radio type and power. These values are read from a spurious constraint file and are used in equation (5.1) for pairs of co-sited radios.

$$\mathrm{a, b} = \begin{array}{cc} 2 & 0 \\ 0 & 41.0 \\ 1 & 0.005 \end{array}$$

Figure 5.1: Examples of a and b values for a given radio type and power.

5.3 Co-Site Intermodulation Product Constraints

Intermodulation products arise when two or more signals from different radios produce a new signal. The intermodulation product constraints arise at co-sited transmitters and involve a linear combination of assigned frequencies, taking the form:

$$f(t_j) \neq \sum_{i=1}^{s} c_i f(t_i) \tag{5.2}$$

where t_j is any transmitter serving a receiver co-sited with the s transmitters involved on the right hand side of equation (5.2). Thus transmitter t_j is denied a frequency as a result of this product.

The intermodulation product described by equation (5.2) is referred to as an s-signal product of order $\sum_{i=1}^{s} |c_i|$. The exact set of intermodulation product constraints to use depends on how closely the radios are co-sited. Typically, the largest manageable set contains up to three signals, and is of fifth order.

As with the spurious constraints, the intermodulation product constraints can be read in from a file. The file shown in figure 5.2 contains intermodulation product constraints for a maximum of three transmitters that block a frequency for a receiver.

T_i	T_j	T_k	Site
3	1	-1	1 4 7
2	-1	0	1 4 7 8 12
1	-1	1	1 4 7 12

Figure 5.2: Intermodulation product co-site constraints for a maximum of three transmitters and a receiver.

The first three columns represent the intermodulation product constraints. The fourth column gives the list of sites to which the product applies (defined in another file). Thus at sites 1, 4, and 7 for example, $f(T_l) \neq 3f(T_i) + f(T_j) - f(T_k)$ where the site contains transmitters T_i, T_j and T_k and a receiver of T_l. The order of the intermodulation products is defined by $\sum_{i=1}^{s} |c_i|$, thus the orders of the intermodulation products represented in figure 5.2 are fifth, third and third respectively. As with the spurious constraints, these formulae need to allow some margin, and $\neq$ can again be taken to read "differs by at least 10kHz," for example.

Chapter 6

Aggregated Channel Carriers (AGCC)

6.1 Overview

Aggregated channel carriers are used in the case when a single frequency channel is insufficient for the data to be transmitted. Therefore, the data are split over a number of channels. Aggregated channel carriers are effectively a single pseudo carrier made up of several carriers in a particular grouping of channels. The grouping is represented by a fixed channel separation between each pair of carriers. The number of carriers in a group along with the channel separations may vary considerably. Essentially the receiver design implies maximum and minimum separation on the channels.

The Aggregated Channel Carriers are represented by equality constraints, and their defined channel separations must be maintained throughout. Hence the problem arises of how the groups should be assigned and once assigned how the groups will move. The Aggregated Channel Carrier groups may be assigned in either a left-to-right or right-to-left direction, or just one of these directions if required. The difficulty to overcome with the inclusion of the Aggregated Channel Carriers is to deal with the high costs produced when the groups are assigned new channels. These high costs might inhibit movement of the groups during runtime. To overcome this a weight is specified for the movement of the groups by the user prior to runtime. This weight allows for the acceptance of the Aggregated Channel Carrier moves when some cost increase is present. This weight is explained in full later in this chapter. With the inclusion of Aggregated Channel Carrier groups the algorithms and software should suffer little or no loss in performance.

6.2 Cost Function

Evaluating the cost of moving an AGCC group is not a difficult process. The process simply consists of evaluating the constraint violations of all the individual carriers in the AGCC group and updating the total.

It is clear that if more than one carrier is moved at once the probability of a drop in cost is very small. Therefore a technique is derived to allow the movement of an AGCC group to be accepted during the early stages of the simulated annealing algorithm (see section 2.3). The simulated annealing algorithm allows for cost increasing moves in its early stages, but the probability of these moves being accepted is still relatively low. Therefore the probability must be increased when an AGCC group is moved. As this probability is proportional to the cooling temperature it will decrease over time, therefore allowing the movement of the AGCC groups to settle down. The probability function is given by:

$$\text{prob} = e^{\frac{-\Delta E}{t X_1}}, \tag{6.1}$$

where X_1 is the weight that increases the acceptance probability for AGCC groups. The value of X_1 is user defined. This approach is novel and designed specifically for the assignment of AGCCs using the simulated annealing algorithm.

6.3 Constraint File Amendments

Aggregated channel carriers are represented by equality constraints. Equality constraints were used in the CALMA project [19], but only to represent a fixed separation between pairs of transmitters, therefore the representation of AGCC groups is a novel approach. AGCC groups may consist of two or more carriers encompassing various channel separations. Frequency assignment problems that involve AGCC must use an amended constraint file that includes the equality constraints. Figure 6.1 gives an example of an amended constraint file that contains an AGCC.

It is clear that an AGCC group consisting of three carriers exists in figure 6.1 The length of the group is six channels, with carriers 1 and 2 being four channels apart, 2 and 3 being two channels, and 1 and 3 being six channels.

1	2	=	4
1	3	=	6
1	4	>	4
1	5	>	4
1	6	>	2
1	7	>	2
1	8	>	2
1	9	>	2
2	3	=	2
2	4	>	4
2	5	>	4

Figure 6.1: Binary constraint file with an AGCC.

6.4 Data Structures

In order to ensure that the equality constraints are never broken, the AGCCs are organised into groups and moved together. This process ensures the carriers are always in the correct sequence. The main problem is the fact that high cost changes may be incurred when more than one carrier is moved at the same time.

A possible alternative would be to assign the carriers individually and to place a high penalty on a broken equality constraint. This option would prove to be inadequate as the probability of a group staying in sequence is very low.

For the software to organise the aggregated channel carriers into groups it must be able to distinguish them from a set of binary constraints. In order to do this a unique data structure containing the groups has been created. The data structure consists of a group leader, the number of other carriers in the group, an array containing the carrier identities and an array containing the channel separations. For simplification, in every case the group leader is the carrier with the lowest identity. Therefore, the group given in figure 6.1 is organised as shown in figure 6.2.

Group leader	1
Number of other carriers	2
Carrier identities	2 3
Carrier separations	4 6

Figure 6.2: AGCC internal organisation.

The following stage is to organise the transmitter data to allow for the AGCC groups. Two elements are added to the variables data structure, a boolean value and an index. Each transmitter has a boolean value identifying whether or not the transmitter is part of an AGCC group, if this value is true then the index value points to the appropriate AGCC group data structure outlined in the previous paragraph and presented in figure 6.2. Therefore, the software can distinguish between a normal request assignment and an AGCC group assignment. Figure 6.3 gives an example of the unique variables data structure when taking into account the AGCC seen in figure 6.2.

Transmitter	Domain	IsAGCC	AGCC group
0	0	F	
1	0	T	0
2	0	T	0
3	0	T	0
4	0	F	

Figure 6.3: Variables data structure incorporating AGCC groups.

For a valid assignment to be made for each AGCC group the frequency domain must be processed. The frequency domain is processed once all files have been read in (i.e. constraint file and variable file). This is a simple process if a domain file is not in use. If this is the case the software is aware that the available frequency domain is in consecutive order, therefore the group may be assigned inside the domain boundaries. However, if a domain file is in use there is a distinct possibility that the frequency domain may contain gaps. To handle this problem two arrays are integrated into the AGCC data structure. These arrays contain lists of valid assignable channels for the AGCC group. Two arrays are needed to allow the AGCC groups to be assigned in both a positive and negative direction. To obtain the two lists the software performs two passes of the frequency domain for each AGCC group, one pass in the positive direction and another in the negative direction. Throughout a pass the software checks for each valid assignment of an entire AGCC group, if the group may be assigned correctly the channel index of the group leader is stored in the relevant list. If an assignment is invalid the channel is ignored. This process removes the need to check for a valid assignment during runtime, significantly improving efficiency. Figure 6.4 shows the entire AGCC data structure for the group presented in figure 6.2.

There is no limit on the number of AGCC groups that may be included in a problem.

Group leader	1
Number of other carriers	2
Carrier identities	2 3
Carrier separations	4 6
Positive channels	0 1 2 3 4 ...
Negative channels	20 19 18 17 ...

Figure 6.4: AGCC full internal organisation.

6.5 Assignment of an AGCC

As mentioned in the previous section AGCC groups may be assigned in either a positive or negative direction. The direction in which the groups are assigned is determined by some random number between 0 and 1. If the number is less than 0.5 then the positive direction is chosen, otherwise the negative direction is chosen.

An AGCC group is always pivoted on its leader. During runtime if any of the carriers in an AGCC group are selected at random to be re-assigned, then the group leader becomes the randomly selected carrier and the group is moved accordingly. If a domain file is used, the group leader is assigned to a channel selected at random from one of its two lists of available channels, depending on the direction of the AGCC group assignment. If a domain file is not specified the group leader is simply assigned to a channel that allows the AGCC group to be assigned validly within the boundaries of the channel domain. Once the group leader is assigned the remaining carriers in the group are assigned in order. This represents a novel approach to the assignment of AGCC groups.

6.6 Test Data

Test data for AGCC groups consists of constraint based problems with the inclusion of a variety of AGCC groups. These groups may consist of different numbers of carriers with wide-ranging channel separations. The performance of an algorithm is based on the movement of the AGCC groups.

Chapter 7

Heterogeneous Wide Band Carriers and CDMA Carriers

7.1 Overview

Wide band carriers exist of various types and bandwidth requirements. As with aggregated channel carriers, movement of a wide band carrier may cause a large increase in cost, so special treatment may be necessary. An aggregated channel carrier may also be a wide band carrier. In chapter 4 all carriers were homogeneous (i.e. all carriers have equally sized bandwidths and were assigned to one channel), but now carriers are heterogeneous. The necessary measurement unit of the spectrum is half the highest common factor of all of the carrier bandwidths. Thus the centre frequency of the band assigned to this carrier can be assigned in terms of this measurement unit.

The main problem with the inclusion of wide band carriers is the fact that there is no established file type to represent wide band carriers. To overcome this problem a file has been designed that allows the software to read in the attributes of the wide band carriers, i.e. the carriers which are wide band and their respective bandwidths. The bandwidths are given as numbers of frequency channels. The assignment of wide band carriers is dealt with by assigning the centre frequency. The algorithms and software should suffer little or no loss in performance.

Some wide band carriers are code-division multiple-access (CDMA) carriers. With CDMA, network stations transmit continuously and together on the same frequency band. Interference exists between the transmissions of the different stations and this interference is resolved by the receiver which identifies the signature of each transmitter. The signature consists of a binary sequence, called a code, which is combined with the information bits at each transmitter. Transmission of the code combined with the information bits requires the availability of a greater radio-frequency bandwidth than that required to transmit the information alone (i.e. a normal request). CDMA carriers are not restricted to satellite systems. Here the assumption is made that several CDMA carriers share the same frequency space, therefore a stack of a given number of CDMA carriers may be assumed to be one pseudo wide band carrier.

7.2 Assignment of a Wide Band Carrier

There are three cases in which the frequency domain may differ, these must be handled in order to validly assign wide band carriers.

- Case 1: Consecutive set of channels specified;
- Case 2: Domain file in use where the wide band carrier is given a selection of channels from the base domain;
- Case 3: Multiple domain file in use where the wide band carrier is given a domain that is not the base domain.

Case 1 is handled by selecting a random channel shown in equation (7.1), where W_t is the bandwidth of carrier t, and f_{min} and f_{max} are the frequency domain boundaries. The wide band carrier has its centre frequency assigned therefore a channel must be selected that does not allow the carrier to overlap the frequency domain limits. This is enforced by equation (7.1).

$$random(f_{min} + \frac{W_t}{2}, f_{max} - \frac{W_t}{2}) \tag{7.1}$$

Case 2 is handled by simply selecting a random channel from a generated list of valid assignable channels (this is fully explained in section 7.5). Case 3 is handled by selecting a random channel from the carrier domain.

With all possible cases handled the wide band carriers will always be assigned validly without any overlapping of the frequency domain. Hence no further checks are needed during runtime, and so there is no loss in performance.

7.3 Cost Function

The cost of a wide band carrier move is simply calculated in the same way as a normal narrow band request. In a similar way to the movement of an AGCC group, the movement of wide band carriers produce high cost changes. The probability of a wide band carrier being moved and producing a reduction in cost is very low. Therefore using the same method seen in section 6.2 the simulated annealing algorithm probability of a wide band carrier move being accepted may be increased. A user defined weight X_2 is used to increase the probability. As before the probability of a wide band carrier move being accepted will decrease in proportion to the temperature, allowing the movement of the carriers to settle.

7.4 Carrier Type File

In the area of radio frequency assignment there is no standard file type to handle the use of wide band carriers. Therefore a new file type has been developed to incorporate wide band carriers into frequency assignment problems. The file contains bandwidth information for any wide band carriers included in a problem. The file may also be extended for many applications.

For the valid assignment of a wide band carrier the carrier must be assigned within the limits of the frequency domain. Therefore the bandwidths are defined as numbers of channels. With this information the software is capable of assigning the carriers correctly. The constraint file can then be used to prevent pseudo carriers from interfering with the wide band carriers by using large channel separations to force any possible interferers to the outer boundaries of the wide band carriers.

The format of the carrier type file simply consists of the transmitter identity and a bandwidth value. Figure 7.1 gives an example of the file format.

A narrow band carrier may be represented by either a bandwidth of one or by not being present in the carrier type file. All transmitters not included in the carrier type file are assumed to be narrow band carriers and are assigned normally.

transmitter	bandwidth
1	10
2	20
4	5
5	5
6	10
7	10
8	20
9	1
27	18

Figure 7.1: Carrier Type file format.

7.5 Data Structures

To incorporate the use of wide band carriers three additions have been made to the unique variables data structure (see section 4.2).

- Boolean value: used to distinguish whether the transmitter is wide band, i.e. true for wide band and false for not;
- Integer value: used to store the number of channels that represent the bandwidth of the wide band carrier;
- Array: used to store valid assignable channels for the wide band carrier if a domain file is used.

Figure 7.2 gives an example of the entire variables data structure with the inclusion of both AGCC groups and wide band carriers.

Transmitter	Domain	IsAGCC	AGCC group	IsWB	Bandwidth	Carrier Domain
0	0	F		F		
1	0	T	0	F		
2	0	T	0	F		
3	0	T	0	F		
4	0	F		F		
5	0	F		T	10	11 12 13 ...
6	0	F		T	5	6 7 8 ...

Figure 7.2: Variables data structure incorporating wide band carriers.

The data structure is designed in a way in which the software can handle problems that contain both AGCC groups and wide band carriers. The structures are organised prior to runtime so as not to constrain efficiency.

The carrier domain array is only used if a domain file has been read in. If this is the case the software performs a pass of the frequency domain (similar to as seen in section 6.4) to determine valid assignable channels. The channels are then stored in the carrier domain array to be accessed at runtime. In the case of a multiple domain file in which a wide band carrier is given a particular domain that is not the base domain (see section 4.4.3) the software assumes that the user has selected specific channels for the wide band carrier. In this case the software will select a channel from this domain at random. If a domain file is not used the software will not use the carrier domain array.

7.6 Test Data

Test data may consist of binary constraint problems with the inclusion of wide band carriers with varying bandwidths. The inclusion of some AGCC groups is also possible. This creates some interesting frequency assignment problems.

Chapter 8

Signal to Interference Ratio for Homogeneous Carriers

8.1 Overview

This section describes the incorporation of Signal to Interference (SIR) calculations into the cost function [46]. This can lead to much more accurate frequency planning, but requires a major modification of the cost function. The SIR is a measure of the signal strength of the wanted received signal relative to the interference present at the receiver point. Each reception point has a serving transmitter with an associated signal strength. Along with the serving transmitter there are also interfering transmitters which also have associated received signal strengths. Problems that use binary constraints essentially only consider the effects of interfering carriers in pairs, while SIR calculations consider multiple sources of interference. The updating of a cost function containing SIR calculations is much more computationally demanding than the updating of a binary constraint cost function, as demonstrated in [21], [25] and chapter 14. Thus the implementation of this part of the work requires in-depth planning.

In order to consider multiple sources of interference, it is possible to generate non-binary constraints [15], each of which involves the channels assigned to more than two carriers. However the number of constraints generated tends to be unrealistically large. An alternative is to replace the binary constraint cost function of the meta-heuristic algorithm by a cost function which measures the extent to which the Signal to Interference Ratio (SIR[1]) at identified reception points is less than a given SIR threshold, proposed in [25] and independently in [9]. Zero cost would then correspond to a satisfactory assignment.

---

[1] the ratio of the average signal strength to the average interference signal strength

8.2 Benefits of an SIR Based Cost Function

An SIR based cost function is clearly more computationally demanding than a binary constraint based cost function, but many solutions produced by solving binary constraints give less satisfactory network coverage than solutions produced using an SIR based cost function [25]. Essentially the real problem of frequency assignment is to give an adequate SIR to as much of the network (or area required) as possible. All sources of interference must be considered. Binary constraints are simply a surrogate for these. If a set of binary constraints is solved to zero violations then the solution can no longer be improved by the use of a binary constraint based cost function. An SIR based cost function can still improve the coverage of the network.

A logical approach is to combine the speed and efficiency of the binary constraint based cost function with the more accurate SIR based cost function. It is interesting to speculate whether the hybrid algorithm (see section 8.7.2) or the combined algorithm (see section 8.7.3) will be more effective. A very clear answer emerges in section 14.4 and is also presented in [21].

8.3 Propagation Models

In this section two examples of propagation models are presented, these are essentially the same as those described in [9], [25] and more recently in [21]. These propagation models are used to provide test data to carry out SIR calculation experiments on networks of receivers and transmitters. The model presented in this research is not restricted to the two propagation models presented here. With the use of the generic intermediate file (section 8.8.3) any propagation model may be used. This novel capability allows frequency assignment problems that arise in all areas of the military (e.g. air, land, maritime and satellite) to be optimised using the algorithms presented in this chapter.

8.3.1 d^{-4} Propagation Model

A simple propagation model is used to calculate test data for both the signal and interference signal strengths at any point. It does not take into account terrain or any factors other than a path loss dependent on distance. In practice propagation losses are calculated as part of a pre-processing stage. For terrestrial problems the generation of test data usually makes the assumption that path loss is given by the signal strength divided by the distance between the transmitter and the receiver raised to the power of four. Therefore, considering a receiver point i and a transmitter k, the received signal strength S, proportional to the power flux density, of the transmitter is given by:

$$S_{i,k} = \frac{P_k}{d_{ik}^4} \tag{8.1}$$

where P_k is a value that may be specified to represent transmitted power and antenna gains. All powers and gains are assumed to be equal for test data. d_{ik} is the distance between transmitter T_k and receiver r_i. For simplicity, P_k is assumed to be 1 for all transmitters. The total interference $I_{i,k}$ at receiver point i is the sum of all the interfering received signal strengths multiplied by some value $\theta_{j,k}$ that is dependent on the channel separation of the transmitter k and transmitter j, where k is the wanted transmitter and j is the unwanted interfering transmitter. $I_{i,k}$ is thus given by:

$$I_{i,k} = \sum_{j=1, j\neq k}^{N_t} S_{i,j}\ \theta_{j,k} \tag{8.2}$$

where N_t represents the total number of transmitters serving or interfering with each receiver. In the following $\theta_{\delta f}$=$\theta_{j,k}$ is typically as shown in equation (8.3) for specific channel separations where δ f = $|f_j$-$f_k|$ is the modulus of the difference (in channels) of the channel assigned to j and the channel assigned to k.

$$\theta_{\delta \mathrm{f}} = \begin{cases} 1, & \delta\mathrm{f} = 0 \\ 0.031623, & \delta\mathrm{f} = 1 \\ 0.001, & \delta\mathrm{f} = 2 \\ 0.0001, & \delta\mathrm{f} = 3 \end{cases} \tag{8.3}$$

Other typical figures for GSM are given in [25]. When the channel separation exceeds 3 the interference will be assumed to be negligible. Other values may be used in different circumstances.

The total SIR at a reception point i is given by:

$$SIR_i = \frac{S_{i,k}}{\sum_{j=1,j\neq k}^{N_t} S_{i,j}\ \theta_{j,k}} \tag{8.4}$$

8.3.2 $|sin(x)/x|^2$ Distribution Propagation Model

Another example of a propagation model has been used for satellite assignment problems. Here the received signal strength is not a function of distance travelled by the signal but of directivity. Specifically, at a distance x from the centre of the beam the received signal strength is realistically represented by $|\frac{\sin(cx)}{cx}|^2$ for some constant c, where x is the distance from the beam centre. The overlapping beams give rise to a cluster of hexagonal cells as shown in figure 8.1. A demand vector describes the number of carriers in each beam. With large demand vectors and cell clusters this can give rise to very complex problems.The receiver test points are considered to be the vertices of the hexagonal cells, with each vertex being served by up to three beams. It is usually assumed that the signal strength at the edge of the beam is half that at the centre. Thus if the vertices are at distance 1 from the centre, c can be taken to be 1.391557378 (see figure 8.2).

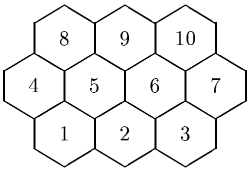

Figure 8.1: An example of a cluster of hexagonal cells

8.4 Binary Constraint Generation

Binary constraints may still be useful when solving frequency assignment problems using SIR calculations, as they may be solved quickly and efficiently. These may be pre-specified (e.g. channel separation constraints for co-sited transmitters) or determined for a pair of transmitters by an SIR

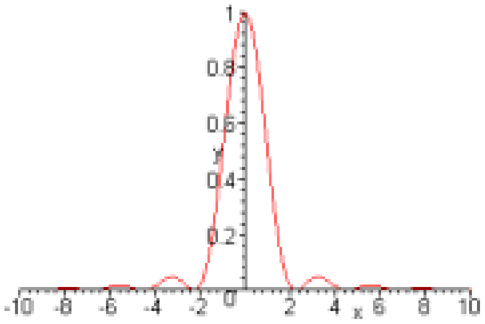

Figure 8.2: $|sin(x)/x|^2$ distribution.

calculation. In some cases transmitter frequency separations may be required to avoid certain couplings between equipments. Suppose that a particular required SIR is given. Constraints for all combinations of pairs of transmitters are calculated in turn. The worst case receiver (i.e. the receiver with the smallest potential SIR as calculated by an appropriate propagation model) is used to calculate the required channel separation. The constraints are then produced in CELAR format. Figure 8.3 presents pseudocode for the generation of binary constraints.

For problems that involve AGCC carriers equality constraints must be added manually. Other binary constraints may still need to be added when considering other forms of interference.

8.5 Data Structures

The data structures for SIR calculations are made up of various two-dimensional tables all interlinked. Information must be stored for serving transmitters, interfering transmitters, received and transmitted signal strengths, SIR and interference totals and θ values for pairs of transmitters. This information must be accessed and updated efficiently as the size of the problems may become large. The structures are designed to allow expansion for multiple carrier types with ease.

The first table contains information for serving transmitters. Information required for a serving transmitter consists of the receiver index, serving transmitter index and the transmitted signal strength obtained from the appropriate propagation model or otherwise (e.g. intermediate file). Figure 8.4 presents an example of this structure, where R is the receiver index, T is the serving transmitter

```
Loop through Receiver points
    Get wanted transmitter signal strength, $T_t$
    Loop through Interfering transmitters
        Get interfering transmitter signal strength, $T_n$
        SIR = $\frac{T_t}{T_n}$
        Channel separation, $D_f = 0$
        While (SIR < Constraint SIR Threshold)
            $D_f = D_f + 1$
            If ($D_f = 1$)
                $Tnew_n = T_n \times \theta_1$
                SIR = $\frac{T_t}{Tnew_n}$
            ElseIf ($D_f = 2$)
                $Tnew_n = T_n \times \theta_2$
                SIR = $\frac{T_t}{Tnew_n}$
            ElseIf ($D_f = 3$)
                $Tnew_n = T_n \times \theta_3$
                SIR = $\frac{T_t}{Tnew_n}$
            ElseIf ($D_f \geq 4$)
                SIR = 1000000
        End While
        If ($D_f$ > current channel separation)
            Store $D_f$
    End Loop
End Loop
```

Figure 8.3: An Example of Binary constraint generation using the d^{-4} propagation model.

index and S is the signal strength at the the respective receiver point. This structure is linked to the transmitter information structure by the serving transmitter identity field.

R_i	T_i	S_i
R_0	T_3	S_0
R_1	T_7	S_1
R_2	T_8	S_2
R_3	T_4	S_3
R_4	T_3	S_4
R_5	T_2	S_5
...	...	...
R_N	T_N	S_N

Figure 8.4: Serving transmitter information data structure.

The next structure is an array that contains the signal strengths of all of the interfering transmitters for each receiver point. The dimensions of the array are |number of receivers|×|number of transmitters|.

Every transmitter in the network is a potential interferer for each reception point except when the transmitter is tuned to the receiver, if this is the case the signal strength of the wanted transmitter is set to -1 in the table. Figure 8.5 presents an example of the interfering transmitter signal strength table, where R is the receiver point index, T is the transmitter index and S is the signal strength.

	T_0	T_1	T_2	T_3	...							T_N
R_0	$S_{0,0}$	$S_{0,1}$	$S_{0,2}$	-1	...							$S_{0,N}$
R_1	$S_{1,0}$	$S_{1,1}$	$S_{1,2}$	$S_{1,3}$	...							$S_{1,N}$
R_2	$S_{2,0}$	$S_{2,1}$	$S_{2,2}$	$S_{2,3}$	...							$S_{2,N}$
R_3	$S_{3,0}$	$S_{3,1}$	$S_{3,2}$	$S_{3,3}$	...							$S_{3,N}$
...												
R_N	$S_{N,0}$	$S_{N,1}$	$S_{N,2}$	$S_{N,3}$	...							$S_{N,N}$

Figure 8.5: Interfering transmitter signal strength table.

When the SIR is calculated for a particular reception point the associated transmitter signal strength is derived from figure 8.4 and then the interfering transmitter signal strengths are derived from figure 8.5.

The following structure is an array that contains the rejection factors (i.e. the θ values, see equation (8.3)) for all pairs of transmitters. The dimensions of the array are |number of receivers|×|number of transmitters|. The array is filled when the initial SIR cost checking of an assignment is performed (see section 8.6), from then on smaller updates are made depending on the new assignment. Serving transmitter array positions contain a -1 as their rejection factor is not required. The rejection factors are explained in section 8.3. Figure 8.6 presents an example of the rejection factor table, where R is the receiver point index, T is the transmitter index and θ is the rejection factor.

Once a new assignment has been generated the rejection factors for the receiver points are updated by examining the channel assigned to the respective serving transmitter of each receiver point and then comparing it to the channel of each interfering transmitter. The rejection factors are then evaluated using equation (8.3) or any other method specified and then stored in the table. The rejection factors are then used to scale the signal strengths of the interfering transmitters. For efficiency the table is not entirely updated during the cost updating process. The cost updating process only updates

	T_0	T_1	T_2	T_3	...							T_N
R_0	$\theta_{0,0}$	$\theta_{0,1}$	$\theta_{0,2}$	-1	...							$\theta_{0,N}$
R_1	$\theta_{1,0}$	$\theta_{1,1}$	$\theta_{1,2}$	$\theta_{1,3}$	...							$\theta_{1,N}$
R_2	$\theta_{2,0}$	$\theta_{2,1}$	$\theta_{2,2}$	$\theta_{2,3}$	...							$\theta_{2,N}$
R_3	$\theta_{3,0}$	$\theta_{3,1}$	$\theta_{3,2}$	$\theta_{3,3}$	...							$\theta_{3,N}$
...												
R_N	$\theta_{N,0}$	$\theta_{N,1}$	$\theta_{N,2}$	$\theta_{N,3}$	...							$\theta_{N,N}$

Figure 8.6: Rejection factor table.

the parts of the table that are relative to the transmitter that has been assigned a new channel. If the transmitter serves the receiver then the rejection factors between the serving transmitter and interfering transmitters need to be re-evaluated, therefore the entire row is updated. Alternatively, if the transmitter is not the serving transmitter for the receiver only the rejection factor between the transmitter and the serving transmitter must be re-evaluated, therefore only one value is updated. Pseudocode is presented in the next section giving a description of the cost updating process.

The final structure is an array that contains the SIR values at each receiver point and also the total interference values for each of the reception points. These values are solely used for cost updating. Figure 8.7 presents an example of this structure, where R is the receiver point index, SIR is the SIR for the receiver point and I is the total interference at the receiver point.

R_i	SIR_i	I_i
R_0	SIR_0	I_0
R_1	SIR_1	I_1
R_2	SIR_2	I_2
R_3	SIR_3	I_3
R_4	SIR_4	I_4
R_5	SIR_5	I_5
...	...	...
R_N	SIR_N	I_N

Figure 8.7: SIR and Interference values for receiver points.

The structures are all linked via reception point indices and transmitter indices and are heavily used in the SIR cost updating process explained in the next section. The structures have the potential to become very large and as SIR calculations are computationally demanding it is essential that these structures can be accessed and updated efficiently.

8.6 Cost Function

The SIR at the receiver points is determined and compared with the required threshold. The cost function involves any deficits in the SIRs at reception points and this cost is zero if all receiver points have a satisfactory SIR. To achieve coverage at a reception point, one or more of the wanted signals from one or more of the wanted transmitters must meet the required SIR threshold, denoted by σ, when the total interference at that point is taken into account. E is the sum of the SIR deficits to the power of a, given by:

$$\mathrm{E} = \sum_{i=1}^{N_r} (\sigma - SIR_i)^a \tag{8.5}$$

where terms are only included if they satisfy, $SIR_i < \sigma$. Once $E = 0$ the network has achieved 100% coverage. The value $a = 2$ has been used in this work. An alternative is to simply maximise coverage (the percentage of receiver points with a satisfied SIR). The advantage of equation (8.5) with $a = 2$ is that large deficits are avoided, and communication may still be possible with small deficits. Equation (8.5) has been shown to work well in [9], [25], [46] and more recently combined with a binary constraint violation cost in [21].

Figure 8.8 presents pseudocode for the SIR cost updating process giving an insight into how the data structures are used.

```
New assignment generated, change of channel $(F)$ for transmitter $T_x$
LOOP through Receiver points, $R$
    IF $T_x$ serves $R$
    THEN
        $Wanted = T_x$
        Get channel of serving transmitter, $F_{Wanted}$
        LOOP through transmitters, $T$
            Get channel of interfering transmitter, $F_{Interferer}$
            Channel difference = $|F_{Wanted}$-$F_{Interferer}|$
            Get $\theta_{Wanted,Interferer}$ and store in figure 8.6
            Get Interferer signal strength, $S_{Interferer}$ from figure 8.5
            Interference, $In_{TOTAL} = In_{TOTAL} + (S_{Interferer} \times \theta_{Wanted,Interferer})$
        END LOOP
        Get Server signal strength, $S_{Wanted}$ from figure 8.4
        $SIR_R = \frac{S_{Wanted}}{In_{TOTAL}}$
        Store $In_{TOTAL}$ in figure 8.7
        Store $SIR_R$ in figure 8.7
    ELSE IF $T_x$ is potential interferer at $R$
    THEN
        $Interferer = T_x$
        Get serving transmitter index, $Wanted$ from figure 8.4
        Get Old Interference total, $In_{TOTAL(OLD)}$ from figure 8.7
        Get Old $\theta_{Wanted,Interferer}$, $\theta_{Wanted,Interferer(OLD)}$ from figure 8.6
        Get channel of serving transmitter, $F_{Wanted}$
        Get channel of interfering transmitter, $F_{Interferer}$
        Channel difference = $|F_{Wanted}$-$F_{Interferer}|$
        Get $\theta_{Wanted,Interferer}$ and store in figure 8.6
        Get Interferer signal strength, $S_{Interferer}$ from figure 8.5
        Old Interference $In_{OLD} = S_{Interferer} \times \theta_{Wanted,Interferer(OLD)}$
        New Interference $In_{NEW} = S_{Interferer} \times \theta_{Wanted,Inteferer}$
        $\Delta$In = $In_{NEW}$ - $In_{OLD}$
        $In_{TOTAL} = In_{TOTAL(OLD)} + \Delta$In
        Get Server signal strength, $S_{Wanted}$ from figure 8.4
        $SIR_R = \frac{S_{Wanted}}{In_{TOTAL}}$
        Store $In_{TOTAL}$ in figure 8.7
        Store $SIR_R$ in figure 8.7
    END IF
END LOOP
```

Figure 8.8: Pseudocode for SIR cost update process.

Required SIR values are often given in decibels (dBs), but when using the cost function, equation (8.5), these dB values must be converted to ratios. The conversion can be accomplished using the formula:

$$\sigma_{ratio} = 10^{\frac{\sigma_{dB}}{10}} \tag{8.6}$$

where σ_{dB} is the required SIR in dB and σ_{ratio} is the required SIR as a ratio. Table 8.1 gives the dB values converted into SIR ratios of received signal strength.

σ_{dB}	σ_{ratio}
9	7.943
10	10
11	12.589
12	15.849
13	19.953
14	25.119
15	31.623
16	39.811
17	50.119

Table 8.1: Conversion table for dB to ratios.

8.6.1 SIR Cost Function for Aggregated Channel Carriers

The cost function for an AGCC involves performing an update for every carrier in the group, as each carrier in the group changes channel. If a move is rejected the updating process is repeated for the AGCC's previous channel assignment.

8.7 Algorithms

The algorithms presented in this section are used to optimise frequency assignment problems using a weighted linear combination of an SIR based cost function and a binary constraint violation cost function.

8.7.1 Pure SIR Algorithm

The pure SIR algorithm uses a cost function that involves only SIR cost, this has been used in [9], [25] and [46], and more recently in [21]. This algorithm does not use any binary constraints. The algorithm may begin from a random start or from a previous assignment.

8.7.2 Hybrid Algorithm

The hybrid algorithm is a novel approach that uses a cost function that combines SIR cost and binary constraint cost. Specifically, the overall cost is $W_1\mathrm{E} + W_2\mathrm{C}$, where W_1 and W_2 are weights, E is the SIR cost and C is the sum of the binary constraint violations. Both the SIR calculations and binary constraints can be weighted in the cost function. This allows for more or less emphasis on either component. The binary constraints may be generated at any SIR threshold.

8.7.3 Combined Algorithm

The combined algorithm is a novel approach that consists of two parts. The first part produces a solution by solving a set of binary constraints only. This solution is then used as a starting point for the pure SIR algorithm. Solutions to binary constraint problems may be obtained quickly but once all the binary constraints are solved the solution can no longer be improved by a binary constraint solving algorithm. In some cases with a sufficiently large frequency domain no improvement is necessary, but when improvement is needed the only possibly way of achieving this is through the use of an SIR algorithm. A logical approach is to generate the binary constraints at an SIR threshold above the required SIR threshold. Good values appear to be 2dB or 3dB above the SIR threshold. See section 14.4 for more information about appropriate choices.

8.8 File Formats

SIR problems require transmitter and receiver information. Transmitter and receiver information may consist of coordinates for transmitter and receiver points. Receiver information also contains serving transmitter numbers. This information is used to generate transmitter and receiver signal strengths by a propagation model.

An intermediate file has also been created that does not require any coordinate information. This file contains received signal strength information for each transmitter at each reception point. It is not specific to any propagation model. The file is simply a list of receivers with their respective serving transmitter specified and the received signal strength given for each serving and interfering transmitter.

SIR problems also require a variable file and may also use a constraint file, domain file and a start file (see section 4.4).

8.8.1 Receiver File

The receiver file contains receiver point coordinates, receiver number and serving transmitter number. Receiver points are repeated depending on the number of transmitters they are tuned to. Figure 8.9 gives an example of a receiver file format.

```
% receiver coordinates
% xmin xmax ymin ymax : 0 10000 0 10000
% transmitter file : testf3.trn
% parameter file : testf3.prm
% format: x y rec_num serving_trans_num
   74.52    3837.99   1   2
   74.52    3837.99   1   4
   74.52    3837.99   1   5
 1011.52    2081.17   2   1
 1011.52    2081.17   2   2
 1011.52    2081.17   2   5
 1310.07    1703.39   3   1
 1310.07    1703.39   3   5
 1310.07    1703.39   3   7
 1473.27    6915.45   4   3
```

Figure 8.9: Example of a receiver file format used by a propagation model.

The software ignores the first five lines of the file as they have no relevance to the solving of an SIR problem; they simply give information about the problem and the origin of the data. Note that the file shown gives three instances of the first receiver with different serving transmitters.

8.8.2 Transmitter File

The transmitter file contains the transmitter coordinates and the transmitter identities. Figure 8.10 gives an example of a transmitter file format.

```
% transmitter coordinates
% xmin xmax ymin ymax : 0 10000 0 10000
% parameter file : testf3.prm
% receiver file : testf3.rec
% format: x y trans_num
     93.82     2135.47     1
    170.88     2453.25     2
    912.22     8961.12     3
   1035.15     4839.97     4
   1170.82     2986.57     5
   1473.93     9908.23     6
   1747.25      488.97     7
   3041.67     8343.62     8
```

Figure 8.10: Example of a transmitter file format used by a propagation model.

As with the receiver file the first five lines of the file are ignored.

8.8.3 Intermediate File

The intermediate file is a novel proposal designed for this work. It is an input file that allows the possibility of alternative propagation models being used. The file contains the receivers, wanted transmitters, interfering transmitters and the associated received signal strength for each transmitter. The use of the intermediate file eliminates the need for the receiver file, transmitter file and the d^{-4} propagation model and allows for the use of any external propagation models. Figure 8.11 gives the format of the intermediate file.

Each receiver has a wanted transmitter with an associated received signal strength. Each receiver also has a list of interfering transmitters with their associated received signal strength, normally consisting of every transmitter in the network. It is acceptable to only specify transmitters with a high potential for interference. The end of the information for a receiver is represented by a full stop.

Receiver point R1: Serving Trans T97 received signal strength;
Interfering Trans T66 received signal strength Interfering Trans T72 received signal strength
Interfering Trans T93 received signal strength . . . Interfering Trans TN received signal strength.
Receiver point R2: Serving Trans T62 received signal strength;
Interfering Trans T28 received signal strength Interfering Trans T37 receiver signal strength
Interfering Trans T123 received signal strength . . . Interfering Trans TN received signal strength.
. . .
Receiver point RN: Serving Trans T97 received signal strength;
Interfering Trans T66 received signal strength Interfering Trans T72 receiver signal strength
Interfering Trans T93 received signal strength . . . Interfering Trans TN received signal strength.

Figure 8.11: Intermediate file format.

The benefit of the intermediate file is that alternative models may be more accurate than any model specified here, and are easily implemented. When the intermediate file is used the positions of the transmitters and receivers are not needed.

Chapter 9

Spectral Overlap Functions for Heterogeneous Carrier Types

9.1 Overview

The previous chapter discussed SIR calculations for narrow band homogeneous carrier types. To allow for wide band heterogeneous carrier types the model must be expanded. Wide band carrier types were introduced in chapter 7. The rejection factors (θ values) discussed in the previous chapter must be calculated differently for pairs of wide band carriers. Each wide band carrier has an associated spectral function. If a wide band carrier is assigned to a channel that is close to another, then the spectral function may overlap with the filter bandwidth of a receiver of the second. The maximum total area of overlap of the two transmitter/receiver cases is then assumed to be the rejection factor or θ value. The receiver filter will always have a rectangular distribution with a specified bandwidth. Two spectral overlap functions are presented in this chapter.

Further topics discussed in this chapter include the necessary modifications to the carrier type file, a generic look-up table and intermodulation in satellite problems.

9.2 Spectral Overlap Function 1

To handle both single carrier types and multiple carrier types a novel process has been developed to evaluate the off tune rejection values for pairs of carrier types. This process takes into account the transmitter spectral function and the receiver filter bandwidth. This simplest case assumes a rectangular distribution for the transmitter spectral function. The rejection factor is evaluated using the area of overlap between the interfering transmitter spectral function and the receiver filter function using the frequency separation between the two centres. The function must take two cases into consideration, when the receiver filter bandwidth is larger than the transmitter bandwidth and vice versa. The two cases are given as follows:

Case 1: $W_r \leq W_t$

$$\text{If } \left(|\delta \mathrm{f}| \leq \frac{W_t - W_r}{2}\right), \qquad \theta = \frac{W_r}{W_t};$$

$$\text{if } \left(\frac{W_t - W_r}{2} < |\delta \mathrm{f}| \leq \frac{W_t + W_r}{2}\right), \qquad \theta = \frac{W_r + W_t}{2W_t} - \frac{|\delta \mathrm{f}|}{W_t};$$

$$\text{if } \left(|\delta \mathrm{f}| > \frac{W_t + W_r}{2}\right), \qquad \theta = 0.$$

Case 2: $W_r > W_t$

$$\text{If } \left(|\delta \mathrm{f}| \leq \frac{W_r - W_t}{2}\right), \qquad \theta = 1;$$

$$\text{if } \left(\frac{W_r - W_t}{2} < |\delta \mathrm{f}| \leq \frac{W_t + W_r}{2}\right), \qquad \theta = 1 - \left(\frac{|\delta \mathrm{f}| - \frac{(W_r - W_t)}{2}}{W_t}\right);$$

$$\text{if } \left(|\delta \mathrm{f}| > \frac{W_t + W_r}{2}\right), \qquad \theta = 0.$$

In the above W_r and W_t represent the receiver filter bandwidth and the transmitted spectral function bandwidth, respectively.

9.3 Spectral Overlap Function 2

A second more accurate process has been provided by QinetiQ to evaluate the rejection factors. This process uses a raised cosine transmitter spectral function and the receiver filter bandwidth. It is widely used in communication systems. The previous process assumed a rectangular transmitter spectral function. A spectral parameter β is used to define the shape of the transmitter spectral function. This value ranges from zero to one, one giving a rectangular spectrum and zero giving a raised cosine spectrum, illustrated in figure 9.1. Therefore, when β=1 theta function 2 should provide the same rejection factor values as theta function 1.

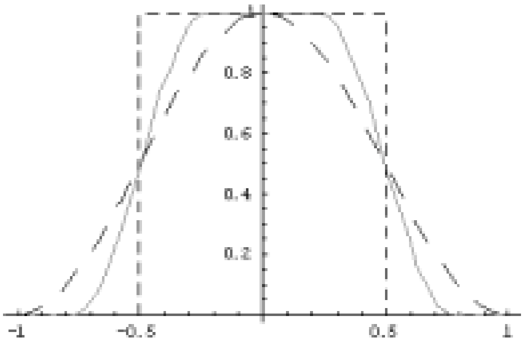

Figure 9.1: Transmitted spectrum shape examples for β=0, β=0.5 and β=1.

The raised cosine function normalised to unit bandwidth and unit area is defined as:

$$\text{If } |x| \leq \frac{\beta}{2}, \qquad y(x) = 1;$$

$$\text{if } \frac{\beta}{2} < |x| \leq \left(1 - \frac{\beta}{2}\right), \qquad y(x) = \frac{1}{2}\left(1 + \cos\left(\frac{\pi\left(|x| - \frac{\beta}{2}\right)}{(1-\beta)}\right)\right);$$

$$\text{if } \left(1 - \frac{\beta}{2}\right) \leq |x|, \qquad y(x) = 0.$$

where,

$$x = \frac{f_x - f_t}{W_t}$$

with f_x representing a frequency from the receiver filter bandwidth and f_t representing the centre frequency of the transmitter.

Also,

$$f_1 = f_r - \frac{W_r}{2} \qquad \text{and} \qquad f_2 = f_r + \frac{W_r}{2},$$

give the receiver filter limits.

The fraction of transmitter power which passes through a rectangular filter at the receiver is thus the integral of the normalised transmitted spectrum between the appropriate limits, found by substituting f_1 and f_2 for f_x giving the two limits x_1 and x_2.

The semi-infinite integral of the raised cosine is given below.

$$\text{If } |x| \le -\left(1 - \frac{\beta}{2}\right), \qquad \int_{-\infty}^{x} y(x)dx = 0;$$

$$\text{if } -\left(1 - \frac{\beta}{2}\right) < |x| \le -\frac{\beta}{2}, \qquad \int_{-\infty}^{x} y(x)dx = \frac{\pi(2-\beta+2x) - 2(1-\beta)\cos\left(\frac{\pi(1+2x)}{2(1-\beta)}\right)}{4\pi};$$

$$\text{if } -\frac{\beta}{2} < |x| \le \frac{\beta}{2}, \qquad \int_{-\infty}^{x} y(x)dx = \frac{1}{2} + x;$$

$$\text{if } \frac{\beta}{2} < |x| \le \left(1 - \frac{\beta}{2}\right), \qquad \int_{-\infty}^{x} y(x)dx = \frac{\pi(2+\beta+2x) + 2(1-\beta)\cos\left(\frac{\pi(1-2x)}{2(1-\beta)}\right)}{4\pi}.$$

$$\text{if } \left(1 - \frac{\beta}{2}\right) < |x|, \qquad \int_{-\infty}^{x} y(x)dx = 1.$$

The fraction of received power may thus be obtained by subtracting two semi-infinite integral values, hence:

$$\theta = \int_{x_1}^{x_2} y(x)dx = \int_{-\infty}^{x_2} y(x)dx - \int_{-\infty}^{x_1} y(x)dx \tag{9.1}$$

9.4 Carrier Type File Amendments

The carrier type file first mentioned in section 7.4 must be expanded to accommodate for receiver filter bandwidths, transmitter powers and β values. So far transmitter powers have been assumed to be 1, but these values may be changed if required using this file. The carrier type file now consists of transmitter identity, transmitter bandwidth, receiver filter bandwidth, transmitter power and β. The receiver filter bandwidth is used when the transmitter is tuned to a receiver. If a transmitter is excluded from this file then it is assumed to be narrow band and is given a bandwidth of 1, a receiver filter bandwidth of 1, transmitter power of 1 and β is not needed. Figure 9.2 gives an example of the carrier type file format.

```
% transmitter and receiver bandwidth and power data
% format : transmitter bandwidth recbw power beta
    1      18      10     1     1
    10      12      12     1     1
    13      28      5      1     0
    24      50      7      1     1
    31      22      5      1     0
    32      44      31     1     1
    50      12      8      1     1
    123      8      3      1     0
    983      2      2      1     1
```

Figure 9.2: Extended Carrier Type file format.

9.5 Test Data

New data have been generated to simulate a geographic area divided up into cells. Hexagonal cells are assumed and the receiver points are considered to be the vertices of the hexagonal cells. Each cell has a demand vector, i.e. a number of transmitters or requests; these demands are positioned at the cell centres. A receiver point may be used for more than one cell depending on its position in the cluster, in a real world situation this may represent two or more cells overlapping. Each transmitter acts as a server to every receiver that surrounds it. With large demand vectors and cell clusters a problem of this type can become extremely large and complex.

9.6 Generic Look-Up Table

A look-up table may be used for defining θ values for pairs of carriers at certain frequency separations. A suggested format is shown in table 9.1.

Trans 1	Trans 2	δf	θ
1	1	1	0.033
1	1	2	0.001
1	2	1	0

Table 9.1: Example of look-up table format.

The look-up table would remove the need to evaluate θ values for pairs of carriers and also removes the dependance on any spectral model, allowing the use of any possible spectral function. It could considerably improve performance. Although on larger problems extremely large data structures would be required to hold all the information (giving rise to memory issues and efficiency problems). On a small scale a look-up table would be advantageous.

9.7 Intermodulation in Satellite Problems

In satellite systems the avoidance of intermodulation products altogether is not feasible, and so their effects must be minimised. This problem is much harder than the co-sited intermodulation problem outlined in section 5.3. Figure 9.3 and figure 9.4 show the build up of the intermodulation products from a random placement of 42 carriers in a hypothetical satellite beam. The vertical scale is in dB and the horizontal scale is the frequency. The figures show the change from a few discrete intermodulation products for a small number of carriers to a uniform *grass* effect and eventually a continuum. Background noise is omitted from the figures.

In [38] a system is described which aims to minimise intermodulation effects for a single satellite beam. The efficient incorporation of these effects into the cost function of a meta-heuristic algorithm for an entire system is challenging and requires much research if intermodulation effects were to be

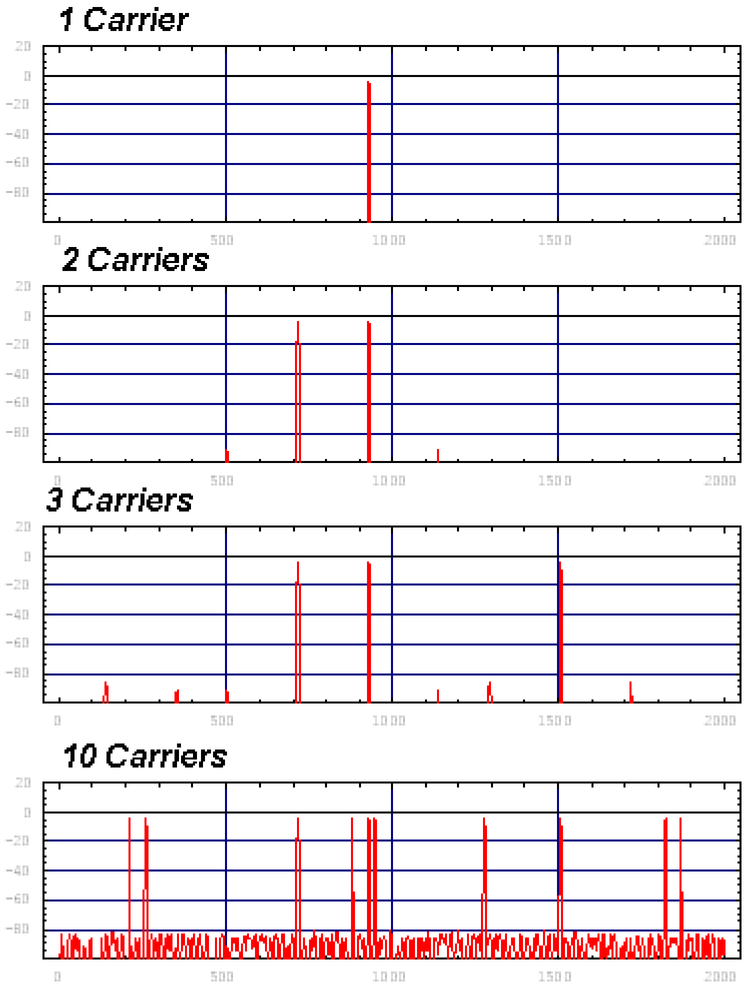

Figure 9.3: Transmitted carriers from a satellite producing intermodulation products. Figure shows intermodulation products produced from 2 carriers, 3 carriers and 10 carriers. This information was provided by QinetiQ.

minimised for multiple satellite beams. This is a significant area for further research.

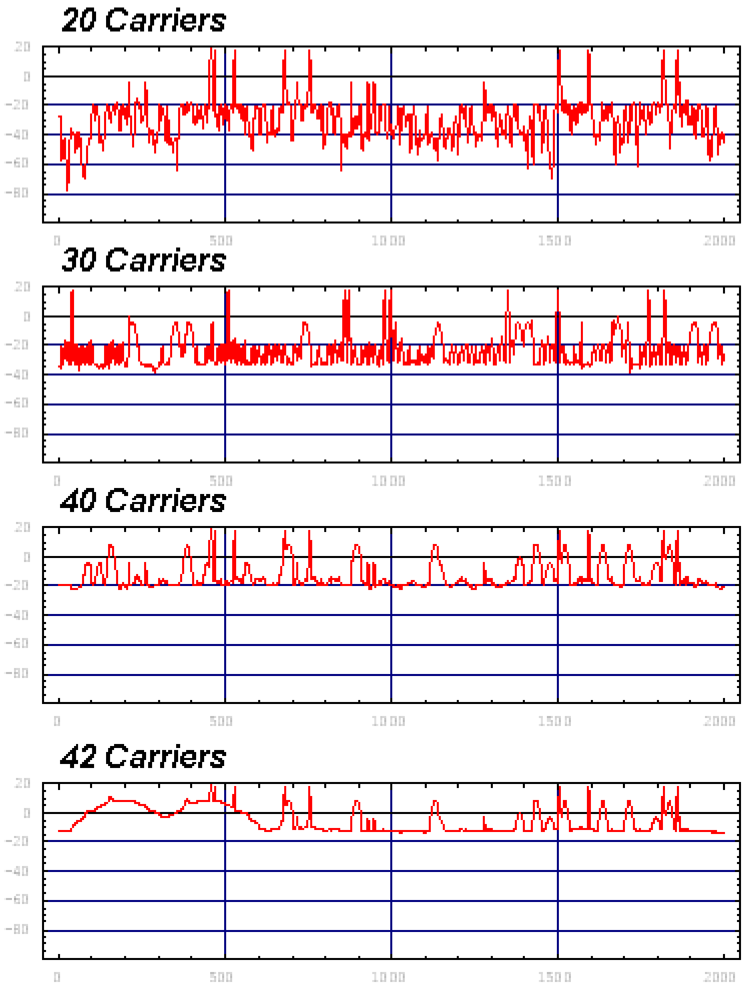

Figure 9.4: Transmitted carriers from a satellite producing intermodulation products. Figure shows intermodulation products produced from 20 carriers, 30 carriers, 40 carriers and 42 carriers. This information was provided by QinetiQ.

Chapter 10

Unsynchronised Frequency Hopping Carriers

10.1 Overview

This chapter presents a detailed description of the assignment of frequency lists in unsynchronised frequency hopping. Much research has been done on frequency hopping for GSM networks [5], [6], [35] and [45]. Frequency hopping in these networks can reduce interference and therefore increase capacity.

The objective of assignment in unsynchronised frequency hopping is to construct and assign lists of frequencies to transmitters. The carrier then hops pseudo-randomly over this list. There is no (or inadequate) synchronisation between the time slots for which a frequency is used in unsynchronised hopping. Frequency hopping leads to improved network performance in practice. The major factors that yield the improvement are known as *frequency diversity* gain and *interference diversity* gain. The frequency diversity [5] gain is obtained by reducing Rayleigh fading [5] of the wanted signal, which occurs when the same signal travels on multiple paths that may cancel each other and lead to a weak received signal. The interference diversity [5] gain arises from the fact that frequency hopping results in a variable interference between cells. Instead of having a constant level of interference between two cells, the interference depends on the frequencies hopped to. The error-control coding copes better with this variable interference. Both types of diversity gain are dependent on the number of frequencies used for hopping; both increase as the number of frequencies increases. Practical observations have led to the conclusion that once the number of frequencies reaches a certain level, further increases do not yield any significant improvement.

For military frequency hopping a model has been designed which can include these considerations from the GSM approach. For GSM networks the main reasons for using frequency hopping are to improve overall usage of the available spectrum and improve network performance. Military frequency hopping shares these aspirations but must consider security issues as well. For military frequency hopping the frequency lists must become large and ideally have a small number of contiguous blocks of frequency channels. Therefore, a frequency list assignment model for unsynchronised frequency hopping carriers has been designed that incorporates the ideas used in GSM systems as well as the extra requirements needed for military systems.

In the algorithms the basic step of "select a transmitter, select a frequency" is replaced by "select a transmitter and add, delete or swap a frequency in its hopping list". The three possibilities are selected with probabilities 0.4, 0.4 and 0.2. These probability values are derived from the work performed in [35] and were shown to work well, hence they are unchanged for the work presented here.

10.2 Cost Function

Measures of co-channel interference between pairs of carriers are scaled down by the probability of a co-channel clash between the two carriers in any time slot. Similarly, measures of adjacent channel interference between pairs of carriers are scaled down by the probability of an adjacent channel clash between the two carriers in any time slot. These probabilities are used to scale down the interference caused by unwanted transmitters.

The SIR cost function for frequency hopping carriers is calculated in a similar way to section 8.6 in the case of narrow band carriers except that the scaling of interfering received signal strengths must be based on an average received signal strength. Therefore the SIR at a reception point i served by transmitter k is given as:

$$SIR_i = \frac{S_{i,k}}{\sum_{j=1,j\neq k}^{N_t} S_{i,j}\ \phi_{j,k}} \tag{10.1}$$

where $S_{i,k}$ is the received signal strength of the tuned transmitter and $S_{i,j}$ is the received signal strength of the interfering transmitter. These values are provided by the appropriate propagation model. The definition of $\phi_{i,j}$ [35] for two lists i and j is given by:

$$\phi_{i,j} = a_{i,j} + (\theta_1 \times b_{i,j}) \tag{10.2}$$

where θ_1 is the adjacent channel factor defined in section 8.3. The values for $a_{i,j}$ [35] and $b_{i,j}$ [35] are calculated using equation (10.3) and equation (10.4), where $0 \leq a \leq 1$ and $0 \leq b \leq 1$.

$$a_{i,j} = \frac{\mathrm{FD} \times \mathrm{ID} \times |\mathrm{list(i)} \cap \mathrm{list(j)}|}{|\mathrm{list(i)}| \times |\mathrm{list(j)}|} \tag{10.3}$$

$$b_{i,j} = \frac{\mathrm{FD} \times \mathrm{ID} \times |\{\mathrm{f{:}\ f} \in \mathrm{list(i)} \cap (\mathrm{f}+1) \in \mathrm{list(j)}\}| + |\{\mathrm{f{:}\ f} \in \mathrm{list(j)} \cap (\mathrm{f}+1) \in \mathrm{list(i)}\}|}{|\mathrm{list(i)}| \times |\mathrm{list(j)}|} \tag{10.4}$$

and the FD and ID values are as explained in the following sections. If FD and ID are equal to 1, $a_{i,j}$ and $b_{i,j}$ would give the probability of a co-channel clash and the probability of an adjacent channel clash respectively.

10.2.1 FD

Frequency diversity [5] measures the effect of reduced Rayleigh fading (see section 10.1), which depends on the list length of the serving carrier. Table 10.1 gives a set of typical values.

10.2.2 ID

Interference diversity [5] measures the extent to which the error control copes better, which depends on the product of the serving carrier and interfering carrier list lengths. Table 10.2 gives a set of typical values.

List Length of Serving Carrier	Frequency Diversity Gain in dB
1	0
2	-2.0
3	-3.2
4	-4.2
5	-5.0
6	-5.6
7	-5.8
8	-6.0
$\geq$9	-6.0

Table 10.1: Typical Frequency Diversity Gain in dB for the Length of the List of the Serving Carrier. The values are taken from [5].

Product of List Lengths	Interference Diversity Gain in dB
1	0
4	-6.1288
9	-6.3788
16	-6.48
$\geq$16	-6.48

Table 10.2: Typical Interference Diversity Gain in dB for the Product of the List Lengths. The values are taken from [5].

10.2.3 Evaluation of $\phi_{i,j}$ for Non-homogeneous Carriers

The generic military frequency assignment model includes non-hopping homogeneous carrier types (comprising narrow band carriers and aggregated channel carriers), non-hopping heterogeneous carrier types (comprising wide band carriers, CDMA carriers and aggregated channel carriers) and unsynchronised hopping carriers. During SIR calculations a combination of any of these types of carriers may arise. Therefore, all possible cases must be accounted for when evaluating the values of $\phi_{i,j}$. The following combinations are possible for all carrier types:

1. Wanted transmitter is non-hopping narrow band, interfering transmitter is non-hopping narrow band.
2. Wanted transmitter is non-hopping narrow band, interfering transmitter is non-hopping wide band.
3. Wanted transmitter is non-hopping narrow band, interfering transmitter is hopping.

4. Wanted transmitter is non-hopping wide band, interfering transmitter is non-hopping narrow band.
5. Wanted transmitter is non-hopping wide band, interfering transmitter is non-hopping wide band.
6. Wanted transmitter is non-hopping wide band, interfering transmitter is hopping.
7. Wanted transmitter is hopping, interfering transmitter is non-hopping narrow band.
8. Wanted transmitter is hopping, interfering transmitter is non-hopping wide band.
9. Wanted transmitter is hopping, interfering transmitter is hopping.

For case 1, equation (8.3) is used, where θ is derived from channel separations. In cases 2, 4 and 5, θ is derived from the appropriate spectral function (section 9.2 and section 9.3). For cases 3, 7 and 9, equation (10.2) is used, as the non-hopping carriers are narrow band. In cases 3 and 7 one of the carriers has a frequency list length of 1. Cases 6 and 8 are handled using:

$$\phi_{i,j} = \frac{\sum_{k=1}^{N} \theta_{i,k}}{N} \tag{10.5}$$

where N is the number of frequencies in the hopping list j, i is the wide band carrier and $\theta_{i,k}$ is derived from the appropriate spectral function (section 9.2 and section 9.3) for each frequency channel k in the hopping list j. Thus the sum of the rejection factors is averaged over the number N of frequency channels in the hopping list, i.e. $\phi_{i,j}$ is the average $\theta_{i,k}$ over all k in the list j. Equation (10.5) is a novel proposal designed specifically to handle the unique cases 6 and 8.

10.2.4 List Length Penalties

A list length penalty is a novel proposal designed to increase the frequency hopping list lengths for security purposes. A desired list length may be specified and then a cost is incurred if the frequency hopping list length is below this value, hence maximising the hopping list lengths. The addition of this capability adds an extra weighted component to the cost function, given in equation (10.6).

$$W_3 \times \sum_{j=1}^{N}(\max\{0, \text{desiredlistlength} - \text{currentlistlength}\})^b \tag{10.6}$$

W_3 is the weight applied to the cost function component, j is the transmitter identity, N is the total number of transmitters and b is the power applied to the deficit. If total network coverage is achieved the cost function will only be equal to zero if all desired frequency hopping list lengths have been satisfied.

10.2.5 Contiguous Block Penalties

A novel proposal has been designed to handle the requirement in military spread spectrum systems to limet the number of contiguous blocks of frequency channels in the hopping lists. Hence, a further weighted component is added to the cost function to minimise the number of contiguous blocks of frequency channels in the frequency channel hopping lists:

$$W_4 \times \sum_{j=1}^{N}(\max\{0, \text{currentcontiguous} - \text{desiredcontiguous}\})^c \tag{10.7}$$

W_4 is the weight applied to the cost function component, j is the transmitter identity, N is the total number of transmitters and c is the power applied to the deficit. Again, if total network coverage is achieved the cost function will only be equal to zero if all desired contiguous counts have been satisfied.

10.2.6 Weighted Linear Combination of Costs

The cost function is comprised of all the components outlined so far, this consists of the binary constraint violations, SIR cost, list length penalties and contiguous block penalties. Each cost function component is weighted by W_1, W_2, W_3 and W_4 respectively. The combined cost function is therefore as follows:

$$\text{C} = (W_1 \times Con_{vio}) + (W_2 \times \text{E}) + (W_3 \times \text{LL}) + (W_4 \times \text{CC}) \tag{10.8}$$

where Con_{vio} is the sum of the binary constraint violations, E is the sum of the SIR deficits (given in equation (8.5)), LL is the sum of the list length penalties (given in equation (10.6)) and CC is the sum of the contiguous block penalties (given in equation (10.7)).

10.3 Data Structures

To incorporate the dynamic construction of lists of frequency channels a new data structure has been designed. The structure is a 'doubly linked list'. In a 'doubly linked list', every list element has both a reference to the next element and a reference to the previous element. There is also a pointer to the 'head' of the list and a pointer to the 'tail' of the list. Figure 10.1 presents a diagrammatic description of the 'doubly linked list' data structure used for the storage of frequency lists.

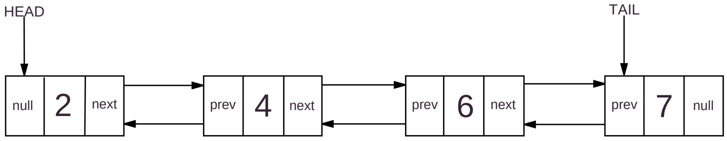

Figure 10.1: Doubly linked list data structure used to represent hopping lists.

The structure provides efficiency when updating and accessing, with increased efficiency for insertions and deletions. The efficiency is not quite as good as normal arrays for direct retrieval. Arrays do not have the capabilities that are required for frequency hopping lists. The capabilities include being able to quickly locate the next or previous non-empty element in a sparsely populated list, as illustrated in figure 10.1. Operations on the structure consist of:

- empty the list;
- add an element;
- remove an element;
- swap elements (add/remove);
- output list.

Figure 10.2 presents pseudocode for the adding of an element operation. The operation to remove an element is similar in nature to the operation to add an element. The swap elements operation comprises of a combination of the add and remove elements operation. The operation of emptying the list consists of removing all stored values. The output of the list requires a run through the list elements.

Operations that involve the comparison of two frequency lists consist of:

- get co-channel count;
- get adjacent channel count.

Figure 10.3 and figure 10.4 illustrate examples of the get co-channel count and the get adjacent channel count operations. These operations are used in the cost function, see section 10.2.

Four additions have also been made to the variables data structure (see section 7.5).

- Boolean value: used to distinguish whether the transmitter is a hopper, i.e. true for hopper and false for non-hopping transmitters;
- Integer value: used to store the desired hopping list length;
- Integer value: used to store the maximum number of contiguous blocks preferred in the hopping list;
- Doubly linked list: used to store the actual hopping list assigned to the transmitter.

Figure 10.5 gives an example of a partial variables data structure with the inclusion of frequency hopping carriers.

The additions made to the variables data structure allow for frequency assignment problems that can now include combinations of homogeneous carrier types, aggregated channel carriers, heterogeneous wide band carrier types, CDMA carriers and unsynchronised hopping carriers.

10.4 Carrier Type File Amendments

The carrier type file first outlined in section 7.4 and again in section 9.4 must be expanded further to include hopping carriers. The software must be aware of which carriers are hoppers as well as

```
T, transmitter chosen at random
F, new frequency channel chosen at random to be added
$LIST_T$, frequency channel list for transmitter T
$pLIST_T$, frequency channel list pointer
$pTEMP$, $pTEMP_{PREV}$, $pTEMP_{NEXT}$, temporary pointers
IF($pLIST_T$[F]→DATA $\neq$ NULL)
    F already in list
ELSE IF($LIST_T$→HEAD = NULL), list is empty
    $pLIST_T$[F]→DATA = F
    $LIST_T$→HEAD = $pLIST_T$[F]
    $LIST_T$→TAIL = $LIST_T$→HEAD
    $LIST_T$→ListLength = 1
ELSE IF(F $<$ $LIST_T$→HEAD), insert before HEAD of list
    $pLIST_T$[F]→DATA = F
    $pTEMP$ = $LIST_T$→HEAD
    $LIST_T$→HEAD = $pLIST_T$[F]
    $LIST_T$→HEAD→NEXT = $pTEMP$
    $pTEMP$→PREV = F
    $LIST_T$→ListLength = $LIST_T$→ListLength + 1
ELSE IF(F $>$ $LIST_T$→TAIL), insert after TAIL of list
    $pLIST_T$[F]→DATA = F
    $pTEMP$ = $LIST_T$→TAIL
    $LIST_T$→TAIL = $pLIST_T$[F]
    $LIST_T$→TAIL→PREV = $pTEMP$
    $pTEMP$→NEXT = F
    $LIST_T$→ListLength = $LIST_T$→ListLength + 1
ELSE, insert into list between HEAD and TAIL
    $pLIST_T$[F]→DATA = F
    WHILE($pTEMP$→DATA = NULL)
        $pTEMP$ = $pLIST_T$[F–]
    END WHILE
    $pTEMP_{PREV}$ = $pTEMP$
    $pTEMP_{PREV}$→NEXT = F
    WHILE($pTEMP$→DATA = NULL)
        $pTEMP$ = $pLIST_T$[F++]
    END WHILE
    $pTEMP_{NEXT}$ = $pTEMP$
    $pTEMP_{NEXT}$→PREV = F
    $pLIST_T$[F]→PREV = $pTEMP_{PREV}$→DATA
    $pLIST_T$[F]→NEXT = $pTEMP_{NEXT}$→DATA
    $LIST_T$→ListLength = $LIST_T$→ListLength + 1
END IF
```

Figure 10.2: Pseudocode for the insertion of a random frequency into a frequency hopping list.

which carriers are wide band or narrow band. Hopping carriers add three components to the carrier type file; a carrier code defining the type of carrier, desired list length and the maximum number of

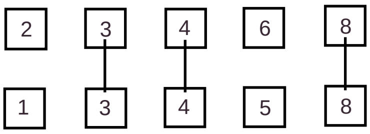

Figure 10.3: An example of two frequency hopping lists with co-channel clashes.

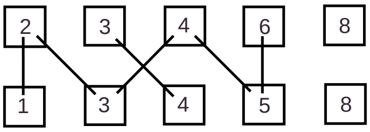

Figure 10.4: An example of two frequency hopping lists with adjacent channel clashes.

contiguous blocks. The carrier code is either a ‘n’ for non-hopping or ‘h’ for hopping. The list length and contiguous values are explained in this chapter. Figure 10.6 gives an example of the carrier type file extended for hopping carriers.

10.5 Test Data

Test data for unsynchronised hopping carriers involves SIR cost based problems with the possible inclusion of any number of carrier types. Results should show:

1. that the inclusion of frequency hopping carriers improves network performance when compared to the non-hopping case;
2. that the inclusion of frequency hopping carriers provides little or no loss in performance when compared to the non-hopping case.

Results for frequency hopping will be presented in chapter 14.

IsHopper	Listlength	Contiguous	Frequency list
T	10	4	1 2 5 7
F	1	0	3
T	15	5	3 5 6 7 8 10 11 23 27
T	5	2	2 3 6 7 8

Figure 10.5: Variables data structure additions, incorporating hopping carriers.

```
% transmitter and receiver bandwidth and power data
% format : transmitter bandwidth recbw power beta hopper listlength contiguous
     1     18    10    1    1    n    1    1
     12     1     1    1    1    h   10    5
     15     1     1    1    1    h   15   10
     17    50     7    1    1    n    1    1
     30    22     5    1    0    n    1    1
     32     1     1    1    1    h   12    7
     37    12     8    1    1    n    1    1
     123    8     3    1    0    n    1    1
     983    1     1    1    1    h    5    5
```

Figure 10.6: Extended Carrier Type file format consisting of wide band and hopping carriers.

Chapter 11

Synchronised Hopping Carriers

11.1 Overview

This chapter discusses the inclusion of synchronised frequency hopping carriers into the model. In synchronised hopping a group of carriers are synchronised. As in unsynchronised frequency hopping, a list of frequencies is assigned to a hopping carrier. Additionally, one or more hopping sequences are specified for the group of carriers. These specify the order in which the carriers hop over their hopping lists. Either there is one sequence per carrier or a single sequence is used with different offsets from the first element of the sequence for different carriers in the group. Hopping sequences (over a hopping list or a specified pattern of hopping lists) are assumed to be specified in advance and the algorithm will determine the correspondence between the pattern of hopping lists (or single hopping list) for the group of synchronised carriers and actual frequencies. It is assumed that the interference between synchronised carriers will be small (or negligible), while the interference with another unsynchronised carrier (or fixed frequency carrier or a carrier from a different synchronised group) will be the same as if the synchronised carrier were hopping pseudo-randomly, as specified in section 10.2.

Also discussed in this chapter are changes to the cost function, changes to the data structures and changes to the carrier type file.

11.2 Cost Function

When a frequency list is changed for a synchronised frequency hopping carrier all other carriers in the synchronised group also have their frequency list changed in the same way. Therefore the change in SIR cost must be evaluated for each synchronised frequency hopping carrier in the group. The method discussed in section 8.6.1 is adopted for groups of synchronised frequency hopping carriers.

Allowing for the fact that interference between synchronised carriers is negligible, the SIR cost for a synchronised frequency hopping carrier is otherwise calculated in the same way as for an unsynchronised frequency hopping carrier, discussed in section 10.2. Frequency list length costs and contiguous frequency block costs also apply to synchronised frequency hopping carriers.

In a similar way to the movement of an AGCC group (section 6.2) and the movement of wide band carriers (section 7.3), the change to a hopping list for a group of synchronised frequency hopping carriers produces high cost changes. A user defined weight X_3 is used to increase the probability of a synchronised frequency hopping carrier move being accepted, according to the formula prob=$e^{\frac{-\Delta E}{tX_3}}$. The probability of a synchronised frequency hopping carrier move being accepted will decrease with the temperature, allowing the frequency lists to converge.

11.3 Data Structures

Two additions have been made to the variables data structure (see section 10.3).

- Boolean value: used to distinguish whether the transmitter is a synchronised frequency hopper, i.e. true for synchronised frequency hopper and false for other transmitter types;
- Integer value: used to store the synchronised group number;

Figure 11.1 gives an example of a partial variables data structure with the inclusion of synchronised frequency hopping carriers.

When a frequency list for a group of synchronised frequency hopping carriers is amended only one list has the changes made, this carrier is considered to be the group leader and is always the transmitter with the lowest identity. The frequency lists of the other carriers in the group are given the same memory address as the frequency list of the group leader. Therefore whenever a change is made to the group leader's frequency list the changes are automatically made for the other carriers in the group.

The additions made to the variables data structure allow for frequency assignment problems that can now include combinations of homogeneous carrier types, aggregated channel carriers, heterogeneous wide band carrier types, CDMA carriers, unsynchronised frequency hopping carriers and synchronised

IsHopper	IsSynch	SynchGroup	Listlength	Contiguous	Frequency list
T	F		10	4	1 2 5 7
T	T	1	10	3	2 8 9 11 13 19 21
F	F		1	0	3
T	T	1	10	3	2 8 9 11 13 19 21
T	T	2	12	5	5 11 28 30 31 32
T	T	1	10	3	2 8 9 11 13 19 21
T	F		15	5	3 5 6 7 8 10 11 23 27
T	T	2	12	5	5 11 28 30 31 32
F	F		1	0	2

Figure 11.1: Variables data structure additions, incorporating synchronised frequency hopping carriers.

frequency hopping carriers.

11.4 Carrier Type File Amendments

The carrier type file first outlined in section 7.4, section 9.4 and section 10.4 must be expanded further to include synchronised frequency hopping carriers. The software must be aware of which carriers are synchronised frequency hoppers as well as which carriers are unsynchronised frequency hoppers, wide band and narrow band. Synchronised frequency hopping carriers add only one component to the carrier type file; the synchronised group number. The carrier code is defined as ‘s’ for a synchronised frequency hopper. Figure 11.2 gives an example of the carrier type file extended for synchronised frequency hopping carriers.

The format of the file is:

- Column 1: Transmitter identity;
- Column 2: Transmitter bandwidth, given as a number of narrow band channels;
- Column 3: Receiver bandwidth, given as a number of narrow band channels;
- Column 4: Transmitter power;
- Column 5: β value, spectral function parameter (see section 9.3);
- Column 6: Carrier type, this is a character that defines whether the carrier is an unsynchronised hopper, synchronised hopper or a non-hopper;

```
% transmitter and receiver bandwidth and power data
% format : transmitter bandwidth recbw power beta hopper listlength contiguous group
    1     18     10     1     1     n     1     1
    2     1      1      1     1     s     10    5     1
    4     1      1      1     1     s     12    7     2
    5     1      1      1     1     s     10    5     1
    6     1      1      1     1     s     12    7     2
    12     1      1      1     1     h     10    5
    15     1      1      1     1     h     15    10
    17     50     7      1     1     n     1     1
    30     22     5      1     0     n     1     1
    32     1      1      1     1     h     12    7
    37     12     8      1     1     n     1     1
    123     8     3      1     0     n     1     1
    983     1     1      1     1     h     5     5
```

Figure 11.2: Extended Carrier Type file format consisting of wide band, unsynchronised frequency hopping carriers and synchronised frequency hopping carriers.

- Column 7: List length, this is the desired length of a frequency list;
- Column 8: Contiguous blocks, this is the maximum number of permitted contiguous blocks of channels in a frequency list;
- Column 9: Group number, this is used to group together synchronised hopping carriers.

11.5 Test Data

Test data for synchronised hopping carriers will involve SIR cost based problems with the possible inclusion of any number of synchronised hopping carrier groups. Experiments carried out on these test data will be presented in chapter 14.

Chapter 12

A Generic Model of Military Frequency Assignment

The aims and objectives of this project have been stated in section 1.1. The model proposed in this chapter encompasses all the necessary considerations and components required to develop a generic system that can handle all types of military frequency assignment problems (ground, satellite, maritime and air). These considerations have been described in detail in previous chapters. The model is successful as a common formulation and it has been demonstrated that almost all parts of the model can be efficiently implemented.

12.1 A Detailed Description of the Model

This section outlines in detail the proposed model of the common formulation for the four areas of military frequency assignment problems. The model is presented in figures 12.1, 12.2 and 12.3. A detailed explanation of the figures is given in this section. The model is broken down into three levels with figure 12.1 representing the top level, giving the six major areas. Level two is presented in figure 12.2 and gives more details of the six areas, listing their attributes. Level three is presented in figure 12.3 and gives more detail of the components contained in the first of the major areas. The arrows on the diagrams represent the way the specific areas of the model interact to eventually produce a solution. The following subsections provide a more detailed description of the model breakdown.

12.1.1 Potential Interference

"Potential interference" addresses the forms of interference that can affect the quality of a received signal. It is broken down into the information, models and constraints under the headings of transmitters, propagation/SIR, intermodulation contribution to SIR (satellite), received signal strengths and constraints. These areas are interlinked and collectively allow the potential interference to be considered; when taken in combination with the carrier type area.

Consideration of potential interference leads to the formulation of binary constraints or received signal strength information, together possibly with intermodulation, spurious emission and spurious response information.

Transmitters

The necessary transmitter information consists of transmitter locations and transmitter powers. The transmitter locations are generally given as coordinates relative to the network. This information may be provided to compute the necessary assessment of received signal strengths and in order to formulate binary constraints.

Propagation / SIR

Propagation models are used to calculate the received signal strengths for SIR problems; these may also be used to generate a set of binary constraints. Propagation models must take into account the transmitter information provided. The propagation model can be any propagation model that provides accuracy when generating received signal strengths. The following models were used for experimentation purposes in this research:

- d^{-4} model (section 8.3);
- $|\frac{sinx}{x}|^2$ model (section 8.3.2);
- terrain based model (section 14.2.2).

The most accurate models for terrestrial systems requires knowledge of the terrain.

Intermodulation contribution to SIR (Satellite)

The inclusion of intermodulation product effects in SIR calculations may be necessary for satellite problems. For these problems the avoidance of intermodulation products altogether is not feasible, and so their effects must be minimised. This issue is complex and has not been studied in detail in this work, but needs to be included in the general model. For this reason the dashed line has been included in figure 12.2. For further information see [47].

Received Signal Strength

The necessary information on received signal strengths is a list of reception points with the received signal strength provided for each serving and potentially interfering transmitter. These may be provided or are computed using a propagation model.

Constraints

Constraints may consist of binary far-site constraints (section 4.4.1), binary co-site constraints (section 4.4.1), binary hopping list constraints, binary equality constraints (section 6.3), intermodulation product constraints [41] and spurious co-site constraints [41].

Binary constraints may be provided or may be generated by the propagation/SIR model for a specific SIR threshold (section 8.4). The other constraints may be provided.

The binary constraints may themselves define the problem representation and may optionally be used in SIR problems.

12.1.2 Problem Representation

Information for the problem representation is derived from the potential interference and the carrier types. The problem representation may require either binary constraints or received signal strengths, or a combination of both, depending on the type of frequency assignment problem. This information is then used by an algorithm.

The problem representation consists of input files that include a binary constraint file (section 4.4.1 and 6.3), variable file (section 4.4.2), carrier type file (section 7.4, 9.4, 10.4 and 11.4), frequency domain file (section 4.4.3), start file (section 4.4.4), intermediate file (section 8.8.3), intermodulation product file [41], spurious file [41] and a satellite intermodulation product file [47].

12.1.3 Carrier Types

The carrier type information defines narrow band carriers (chapter 4), wide band carriers (chapter 7), CDMA carriers (chapter 7), aggregated channel carriers (chapter 6), unsynchronised hopping carriers (chapter 10), synchronised hopping carriers (chapter 11) and the effects of frequency separations.

The types of carriers involved in a problem are known before the assignment process begins. The problem representation is used to tell the algorithm which types of carriers are involved in the problem. It is important that the cost function knows the types of carriers involved in a problem especially when SIR calculations and hopping list calculations are being used.

12.1.4 Cost Function

The cost function is used by the algorithm to evaluate the cost of an assignment. The cost function incorporates the number of binary constraint violations (section 4.3.1), a cost derived from SIR calculations (section 8.6), a cost derived from satellite intermodulation SIR, hopping list length penalties (section 10.2.4) and hopping contiguous penalties (section 10.2.5). The overall cost is a weighted linear combination of the cost functions (section 8.7) that may depend on the type of carriers involved in the problem as well as the algorithm used.

12.1.5 Algorithms

The algorithms consist of a set of meta-heuristic algorithms or other frequency assignment algorithms. Those implemented for this research include hill climbing (section 2.2), simulated annealing (section 2.3), SIR hill climbing (section 8.7.1), pure SIR simulated annealing (section 8.7.1), SIR simulated annealing hybrid (section 8.7.2) and SIR simulated annealing combined (section 8.7.3). Algorithms that have not been implemented for this research include threshold accepting (section 2.3.1) and tabu search (section 2.4), sequential algorithms and many others, hence the inclusion of the dashed line figure 12.2.

The algorithms use the problem representation and the cost function to assign frequencies to carriers and then evaluate how good the assignment is. The objective of the algorithm is to produce a solution that contains an assignment with as little interference as possible, as measured by the cost function.

12.1.6 Solution

The solution produced by the algorithm is a frequency assignment that can be output to a file, the frequency assignment has associated costs depending on the carrier types and cost functions used. Two types of output files have been defined one for non-hopping frequency assignment (section 4.4.4) and another for hopping frequency assignment. The solution is the final stage of the frequency assignment improvement process, although an assignment can be used as a start assignment in an effort to find an improved assignment.

12.2 Model Summary

This research has indicated that the model includes all common considerations necessary in military frequency assignment (chapter 3). The software implemented here, section 13.3, together with the work in [41] shows that efficient implementation is possible, except possibly, for the satellite intermodulation effects, which need a separate study.

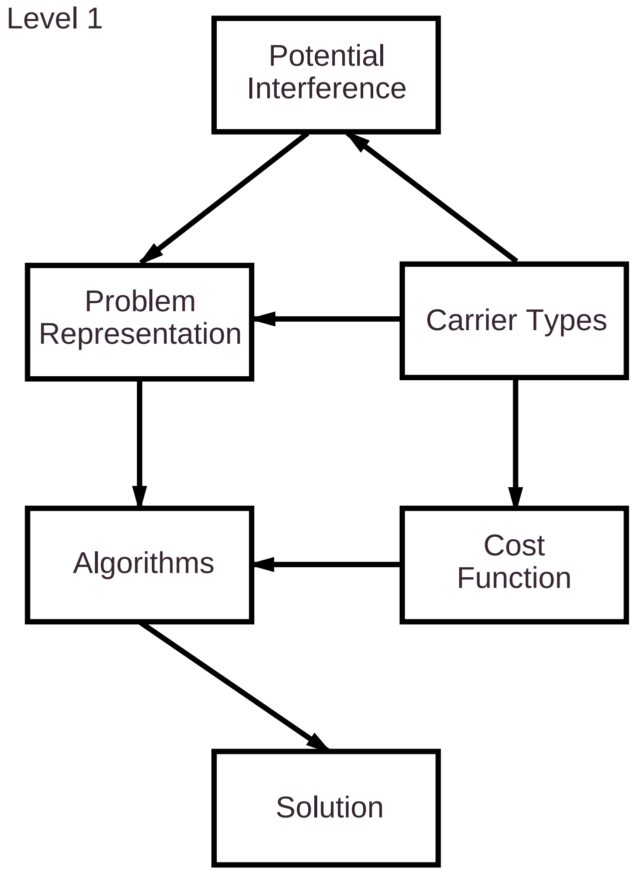

Figure 12.1: Level 1 introduces the main components of the model and how they are linked together.

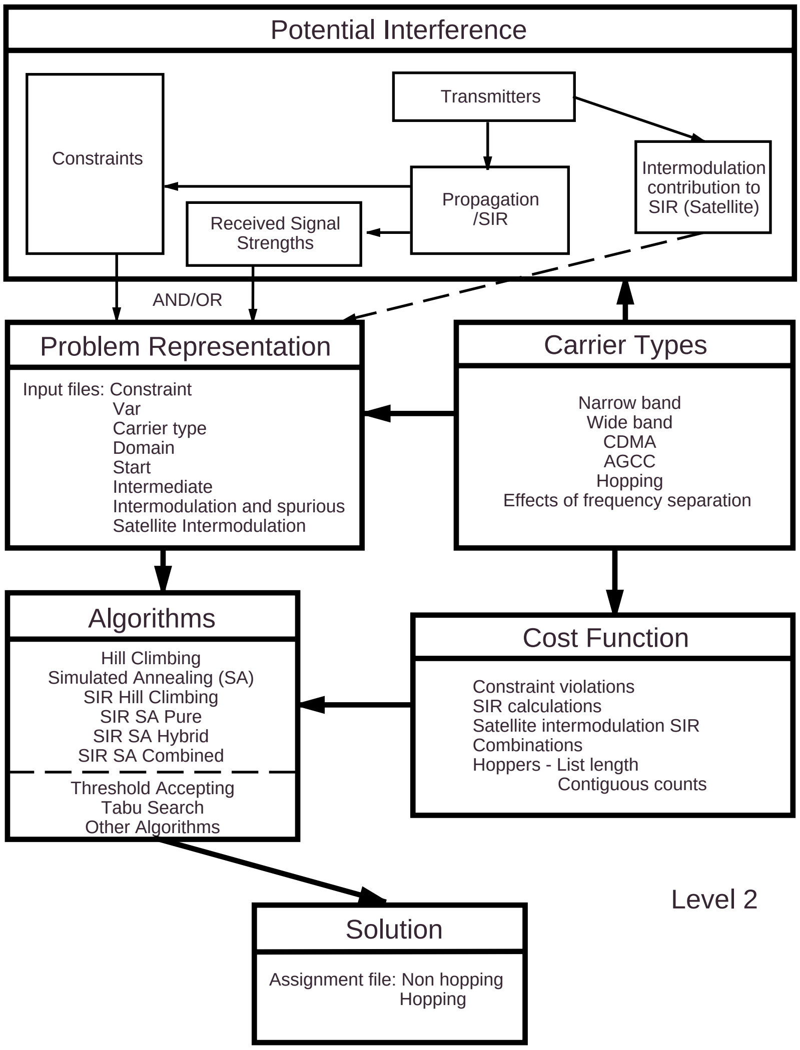

Figure 12.2: Level 2 gives a more detailed view of the components of the model by listing their attributes.

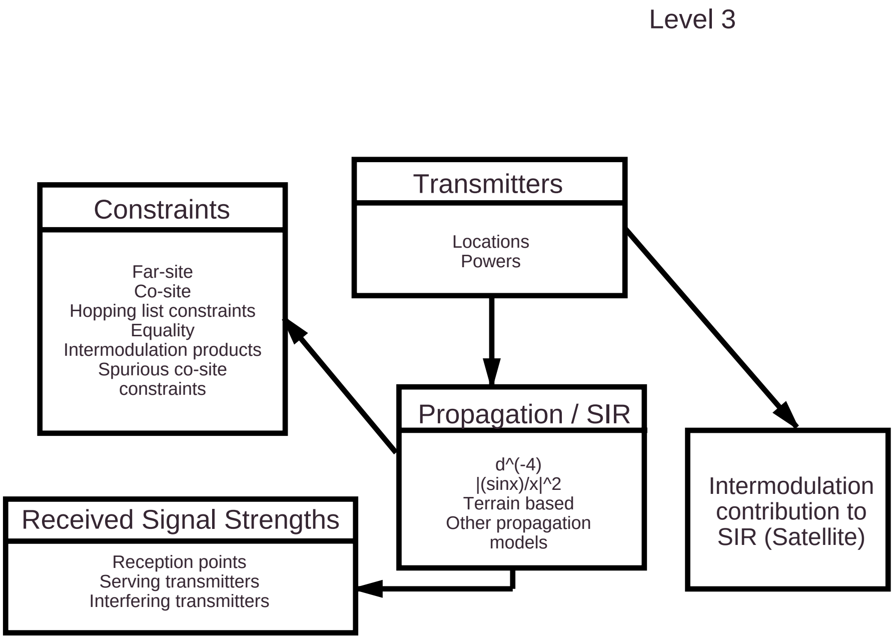

Figure 12.3: Level 3 gives a detailed view of the potential interference component only.

Chapter 13

Software

13.1 Overview

This chapter will outline the software developed specifically for this project. The software has been developed using Microsoft Visual C++ in a Windows environment. The software has been named "FAPSolver" and is used as a platform on which to perform many varied experiments encompassing all aspects of work carried out for this project. FAPSolver has been compared with FASoft [26] in the area of simple binary constraint based problems and ANTS [21] in the area of SIR calculations involving narrow band carriers. FAPSolver uses the hill climbing algorithm (section 2.2) and the simulated annealing algorithm (section 2.3). The objective of this research is to define a common formulation of military frequency assignment problems and then apply meta-heuristic algorithms to them with little or no loss in performance. The use of the simulated annealing algorithm is enough to show that this can be achieved (see section 14.7).

13.2 User Interface

The primary user interface for FAPSolver consists of a dialogue-like collection of controls, consisting of a dialogue box as the main window. The dialogue box is split into six sections that consist of method, status, input files, output files, options and output. These sections are made up of various buttons and displays providing the user with detailed information on the current problem. Once the user has started the algorithm a small output window appears giving the user up-to-date information on the progress of the current solution.

Each algorithm has its own options dialogue box. These dialogue boxes give the user the opportunity to fine-tune the way any of the algorithms work by allowing the adjustment of all of the algorithm's parameters.

Therefore, a single coherent GUI (Graphical User Interface) gives the user complete control of the way in which an algorithm will attempt to produce a solution to a given problem. A screenshot of the GUI is presented in figure 13.1.

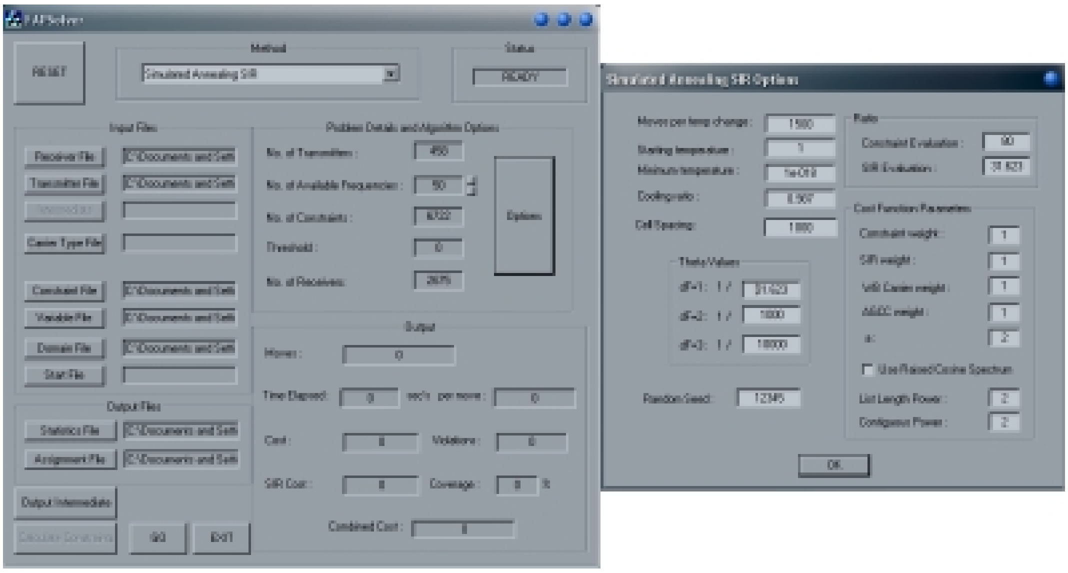

Figure 13.1: Graphical User Interface for FAPSolver.

13.3 Software Architecture

The software architecture is best described using the software engineering technique "the Unified Modelling Language" (UML) [36] and the case tool "Rational Rose". UML allows the software to be broken down into classes, showing the ways in which they are associated.

The software is made up of *entity* classes and *control* classes, with a front end *boundary* class. The front end boundary class named FAPSolverUI is the Graphical User Interface, discussed in section 13.2, that is used by the user to initiate a sequence of events that cause the control classes to operate. The entity classes reflect the data stored in the system and consist of Transmitter, Constraint, Domain, Frequency, SIRTransmitter and SIRReceiver. Each entity contains information that can be accessed by the control classes. The control classes reflect the algorithms, and they consist of HillClimbAlgorithm, SimulatedAnnealingAlgorithm, SIRHillClimbAlgorithm and SIRSimulatedAnnealingAlgorithm.

In UML the entity classes represent the static structure of the software design, figure 13.2 presents a UML diagram showing the entity classes and how they are associated. The Constraint entity contains the binary constraints for pairs of transmitters. The Transmitter entity contains the information that is relevant to the specific type of carrier as described in section 6.4, section 7.5 and section 10.3, e.g. if the transmitter is a wide band carrier the bandwidth of the carrier must be stored. The Transmitter entity also stores the identity of the transmitter's associated frequency domain and the frequency assigned to the transmitter. A transmitter can be associated with only one frequency domain and is assigned to a single frequency unless the transmitter is a hopper; if this is the case the transmitter can be assigned to one or many frequencies. A frequency may be assigned to zero or many transmitters. A frequency domain can contain one to many frequencies and can be associated with one to many transmitters. The SIRTransmitter entity "inherits" all of the attributes of the Transmitter entity, but also contains further information specific to SIR problems as described in section 8.5, e.g. the signal strength of the transmitter. The SIRTransmitter entity is essentially used to store information for the interfering transmitters in SIR problems. A transmitter in an SIR problem may be the wanted transmitter for one to many receivers in a network. The SIRReceiver entity contains information specific to SIR receiver points as described in section 8.5, e.g. the receiver signal strength and the bandwidth of the receiver filter used in wide band problems. Specific receiver point information can only apply to one transmitter.

Figure 13.3 presents a UML class diagram of the entire software architecture. As stated earlier the boundary FAPSolverUI class provides the entry-point of the information for the entity classes and the parameters for the control classes. The user can run one algorithm at a time. The HillClimbAlgorithm control class refers to the binary constraint solving hill climb algorithm and contains the binary constraint cost function. The hill climb algorithm interacts with the Transmitter entity and the Domain entity, i.e. it assigns frequencies contained in frequency domains to the transmitters. The SimulatedAnnealingAlgorithm control class refers to the binary constraint solving simulated annealing algorithm and inherits the attributes of the HillClimbAlgorithm control class and adds its own attributes. The SIRHillClimbAlgorithm refers to the SIR hill climb algorithm that inherits all of the attributes of the HillClimbAlgorithm control class. This allows the SIR hill climb algorithm

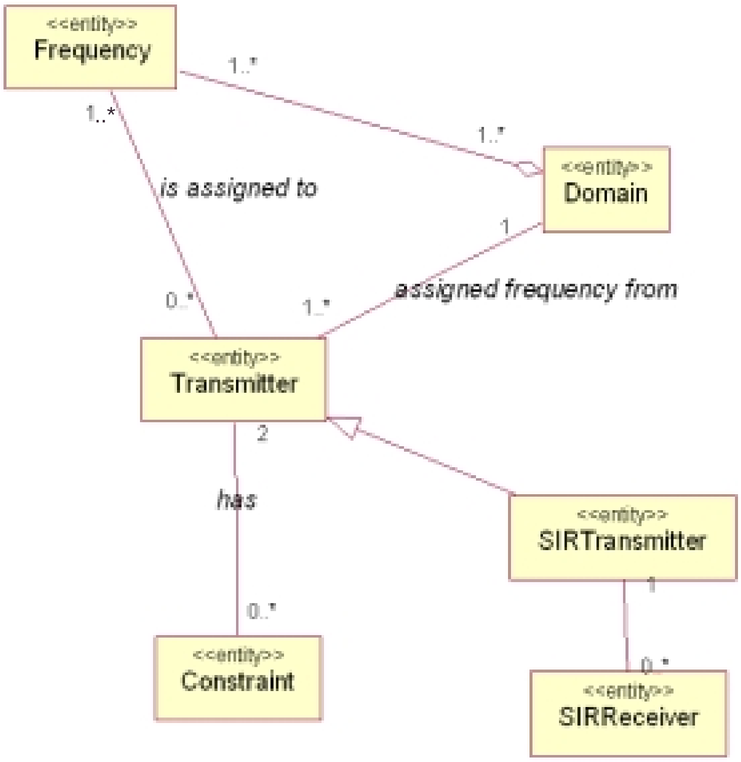

Figure 13.2: UML class diagram of the entity classes.

to combine the binary constraint solving attributes with its own SIR attributes. The SIR hill climb algorithm interacts with the SIRTransmitter entity and the SIRReceiver entity to solve SIR problems. The SIRSimulatedAnnealingAlgorithm control class inherits all of the attributes from the SIRHillClimbAlgorithm control class. This inheritance allows for the hybrid simulated annealing algorithm that solves binary constraint problems, SIR problems and combinations of the two types of problems.

13.4 Data Structures

This section is used to show how the data structures presented in this thesis interact. Figure 13.4 presents a diagram that shows how the data structures relate to one another.

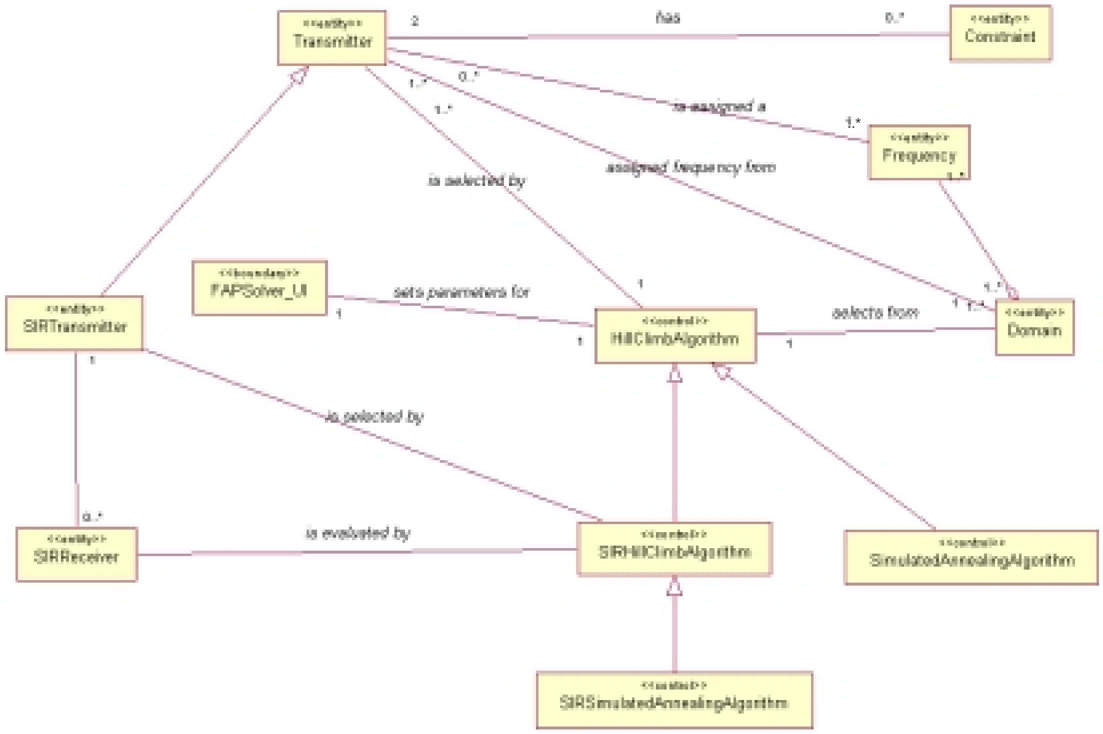

Figure 13.3: UML class diagram of the entire software architecture.

The algorithm accesses the variables data structure (section 4.2.1) at runtime to determine the type of carrier that is being assigned a frequency. If the carrier is an AGCC (chapter 6) then the AGCC data structure (section 6.4) is accessed, similarly if the carrier is a synchronised hopping carrier then the synchronised hopping group data structure is accessed. The synchronised hopping group data structure is of similar format to the AGCC data structure, with the exception of carrier separations, positive channels and negative channels (figure 6.4). If a frequency domain has been specified then the frequency domain structure (section 4.2.3) is accessed. The frequency assigned to the transmitter is eventually stored in the assignment data structure (section 4.2.4).

The binary constraint data structure (section 4.2.2) is filled with data by either a binary constraint file (section 4.4.1) or by binary constraint generation (section 8.4) using the information held in the interfering transmitter data structure (section 8.5) and the serving transmitter data structure (sec-

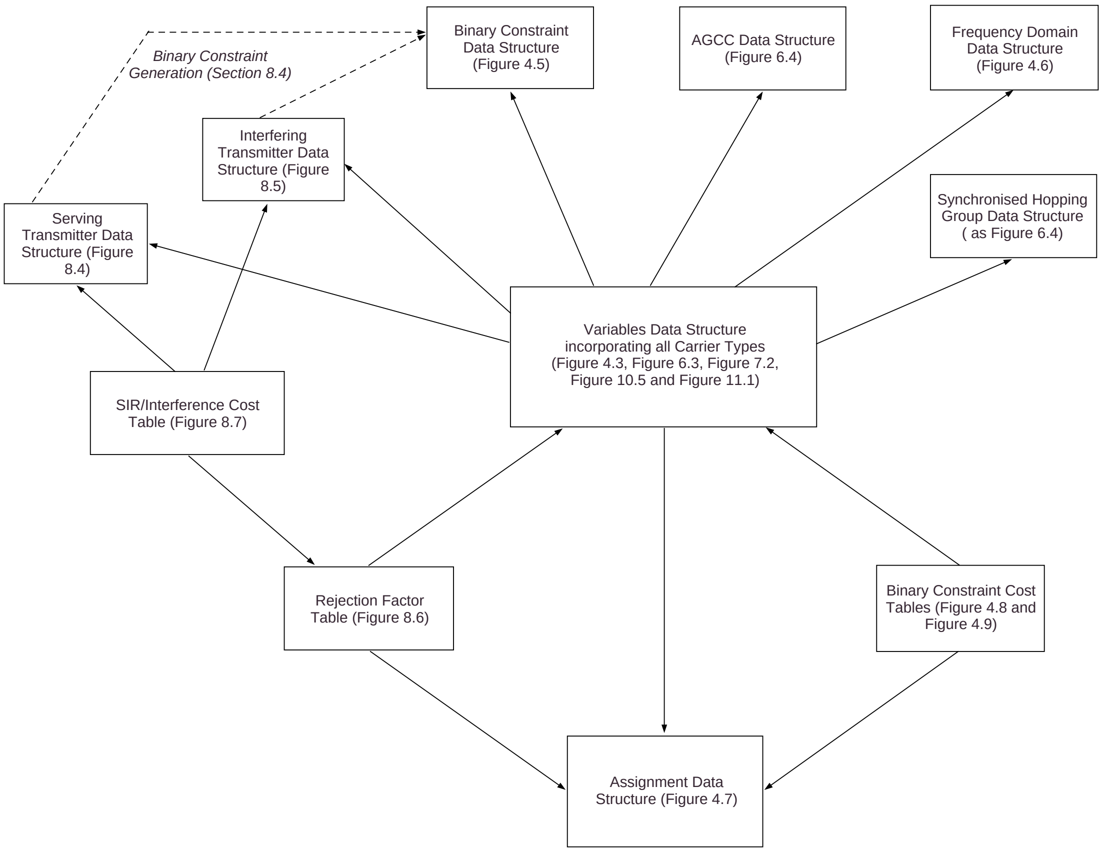

Figure 13.4: Data structure interaction diagram.

tion 8.5). The information stored in the interfering data structure and the serving transmitter data structure is derived by either a propagation model (section 8.3) or the intermediate file (section 8.8.3).

The cost function (section 10.2.6) can use a combination of binary constraint violations (section 4.3) and SIR cost (section 8.6). The binary constraint violations are stored in the binary constraint cost tables (section 4.3.1), and are derived from the frequencies assigned to pairs of transmitters (stored in the assignment data structure) and the channel separation constraint between the pair of transmitters (stored in the binary constraint data structure). The SIR cost is derived from the serving transmitter signal strength, stored in the serving transmitter data structure and the interfering transmitter signal strengths, stored in the interfering transmitter data structure. The interfering signal strengths are multiplied by a rejection factor (section 10.2.3). The rejection factors data structure (section 8.5) is filled by examining the current assignment of pairs of transmitters, these rejection factor values depend also on the type of carriers involved (section 10.2.3). The SIR cost and the interference at a receiver point are both stored in the SIR/Interference cost table (section 8.5).

In summary, the software presented here is an innovative system that can handle multiple carrier types for military frequency assignment problems. The software is capable of applying algorithms to solve binary constraint problems, SIR problems and combinations of the two. Section 14 presents the experiments performed using the software described here; the results of these experiments will determine if the software fulfils the aim of this research, which is to define a common formulation of military frequency assignment problems and then apply meta-heuristic algorithms to them with little or no loss in performance.

Chapter 14

Experiments and Results

14.1 Binary Constraint Experiments

Experiments have been performed to illustrate the performance of FAPSolver. Three sets of binary constraints have been generated using an SIR of 18dB (as described in section 8.4) from a 95 transmitter problem, a 458 transmitter problem and a 1794 transmitter problem. The simulated annealing binary constraint solving algorithm is used on the three sets of binary constraints. For the 95 transmitter problem a domain of 15 consecutive frequencies is available. For the 458 transmitter problem a domain of 20 consecutive frequencies is available. For the 1794 transmitter problem a domain of up to 257 frequencies is available, but some frequencies are blocked for certain transmitters. For each experiment the same parameters, derived from trial and error, are used for consistency and are as follows:

- Start temperature: 1.0
- Stop temperature: 0.001
- Iterations per temperature: 1500
- Cooling Ratio: 0.987

The results were then compared with those that can be obtained with the existing software package FASoft [26], they are given in table 14.1.

The results in table 14.1 indicate that FAPSolver has broadly similar performance to FASoft as binary constraint solving software. It is extremely important that FAPSolver can solve binary constraints quickly and efficiently as it is a major component of the generic model. The results given in table 14.1

Problem	Software	Constraint Violations	Runtime (seconds)
95 Transmitter	FAPSolver	46	7
95 Transmitter	FASoft	44	20
[1]458 Transmitter	FAPSolver	8	37
458 Transmitter	FASoft	5	50
1794 Transmitter	FAPSolver	0	43
1794 Transmitter	FASoft	0	34

Table 14.1: Results for binary constraint problems, comparing FAPSolver and FASoft runtimes. Experiments were performed on a Pentium 4 2.4GHz processor.

confirm that FAPSolver achieves this.

Further experiments have been performed to analyse the performance of the simulated annealing binary constraint solving algorithm when including different carrier types. Three problems are considered here; the 95 transmitter problem, the 458 transmitter problem and the 1794 transmitter problem. Four experiments were run for each problem with increasing numbers of AGCC groups and wide band carriers. The AGCC groups (with carriers in listed ascending order) are as follows:

1	2	3	4	5	6	7	8
1 2 = 1	5 6 = 2	8 9 = 3	13 14 = 1	17 18 = 2	19 20 = 2	23 24 = 3	26 27 = 1
2 3 = 2	6 7 = 2	9 10 = 4	14 15 = 1		20 21 = 2	24 25 = 1	27 28 = 1
3 4 = 1		10 11 = 1	15 16 = 1		21 22 = 2		28 29 = 1
		11 12 = 1					29 30 = 2

Figure 14.1: Aggregated Channel Carrier (AGCC) groups.

The format of the AGCC groups is i j = c, where i and j are transmitters and c is the fixed channel separation. The wide band carriers are given in figure 14.2.

The first experiment does not include any of the AGCC groups or wide band carriers. The second experiment includes AGCC groups 1 and 2, and wide band carriers 1 and 2. The third experiment includes AGCC groups 1 to 4, and wide band carriers 1 to 4. The final experiment includes all of the AGCC groups and all of the wide band carriers. The experiments use the algorithm parameters that were presented earlier in this section. The results are presented in table 14.2.

---

[1] The result for this problem was obtained using 2000 iterations per temperature and a start temperature of 2.

Carrier No.	Transmitter Identity	Bandwidth (channels)
1	31	5
2	32	7
3	33	9
4	34	5
5	35	11
6	36	15
7	37	17
8	38	5

Figure 14.2: Wide band carriers.

Problem	AGCC Groups	Wide Band Carriers	Constraint Violations	Runtime (seconds)
95 Transmitter	-	-	15	6
95 Transmitter	2	2	14	8
95 Transmitter	4	4	17	9
95 Transmitter	8	8	23	12
458 Transmitter	-	-	8	36
458 Transmitter	2	2	11	37
458 Transmitter	4	4	12	38
458 Transmitter	8	8	13	40
1794 Transmitter	-	-	452	216
1794 Transmitter	2	2	457	219
1794 Transmitter	4	4	460	217
1794 Transmitter	8	8	465	219

Table 14.2: Results for binary constraint problems, comparing runtimes when adding AGCC groups and wide band carriers. Experiments were performed on a Pentium 4 2.4GHz processor.

The results presented in table 14.2 show that there is very little loss in performance when including the AGCC groups and wide band carriers.

14.2 SIR Experiments

The main objectives of the SIR experiments are:

- to show which algorithm produces the best results: the simulated annealing SIR algorithm from a random start, the simulated annealing hybrid algorithm, the simulated annealing constraint solving algorithm or the simulated annealing SIR algorithm from a constraint start;
- to evaluate the ideal SIR threshold for generating binary constraints. Initial assumptions suggest

that the ideal SIR threshold for generating the binary constraints is roughly 2 or 3 decibels above the evaluation SIR threshold;

- to determine the importance of weighting the sum of binary constraints violations in a hybrid algorithm when attempting to obtain good results.

The hill climbing algorithm is not used for the experiments in this section as it is essentially less effective than the simulated annealing algorithm (see chapter 2).

The results of all of the SIR experiments presented in this chapter use the same cost function (equation 8.5) for SIR costs.

14.2.1 The 95 Transmitter and the 458 Transmitter Problem

The networks of 95 transmitters and 458 transmitters was created by a problem generator written by Allen and Dunkin [14]. The problem generator places one or more conurbations over a defined region. For each conurbation the factors that influence the population (and therefore transmitter density) are specified and a probability map produced. The problem generator proceeds to place transmitters over the region, with a probability that a transmitter will be at any point given by the above probability map. Any number of transmitter placements are possible by varying a parameter. The problem generator was also used to locate the reception points within the region. These are calculated using so called *Voronoi* polygons surrounding each transmitter. These are polygons such that every point within the polygon is closer to that transmitter than any other. It is assumed also, for simplicity, that all points within a polygon are tuned to the transmitter that the polygon surrounds. The points and intersections of these polygons are tuned to one, two or more transmitters and provide appropriate worst case reception points. The advantage of these problems is that the transmitter coordinates are given. This is not the case for many published benchmarks.

The 95 transmitter problem and the 458 transmitter problem have been presented as minimum span problems in [33] and [46] and so the runtimes are only indirectly comparable with the results presented in table 14.13 and table 14.24. The 95 transmitter problem and 458 transmitter problem have been presented as fixed spectrum problems in this work and (in the case of the 458 transmitter problem) recently in [21]. SIR experiments were performed in [25] but used GSM data.

The experiments use both 9dB and 15dB SIR evaluation thresholds over some domains of consecutive frequencies. The algorithms used are:

- simulated annealing constraint solving algorithm;
- simulated annealing (pure) SIR algorithm from a random starting assignment;
- simulated annealing weighted hybrid algorithm;
- simulated annealing SIR algorithm from a binary constraint starting assignment.

For the 9dB SIR evaluation threshold experiments the binary constraints are generated at SIR thresholds of 10dB, 11dB, 12dB, 13dB, 14dB and 15dB, while for the 15dB SIR evaluation threshold experiments the binary constraints are generated at SIR thresholds of 16dB, 17dB, 18dB, 19dB, 20dB, 21dB and 22dB. The simulated annealing hybrid algorithm uses various weights to give priority to the solution of the binary constraints, the weights used are 1, 10 and 100. It might be that with greater emphasis on the solution of the binary constraints better results will be obtained. The pure simulated annealing SIR algorithm is the only algorithm that does not include the solution of any binary constraints.

Further experiments on the 458 transmitter problem were performed for the work described in [21]. For these experiments the SIR evaluation threshold is 14dB. The frequency domain consists of 3 domains. Domain 1 has 18 frequencies and is $\{0, 1, 2 \dots 17\}$. Domain 2 also has 18 frequencies, and is $\{5, 6, 7, \dots 22\}$. Domain 3 also has 18 frequencies, and is $\{10, 11, 12, \dots 27\}$. Then transmitters $1, 2, 3 \dots, 100$ and $301, 302 \dots, 400$ must be assigned a frequency from domain 1, transmitters $101, 102, 103 \dots, 200$ and $401, 402 \dots, 458$ must be assigned a frequency from domain 2 and transmitters $201, 202, 203 \dots, 300$ must be assigned a frequency from domain 3. The experiments use $\theta_3 = \frac{1}{7541.257167}$ rather than the value specified in equation (8.3). These experiments, as well as those that will be described in section 14.2.3 and section 14.2.4 were all run on the same computer (a XEON 2.67GHz processor) to allow exact comparison of runtimes for the different methods.

14.2.2 Radiocommunication Agency Terrain Database Computed Data

Experiments were carried out using some data previously provided by the Radiocommunication Agency (now part of Ofcom). The data used the coordinates of the 458 transmitter problem. These data supplied transmission losses expressed in decibels for each transmitter and each of the reception points. The propagation loss algorithm used a terrain database and is based on ITU recommendation 470. There appears to be a problem with the data as the algorithm used defaults to free space loss at short distances. This causes the required number of frequencies to be considerably larger than when using the d^{-4} propagation model. The data is not necessarily more realistic than using path loss of d^{-4} on the original 458 transmitter problem, however it does allow for further experiments to be carried out on a different type of data.

The experiments consist of a 15dB SIR evaluation over a domain of 150 consecutive frequencies. The algorithms used are:

- simulated annealing constraint solving algorithm;
- simulated annealing (pure) SIR algorithm from a random starting assignment;
- simulated annealing weighted hybrid algorithm;
- simulated annealing SIR algorithm from a binary constraint starting assignment.

The binary constraints are generated at SIR thresholds of 16dB, 17dB, 18dB, 19dB, 20dB, 21dB and 22dB for this 15dB SIR evaluation. The simulated annealing hybrid algorithm uses various weights to give priority to the solution of the binary constraints, the weights used are 1, 10 and 100. The aim of these experiments to further evaluate the effectiveness of the different algorithms and their various parameters.

14.2.3 A Satellite Problem HEX358

The problem HEX358 represents a region served by 61 slightly overlapping circular satellite beams. These are represented by regular hexagons as shown in figure 14.3.

The co-ordinates of the cell centres are given by $x = 1000 \times (j - 1) + 500 \times i$, $y = 500\sqrt{3} \times i$, with $i, j \in \{1, 2, \ldots, 9\}$, $5 < i + j < 15$. They are generated sequentially in the order given by the sequence $(1, 5), (1, 6), \ldots, (1, 9), (2, 4), \ldots, (9, 5)$ for (i, j). The set of receiver test points R corresponds to the vertices of the regular hexagons. Each vertex may correspond to one, two or three receiver test points with different serving transmitters at the centre of an adjacent cell. Each beam has a *demand* value representing the number of carriers (transmitters) in the beam. For HEX358 these are given by the vector:

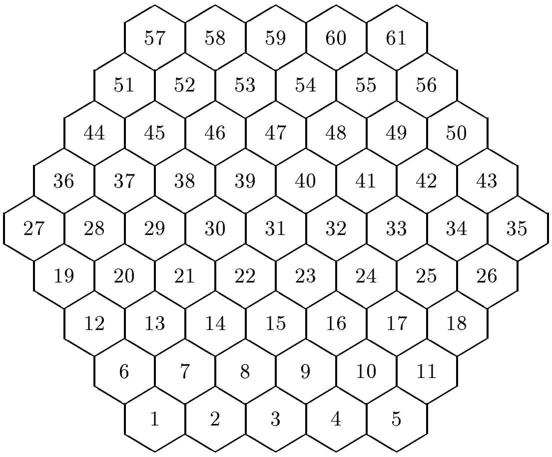

Figure 14.3: The cellular geometry of HEX358, with 61 satellite beams

$$\begin{aligned}&(6,8,7,7,5,4,8,8,7,5,5,5,6,4,4,4,6,8,7,7,5,4,8,8,7,5,5,5,6,4,4,\\&\quad 3,3,5,5,7,6,8,7,7,5,4,8,8,7,5,5,5,6,4,4,4,6,8,7,7,5,4,8,8,7\quad)\end{aligned}$$

where element i of the vector ($i \in \{1,61\}$) is the demand for cell i. The received signal strengths P_i at the cell centres are considered equal and the received signal strength at distance d from the cell centre is given by $P_i \times \left(\frac{\sin cd}{cd}\right)^2$, where $c = \frac{d\sqrt{3}\times 1.391557378}{1000}$ (as outlined in section 8.3.2) so that the value at a receiver test point is half the value at the corresponding cell centre. The problem is evaluated at a required SIR of 15dB. The frequency domain consists of 3 domains. Domain 0 is $\{0,1,2,\ldots 116\}$ with 117 frequencies. Domain 1 has 72 frequencies and is $\{45,46,47,\ldots 116\}$. Domain 2 also has 72 frequencies, and is $\{0,1,2,\ldots 71\}$. Number the carriers from 1 to 358, with the 6 carriers of cell 1 first, then the 8 carriers of cell 2 etc.. Then carrier i must be assigned a frequency from domain i mod 3. These experiments and those described in section 14.2.4 use different values for θ from those presented in equation (8.3), they are as follows:

$$\begin{aligned}\theta_1 &= 1/25.11886\\ \theta_2 &= 1/630.9573448\\ \theta_3 &= 1/4158.60336847\end{aligned}$$

14.2.4 A Satellite Problem HEX1794

In the problem HEX1794 the region served is covered by 310 slightly overlapping circular satellite beams, in a similar way to HEX358.

The co-ordinates of the 310 cell centres are given by $x = 1000 \times (j-1) + 500 \times i$, $y = 500\sqrt{3} \times i$, with $i, j \in \{1, 2, \ldots, 20\}$, $10 < i + j < 32$. They are generated sequentially in the order given by the sequence $(1,10), (1,11), \ldots, (1,20), (2,9), \ldots, (20,11)$ for (i,j). Again the set of receiver test points R corresponds to the vertices of the regular hexagons. Each vertex may correspond to one, two or three receiver test points with different serving transmitters at the centre of an adjacent cell. The *demand* values representing the number of carriers (transmitters) in each beam are given by the vector:

$$
\begin{array}{l}
(6,8,7,7,5,4,8,8,7,5,5,5,6,4,4,4,6,8,7,7,5,4,8,8,7,5,5,5,6,4,4, \\
3,3,5,5,7,6,8,7,7,5,4,8,8,7,5,5,5,6,4,4,4,6,8,7,7,5,4,8,8,7,5, \\
5,5,6,4,4,3,3,5,5,8,7,7,5,4,8,8,7,5,7,6,8,7,7,5,4,8,8,7,5,5,5, \\
6,4,4,4,6,8,7,7,5,4,8,8,7,5,5,5,6,4,4,3,3,5,5,7,6,8,7,7,5,4,8, \\
8,7,5,5,5,6,4,4,4,6,8,7,7,5,4,8,8,7,5,5,5,6,4,4,3,3,5,5,8,7,7, \\
5,4,8,8,7,5,7,6,8,7,7,5,4,8,8,7,5,5,5,6,4,4,4,6,8,7,7,5,4,8,8, \\
7,5,5,5,6,4,4,3,3,5,5,7,6,8,7,7,5,4,8,8,7,5,5,5,6,4,4,4,6,8,7, \\
7,5,4,8,8,7,5,5,5,6,4,4,3,3,5,5,8,7,7,5,4,8,8,7,5,7,6,8,7,7,5, \\
4,8,8,7,5,5,5,6,4,4,4,6,8,7,7,5,4,8,8,7,5,5,5,6,4,4,3,3,5,5,7, \\
6,8,7,7,5,4,8,8,7,5,5,5,6,4,4,4,6,8,7,7,5,4,8,8,7,5,5,5,6,4,4 \quad).
\end{array}
$$

The received signal strengths P_i at the cell centres are considered equal and the received signal strength at distance d from the cell centre is given by $P_i \times \left(\frac{\sin cd}{cd}\right)^2$, where $c = \frac{d\sqrt{3} \times 1.391557378}{1000}$ so that the value at a receiver test point is half the value at the corresponding cell centre.

The frequency domain consists of 3 domains. Domain 0 is $\{0, 1, 2, \ldots 256\}$ with 257 frequencies. Domain 1 has 212 frequencies and is $\{45, 46, 47, \ldots 256\}$. Domain 2 also has 212 frequencies, and is $\{0, 1, 2, \ldots 211\}$. Number the carriers from 1 to 1794, with the 6 carriers of cell 1 first, then the 8 carriers of cell 2 etc.. If $i \leq 1000$ then carrier i must be assigned a frequency from domain $i \bmod 3$. If $i > 1000$ then carrier i must be assigned a frequency from domain 0.

14.3 Tables of Results

The results in this section are presented as follows, unless stated otherwise. The first column gives the algorithm used:

- SA constraints: simulated annealing binary constraint solving algorithm ;
- SA SIR random: SIR simulated annealing algorithm from a random start with no binary constraints;
- SA SIR hybrid: a simulated annealing algorithm with a (weighted) use of number of binary constraint violations and SIR cost in the cost function;
- SA SIR (bc start): SIR simulated annealing algorithm using a starting assignment from a binary constraint (bc) solution.

The second column gives the SIR threshold when generating the binary constraints, the third gives the weight assigned to the number of violations of the binary constraints for the hybrid algorithm, the fourth is the weight assigned to the SIR based cost, the fifth is the SIR cost and the last column is either the network coverage, i.e. the percentage of reception points with an adequate SIR, or the runtime given in hours.

The experiment parameters presented here are derived from experimental trials, and appear suitable to the specific problems presented in this chapter.

The results presented in table 14.3, table 14.4, table 14.5, table 14.6, table 14.7, table 14.8, table 14.9, table 14.10, table 14.11 and table 14.12 used the following algorithm parameters:

- Start temperature: 1.0
- Stop temperature: 10^{-18}
- Iterations per temperature: 1500
- Cooling Ratio: 0.987

The results presented in table 14.13 and table 14.14 used the following algorithm parameters:

- Start temperature: *as given in tables*
- Stop temperature: 10^{-18}

- Iterations per temperature: 2000
- Cooling Ratio: 0.999

The results presented in table 14.15 used the following parameters:

- Start temperature: *as given in tables*
- Stop temperature: 10^{-18}
- Iterations per temperature: 700
- Cooling Ratio: 0.999

Method	Threshold when generating constraints (dB)	Constraint weight	SIR weight	Cost	Coverage
SA constraints	10	-	-	**2876.91**	**72.64%**
SA constraints	11	-	-	4090.69	69.50%
SA constraints	12	-	-	5837.25	60.63%
SA constraints	13	-	-	7453.10	57.12%
SA constraints	14	-	-	8479.08	51.39%
SA constraints	15	-	-	13593.54	35.12%
SA SIR random	-	-	-	**1764.09**	**75.05%**
SA SIR hybrid	10	1	1	1764.09	75.05%
SA SIR hybrid	11	1	1	1758.17	76.34%
SA SIR hybrid	12	1	1	**1755.43**	**76.16%**
SA SIR hybrid	13	1	1	1860.94	72.27%
SA SIR hybrid	14	1	1	1892.36	74.86%
SA SIR hybrid	15	1	1	1829.14	74.49%
SA SIR hybrid	10	10	1	1875.71	73.01%
SA SIR hybrid	11	10	1	2013.96	76.52%
SA SIR hybrid	12	10	1	1858.46	77.63%
SA SIR hybrid	13	10	1	2008.82	76.16%
SA SIR hybrid	14	10	1	1988.22	73.57%
SA SIR hybrid	15	10	1	2019.54	76.71%
SA SIR hybrid	10	100	1	2434.29	76.89%
SA SIR hybrid	11	100	1	2267.90	74.49%
SA SIR hybrid	12	100	1	2370.11	73.57%
SA SIR hybrid	13	100	1	3730.46	68.39%
SA SIR hybrid	14	100	1	3118.89	75.79%
SA SIR hybrid	15	100	1	4962.55	63.40%
SA SIR (bc start)	10	-	-	1727.97	73.20%
SA SIR (bc start)	11	-	-	**1663.63**	**76.52%**
SA SIR (bc start)	12	-	-	1783.82	76.89%
SA SIR (bc start)	13	-	-	1728.51	74.68%
SA SIR (bc start)	14	-	-	1862.50	73.94%
SA SIR (bc start)	15	-	-	1750.63	75.05%

Table 14.3: Results for the 95 transmitter problem with a domain of 10 consecutive frequencies. The required SIR is 9dB.

Method	Threshold when generating constraints (dB)	Constraint weight	SIR weight	Cost	Coverage
SA constraints	10	-	-	**2187.33**	**77.08%**
SA constraints	11	-	-	3239.84	76.16%
SA constraints	12	-	-	3932.03	72.09%
SA constraints	13	-	-	6071.86	62.48%
SA constraints	14	-	-	7930.63	55.64%
SA constraints	15	-	-	8941.17	52.87%
SA SIR random	-	-	-	**1173.23**	**79.30%**
SA SIR hybrid	10	1	1	1173.23	79.30%
SA SIR hybrid	11	1	1	**1136.60**	**78.56%**
SA SIR hybrid	12	1	1	1141.81	77.63%
SA SIR hybrid	13	1	1	1201.35	79.11%
SA SIR hybrid	14	1	1	1287.73	78.74%
SA SIR hybrid	15	1	1	1308.93	79.85%
SA SIR hybrid	10	10	1	1348.49	79.30%
SA SIR hybrid	11	10	1	1280.29	79.11%
SA SIR hybrid	12	10	1	1352.46	80.41%
SA SIR hybrid	13	10	1	1250.09	79.85%
SA SIR hybrid	14	10	1	1483.77	80.22%
SA SIR hybrid	15	10	1	1480.69	78.93%
SA SIR hybrid	10	100	1	1655.61	79.85%
SA SIR hybrid	11	100	1	1771.89	79.30%
SA SIR hybrid	12	100	1	2016.72	77.63%
SA SIR hybrid	13	100	1	2408.30	74.31%
SA SIR hybrid	14	100	1	3311.52	74.49%
SA SIR hybrid	15	100	1	4021.19	69.13%
SA SIR (bc start)	10	-	-	1192.59	80.04%
SA SIR (bc start)	11	-	-	1272.58	78.56%
SA SIR (bc start)	12	-	-	**1093.65**	**79.85%**
SA SIR (bc start)	13	-	-	1175.21	78.74%
SA SIR (bc start)	14	-	-	1230.70	79.30%
SA SIR (bc start)	15	-	-	1131.84	80.22%

Table 14.4: Results for the 95 transmitter problem with a domain of 11 consecutive frequencies. The required SIR is 9dB.

Method	Threshold when generating constraints (dB)	Constraint weight	SIR weight	Cost	Coverage
SA constraints	16	-	-	**6174.69**	**86.51**%
SA constraints	17	-	-	13972.97	88.91%
SA constraints	18	-	-	51826.42	83.36%
SA constraints	19	-	-	74492.14	79.11%
SA constraints	20	-	-	97337.39	74.12%
SA constraints	21	-	-	157580.34	58.78%
SA constraints	22	-	-	181028.94	51.76%
SA SIR random	-	-	-	**7399.29**	**79.48**%
SA SIR hybrid	16	1	1	7399.19	79.48%
SA SIR hybrid	17	1	1	7399.19	79.48%
SA SIR hybrid	18	1	1	7399.19	79.48%
SA SIR hybrid	19	1	1	7399.19	79.48%
SA SIR hybrid	20	1	1	7399.19	79.48%
SA SIR hybrid	21	1	1	7399.19	79.48%
SA SIR hybrid	22	1	1	7399.19	79.48%
SA SIR hybrid	16	10	1	6545.75	80.41%
SA SIR hybrid	17	10	1	7399.29	79.48%
SA SIR hybrid	18	10	1	**5312.66**	**81.33**%
SA SIR hybrid	19	10	1	5905.29	84.10%
SA SIR hybrid	20	10	1	6825.47	79.48%
SA SIR hybrid	21	10	1	6576.08	81.33%
SA SIR hybrid	22	10	1	7747.17	81.89%
SA SIR hybrid	16	100	1	7685.27	85.03%
SA SIR hybrid	17	100	1	8003.01	81.70%
SA SIR hybrid	18	100	1	9029.39	83.18%
SA SIR hybrid	19	100	1	6429.85	82.81%
SA SIR hybrid	20	100	1	9162.43	81.15%
SA SIR hybrid	21	100	1	5595.40	84.47%
SA SIR hybrid	22	100	1	7793.29	82.99%
SA SIR (bc start)	16	-	-	3768.78	86.88%
SA SIR (bc start)	17	-	-	**3333.16**	**89.83**%
SA SIR (bc start)	18	-	-	4911.03	89.09%
SA SIR (bc start)	19	-	-	4480.12	85.40%
SA SIR (bc start)	20	-	-	7963.25	85.40%
SA SIR (bc start)	21	-	-	7186.38	82.07%
SA SIR (bc start)	22	-	-	7480.02	82.99%

Table 14.5: Results for the 95 transmitter problem with a domain of 25 consecutive frequencies. The required SIR is 15dB.

Method	Threshold when generating constraints (dB)	Constraint weight	SIR weight	Cost	Coverage
SA constraints	16	-	-	**4559.20**	**90.02%**
SA constraints	17	-	-	7662.88	91.87%
SA constraints	18	-	-	33335.95	85.77%
SA constraints	19	-	-	63961.88	82.62%
SA constraints	20	-	-	73658.70	79.85%
SA constraints	21	-	-	113786.01	66.17%
SA constraints	22	-	-	188548.34	51.76%
SA SIR random	-	-	-	**8003.20**	**80.78%**
SA SIR hybrid	16	1	1	8003.20	80.78%
SA SIR hybrid	17	1	1	8003.20	80.78%
SA SIR hybrid	18	1	1	8003.20	80.78%
SA SIR hybrid	19	1	1	8003.18	81.70%
SA SIR hybrid	20	1	1	7969.05	82.44%
SA SIR hybrid	21	1	1	7969.05	82.44%
SA SIR hybrid	22	1	1	4773.93	85.21%
SA SIR hybrid	16	10	1	**4043.61**	**89.28%**
SA SIR hybrid	17	10	1	5790.42	84.66%
SA SIR hybrid	18	10	1	6490.86	83.55%
SA SIR hybrid	19	10	1	6188.72	82.07%
SA SIR hybrid	20	10	1	5672.45	85.40%
SA SIR hybrid	21	10	1	5574.33	83.73%
SA SIR hybrid	22	10	1	8864.49	80.78%
SA SIR hybrid	16	100	1	4890.41	85.40%
SA SIR hybrid	17	100	1	6278.87	85.21%
SA SIR hybrid	18	100	1	7888.21	81.33%
SA SIR hybrid	19	100	1	6135.99	83.92%
SA SIR hybrid	20	100	1	5222.44	87.80%
SA SIR hybrid	21	100	1	7814.69	85.77%
SA SIR hybrid	22	100	1	7247.34	83.36%
SA SIR (bc start)	16	-	-	2870.37	88.72%
SA SIR (bc start)	17	-	-	**2030.41**	**90.02%**
SA SIR (bc start)	18	-	-	3292.46	88.72%
SA SIR (bc start)	19	-	-	4467.94	88.17%
SA SIR (bc start)	20	-	-	3859.06	85.77%
SA SIR (bc start)	21	-	-	5173.06	85.21%
SA SIR (bc start)	22	-	-	6320.76	82.81%

Table 14.6: Results for the 95 transmitter problem with a domain of 26 consecutive frequencies. The required SIR is 15dB.

Method	Threshold when generating constraints (dB)	Constraint weight	SIR weight	Cost	Coverage
SA constraints	10	-	-	**2979.59**	**81.83%**
SA constraints	11	-	-	3317.72	85.12%
SA constraints	12	-	-	5440.34	84.45%
SA constraints	13	-	-	11008.01	81.01%
SA constraints	14	-	-	16661.85	77.50%
SA constraints	15	-	-	34683.79	62.58%
SA SIR random	-	-	-	**2593.51**	**83.03%**
SA SIR hybrid	10	1	1	**2533.98**	**82.99%**
SA SIR hybrid	11	1	1	2700.07	83.29%
SA SIR hybrid	12	1	1	2750.81	82.77%
SA SIR hybrid	13	1	1	2710.06	83.14%
SA SIR hybrid	14	1	1	2564.24	83.93%
SA SIR hybrid	15	1	1	2729.10	83.74%
SA SIR hybrid	10	10	1	2958.46	82.28%
SA SIR hybrid	11	10	1	2752.38	82.92%
SA SIR hybrid	12	10	1	2914.72	83.18%
SA SIR hybrid	13	10	1	2910.30	83.51%
SA SIR hybrid	14	10	1	3350.51	82.69%
SA SIR hybrid	15	10	1	3325.45	82.47%
SA SIR hybrid	10	100	1	3050.59	83.85%
SA SIR hybrid	11	100	1	2934.29	84.45%
SA SIR hybrid	12	100	1	3933.96	83.71%
SA SIR hybrid	13	100	1	5547.90	82.88%
SA SIR hybrid	14	100	1	6704.00	80.90%
SA SIR hybrid	15	100	1	11489.74	77.08%
SA SIR (bc start)	10	-	-	2085.12	83.74%
SA SIR (bc start)	11	-	-	**1826.39**	**85.72%**
SA SIR (bc start)	12	-	-	2221.76	86.13%
SA SIR (bc start)	13	-	-	2497.30	84.56%
SA SIR (bc start)	14	-	-	2620.42	84.00%
SA SIR (bc start)	15	-	-	2754.37	82.50%

Table 14.7: Results for the 458 transmitter problem with a domain of 10 consecutive frequencies. The required SIR is 9dB.

Method	Threshold when generating constraints (dB)	Constraint weight	SIR weight	Cost	Coverage
SA constraints	10	-	-	2197.91	85.94%
SA constraints	11	-	-	1591.95	88.45%
SA constraints	12	-	-	**1197.18**	**90.13**%
SA constraints	13	-	-	3251.21	90.73%
SA constraints	14	-	-	8239.61	87.21%
SA constraints	15	-	-	26402.69	71.55%
SA SIR random	-	-	-	**932.44**	**89.57**%
SA SIR hybrid	10	1	1	910.83	89.91%
SA SIR hybrid	11	1	1	**875.75**	**89.76**%
SA SIR hybrid	12	1	1	954.85	89.72%
SA SIR hybrid	13	1	1	1179.08	89.42%
SA SIR hybrid	14	1	1	1012.28	89.57%
SA SIR hybrid	15	1	1	1253.22	88.56%
SA SIR hybrid	10	10	1	1098.83	89.61%
SA SIR hybrid	11	10	1	1208.69	89.05%
SA SIR hybrid	12	10	1	1385.03	88.79%
SA SIR hybrid	13	10	1	1330.71	89.16%
SA SIR hybrid	14	10	1	1530.08	89.68%
SA SIR hybrid	15	10	1	1675.86	88.26%
SA SIR hybrid	10	100	1	1117.09	89.27%
SA SIR hybrid	11	100	1	1322.55	90.36%
SA SIR hybrid	12	100	1	1390.98	89.87%
SA SIR hybrid	13	100	1	2426.63	88.22%
SA SIR hybrid	14	100	1	3707.88	87.36%
SA SIR hybrid	15	100	1	7610.70	84.00%
SA SIR (bc start)	10	-	-	963.26	89.42%
SA SIR (bc start)	11	-	-	868.88	90.58%
SA SIR (bc start)	12	-	-	**816.48**	**90.62**%
SA SIR (bc start)	13	-	-	822.60	91.44%
SA SIR (bc start)	14	-	-	1023.97	90.92%
SA SIR (bc start)	15	-	-	1050.77	89.79%

Table 14.8: Results for the 458 transmitter problem with a domain of 11 consecutive frequencies. The required SIR is 9dB.

Method	Threshold when generating constraints (dB)	Constraint weight	SIR weight	Cost	Coverage
SA constraints	16	-	-	24246.13	89.16%
SA constraints	17	-	-	17030.49	91.78%
SA constraints	18	-	-	9103.20	93.57%
SA constraints	19	-	-	**5770.89**	**94.95%**
SA constraints	20	-	-	19211.22	95.96%
SA constraints	21	-	-	141917.59	91.33%
SA constraints	22	-	-	303417.56	83.51%
SA SIR random	-	-	-	**4435.04**	**91.78%**
SA SIR hybrid	16	1	1	5498.83	90.69%
SA SIR hybrid	17	1	1	6100.17	91.07%
SA SIR hybrid	18	1	1	4316.49	92.26%
SA SIR hybrid	19	1	1	5213.78	90.80%
SA SIR hybrid	20	1	1	**3747.58**	**92.15%**
SA SIR hybrid	21	1	1	5486.67	90.88%
SA SIR hybrid	22	1	1	3999.78	91.78%
SA SIR hybrid	16	10	1	5284.90	91.89%
SA SIR hybrid	17	10	1	6061.33	92.04%
SA SIR hybrid	18	10	1	4828.55	93.16%
SA SIR hybrid	19	10	1	4551.71	92.07%
SA SIR hybrid	20	10	1	7539.21	89.76%
SA SIR hybrid	21	10	1	5905.81	90.80%
SA SIR hybrid	22	10	1	6281.16	90.84%
SA SIR hybrid	16	100	1	4140.00	94.17%
SA SIR hybrid	17	100	1	5543.99	94.02%
SA SIR hybrid	18	100	1	5507.54	94.13%
SA SIR hybrid	19	100	1	7922.41	93.50%
SA SIR hybrid	20	100	1	6641.69	93.79%
SA SIR hybrid	21	100	1	9354.31	91.70%
SA SIR hybrid	22	100	1	9707.13	92.15%
SA SIR (bc start)	16	-	-	3730.83	92.41%
SA SIR (bc start)	17	-	-	2834.62	93.87%
SA SIR (bc start)	18	-	-	1848.33	93.64%
SA SIR (bc start)	19	-	-	**1222.93**	**95.59%**
SA SIR (bc start)	20	-	-	1718.81	95.18%
SA SIR (bc start)	21	-	-	4137.47	93.53%
SA SIR (bc start)	22	-	-	6772.52	90.84%

Table 14.9: Results for the 458 transmitter problem with a domain of 24 consecutive frequencies. The required SIR is 15dB.

Method	Threshold when generating constraints (dB)	Constraint weight	SIR weight	Cost	Coverage
SA constraints	16	-	-	24442.58	90.65%
SA constraints	17	-	-	13597.52	92.56%
SA constraints	18	-	-	8481.14	94.92%
SA constraints	19	-	-	**6105.58**	**95.66%**
SA constraints	20	-	-	16480.58	96.56%
SA constraints	21	-	-	65964.61	95.96%
SA constraints	22	-	-	228352.05	87.59%
SA SIR random	-	-	-	**2717.77**	**92.79%**
SA SIR hybrid	16	1	1	2095.60	93.57%
SA SIR hybrid	17	1	1	2366.27	94.73%
SA SIR hybrid	18	1	1	3676.78	92.67%
SA SIR hybrid	19	1	1	2233.45	93.83%
SA SIR hybrid	20	1	1	1180.34	94.73%
SA SIR hybrid	21	1	1	1738.48	94.73%
SA SIR hybrid	22	1	1	2322.04	94.02%
SA SIR hybrid	16	10	1	1412.98	95.89%
SA SIR hybrid	17	10	1	2024.49	94.47%
SA SIR hybrid	18	10	1	**1159.60**	**96.11%**
SA SIR hybrid	19	10	1	1567.60	95.33%
SA SIR hybrid	20	10	1	1924.20	94.50%
SA SIR hybrid	21	10	1	2439.75	94.84%
SA SIR hybrid	22	10	1	2557.93	93.31%
SA SIR hybrid	16	100	1	2803.31	95.33%
SA SIR hybrid	17	100	1	2214.59	95.70%
SA SIR hybrid	18	100	1	3204.22	95.66%
SA SIR hybrid	19	100	1	3722.49	95.33%
SA SIR hybrid	20	100	1	5576.22	93.68%
SA SIR hybrid	21	100	1	6427.26	95.14%
SA SIR hybrid	22	100	1	6613.85	92.93%
SA SIR (bc start)	16	-	-	1627.86	94.47%
SA SIR (bc start)	17	-	-	1062.46	95.51%
SA SIR (bc start)	18	-	-	1344.78	95.63%
SA SIR (bc start)	19	-	-	**508.66**	**95.70%**
SA SIR (bc start)	20	-	-	1580.68	96.41%
SA SIR (bc start)	21	-	-	1482.79	95.70%
SA SIR (bc start)	22	-	-	1254.99	94.69%

Table 14.10: Results for the 458 transmitter problem with a domain of 25 consecutive frequencies. The required SIR is 15dB.

Domain	Start temperature	End temperature	Cost	Coverage
20	1.0	10^{-18}	846514.375	51.18%
25	1.0	10^{-18}	728834.380	57.16%
30	1.0	10^{-18}	625882.000	63.51%
35	1.0	10^{-18}	567513.690	66.43%
40	1.0	10^{-18}	513595.030	70.17%
45	1.0	10^{-18}	457442.160	72.97%
50	1.0	10^{-18}	392625.690	74.54%
100	1.0	10^{-18}	149178.560	92.41%
150	1.0	10^{-18}	53630.195	97.01%
200	1.0	10^{-18}	503.837	99.85%
250	1.0	10^{-18}	0.000	100.00%

Table 14.11: Results for the terrain database computed 458 transmitter problem of section 14.2.2 using the pure SIR simulated annealing algorithm from a random start and evaluated at a required SIR of 15dB.

Method	Threshold when generating constraints (dB)	Constraint weight	SIR weight	Cost	Coverage
SA constraints	16	-	-	**92058.06**	**95.18%**
SA constraints	17	-	-	101776.79	95.03%
SA constraints	18	-	-	118656.51	94.77%
SA constraints	19	-	-	113259.37	95.07%
SA constraints	20	-	-	102326.44	95.70%
SA constraints	21	-	-	141153.70	94.09%
SA constraints	22	-	-	143214.70	93.72%
SA SIR random	-	-	-	**53630.20**	**97.01%**
SA SIR hybrid	16	1	1	49975.21	97.31%
SA SIR hybrid	17	1	1	50990.32	97.83%
SA SIR hybrid	18	1	1	56722.70	97.23%
SA SIR hybrid	19	1	1	57900.30	97.05%
SA SIR hybrid	20	1	1	58188.44	96.93%
SA SIR hybrid	21	1	1	57968.89	97.31%
SA SIR hybrid	22	1	1	57883.29	96.97%
SA SIR hybrid	16	10	1	**46399.11**	**97.42%**
SA SIR hybrid	17	10	1	56169.20	96.93%
SA SIR hybrid	18	10	1	52979.41	97.31%
SA SIR hybrid	19	10	1	53962.79	97.31%
SA SIR hybrid	20	10	1	57800.41	97.01%
SA SIR hybrid	21	10	1	66439.12	96.79%
SA SIR hybrid	22	10	1	52131.13	97.16%
SA SIR hybrid	16	100	1	51477.99	97.23%
SA SIR hybrid	17	100	1	50605.88	97.38%
SA SIR hybrid	18	100	1	51988.30	97.23%
SA SIR hybrid	19	100	1	49038.24	97.46%
SA SIR hybrid	20	100	1	55470.90	97.20%
SA SIR hybrid	21	100	1	63946.96	96.97%
SA SIR hybrid	22	100	1	68780.91	96.41%
SA SIR (bc start)	16	-	-	**26780.16**	**98.24%**
SA SIR (bc start)	17	-	-	32160.16	98.36%
SA SIR (bc start)	18	-	-	27697.92	98.43%
SA SIR (bc start)	19	-	-	29704.21	98.62%
SA SIR (bc start)	20	-	-	30317.24	98.47%
SA SIR (bc start)	21	-	-	37644.61	98.17%
SA SIR (bc start)	22	-	-	35487.37	98.32%

Table 14.12: Results for terrain database computed 458 transmitter problem with a domain of 150 consecutive frequencies. The required SIR is 15dB.

Method	Threshold when generating constraints (dB)	Constraint weight	SIR weight	Start temperature	Cost	Run Time (hours)
SA constraints	15	-	-	1.0	14983.24	0.02
SA constraints	16	-	-	1.0	9761.59	0.02
SA constraints	17	-	-	1.0	4885.82	0.02
SA constraints	18	-	-	1.0	**3664.27**	1.00
SA constraints	19	-	-	1.0	50452.50	0.10
SA constraints	20	-	-	1.0	140974.77	0.10
SA SIR random	-	-	-	1.0	**235.00**	17.86
SA SIR hybrid	17	1	1	1.0	**75.01**	11.10
SA SIR (bc start)	15	-	-	0.001	96.27	20.02
SA SIR (bc start)	16	-	-	0.001	86.43	18.02
SA SIR (bc start)	17	-	-	0.001	112.71	16.29
SA SIR (bc start)	18	-	-	0.001	**63.07**	16.40
SA SIR (bc start)	19	-	-	0.0001	542.17	15.08
SA SIR (bc start)	20	-	-	0.0001	57.24	17.03

Table 14.13: Results for the 458 transmitter problem as presented in [21]. A domain of up to 28 frequencies are available, but some frequencies are blocked for certain transmitters. The required SIR is 14dB. The table illustrated typical runtimes for the various methods on a 2.67GHz XEON processor.

Method	Threshold when generating constraints (dB)	Constraint weight	SIR weight	Start temperature	Cost	Run Time (hours)
SA constraints	15	-	-	1.0	84797.09	0.007
SA constraints	16	-	-	1.0	23833.22	0.01
SA constraints	17	-	-	1.0	23833.22	0.01
SA constraints	18	-	-	1.0	**2723.63**	0.11
SA constraints	19	-	-	1.0	742426.88	0.94
SA SIR random	-	-	-	1.0	**4009.22**	12.57
SA SIR hybrid	18	1	1	1.0	**3109.29**	12.64
SA SIR (bc start)	15	-	-	0.001	6490.46	2.84
SA SIR (bc start)	16	-	-	0.001	6014.93	6.84
SA SIR (bc start)	17	-	-	0.001	4129.34	9.34
SA SIR (bc start)	18	-	-	0.0001	**905.56**	8.43
SA SIR (bc start)	19	-	-	0.0001	29487.70	8.56

Table 14.14: Results for the HEX358 problem as presented in [21]. A domain of up to 117 frequencies are available, but some frequencies are blocked for certain transmitters. The required SIR is 15dB. The table illustrated typical runtimes for the various methods on a 2.67GHz XEON processor.

Method	Threshold when generating constraints (dB)	Constraint weight	SIR weight	Start temperature	Cost	Run Time (hours)
SA constraints	15	-	-	1.0	287445.97	0.01
SA constraints	16	-	-	1.0	161603.89	0.01
SA constraints	17	-	-	1.0	161603.89	0.01
SA constraints	18	-	-	1.0	97248.54	0.02
SA constraints	19	-	-	1.0	19345.59	1.63
SA constraints	20	-	-	1.0	**15724.39**	1.55
SA constraints	21	-	-	1.0	1577874.50	22.70
SA SIR random	-	-	-	1.0	**2982.14**	18.33
SA SIR hybrid	19	1	1	1.0	**4361.56**	15.91
SA SIR hybrid	20	1	1	1.0	5557.26	22.80
SA SIR (bc start)	15	-	-	0.001	2437.00	22.83
SA SIR (bc start)	16	-	-	0.001	2454.00	16.51
SA SIR (bc start)	17	-	-	0.001	2454.00	13.44
SA SIR (bc start)	18	-	-	0.001	2781.71	23.06
SA SIR (bc start)	19	-	-	0.0001	1080.20	12.68
SA SIR (bc start)	20	-	-	0.0001	**358.93**	18.17

Table 14.15: Results for the HEX1794 problem as presented in [21]. A domain of up to 257 frequencies are available, but some frequencies are blocked for certain transmitters. The required SIR is 15dB. The table illustrated typical runtimes for the various methods on a 2.67GHz XEON processor.

14.4 Analysis of Results

14.4.1 95 Transmitter SIR Problem, 9dB SIR Evaluation

This set of experiments were performed on the 95 transmitter problem at an SIR threshold of 9dB. Domains of 10 and 11 consecutive frequencies were used for the results presented in table 14.3 and table 14.4. The best results for each method are highlighted in bold. In both cases the best overall result is obtained using the simulated annealing pure SIR algorithm with a binary constraint starting assignment.

Analysis of the Constraint Weight

The results from the 95 transmitter problem evaluated at an SIR threshold of 9dB can be used to provide understanding on the use of a weight on the binary constraint cost function component. The weights are used in the simulated annealing hybrid algorithm and may give more priority to the solving of binary constraints rather than the reduction of the SIR cost. The weight values that are being compared are 1, 10 and 100. The constraints were generated at SIR thresholds of 10dB, 11dB, 12dB, 13dB, 14dB and 15dB.

Figure 14.4 presents a chart of the results obtained using the simulated annealing SIR hybrid algorithm on the 95 transmitter problem, using a domain of 10 consecutive frequencies. Using the results from table 14.3 the average SIR cost can be evaluated for each constraint weight and is presented in table 14.16. The averages show that using a constraint weight of 1 produced the best overall results while using a constraint weight of 100 produced the worst overall results.

Constraint weight	Average SIR Cost
1	1810.02
10	1960.79
100	3147.37

Table 14.16: Average SIR costs for the 95 transmitter problem solved at 9dB using a domain of 10 consecutive frequencies. Results are obtained from the simulated annealing hybrid algorithm.

Figure 14.5 presents a chart of the results obtained using the simulated annealing SIR hybrid algorithm on the 95 transmitter problem, using a domain of 11 consecutive frequencies. Using the results from table 14.4 the average SIR cost can be evaluated for each constraint weight and is presented in table 14.17. The averages show that using a constraint weight of 1 produced the best overall results

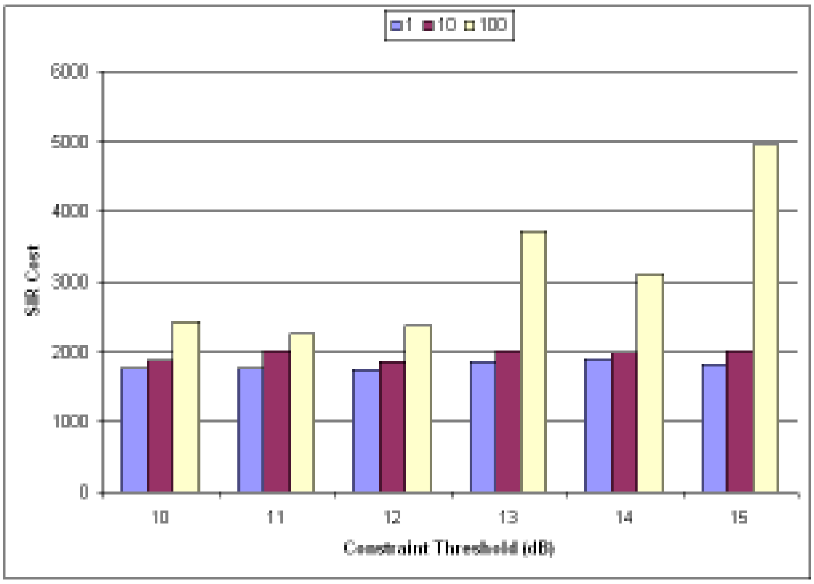

Figure 14.4: 95 transmitter problem evaluated at 9dB using a domain of 10 consecutive frequencies. Results show SIR costs for the hybrid algorithm using constraint weights of 1, 10 and 100.

while using a constraint weight of 100 produced the worst overall results.

Constraint weight	Average SIR Cost
1	1208.28
10	1365.97
100	2530.87

Table 14.17: Average SIR costs for the 95 transmitter problem solved at 9dB using a domain of 11 consecutive frequencies. Results are obtained from the simulated annealing hybrid algorithm.

In summary, the analysis shows that for this problem it is best to use a constraint weight of 1 when using the simulated annealing hybrid algorithm.

Analysis of the Binary Constraint Generation SIR Threshold

This section analyses the results to provide an understanding of the use of an appropriate SIR threshold when generating binary constraints. It is known that the binary constraints must be generated at a higher SIR threshold than the evaluation SIR threshold otherwise they would have little or no impact on the overall solution. What is not known is by how much the SIR evaluation threshold must

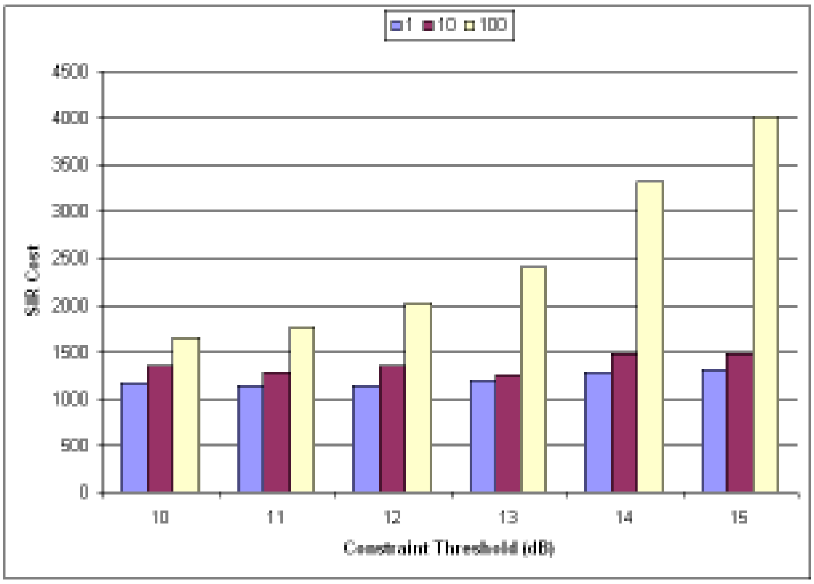

Figure 14.5: 95 transmitter problem evaluated at 9dB using a domain of 11 consecutive frequencies. Results show SIR costs for the hybrid algorithm using constraint weights of 1, 10 and 100.

exceed the binary constraint generation threshold in order to attain the best possible impact on the overall solution of an SIR problem. This section will attempt to obtain some conclusions based on results for domains of 10 and 11 consecutive frequencies with the binary constraints generated at SIR thresholds of 10dB, 11dB, 12dB, 13dB, 14dB and 15dB.

Figure 14.6 presents a chart of the results obtained by solving the 95 transmitter problem using the simulated annealing binary constraint solving algorithm. The chart presents the results for domains of 10 and 11 consecutive frequencies obtained from table 14.3 and table 14.4. It is clear from this chart that when used alone binary constraints generated at thresholds of 13dB, 14dB and 15dB give poor results when compared to lower thresholds.

Figure 14.7 presents a chart of the results obtained by solving the 95 transmitter problem using the simulated annealing pure SIR algorithm from a binary constraint start. The results presented in figure 14.6 are used as starting assignments. Figure 14.7 presents the results for domains of 10 and 11 consecutive frequencies. It can be deduced from the chart when a domain of 10 consecutive frequencies is used the best result is obtained from the binary constraints generated at a threshold of 11dB. When a domain of 11 consecutive frequencies is used the best result is obtained from the binary constraints

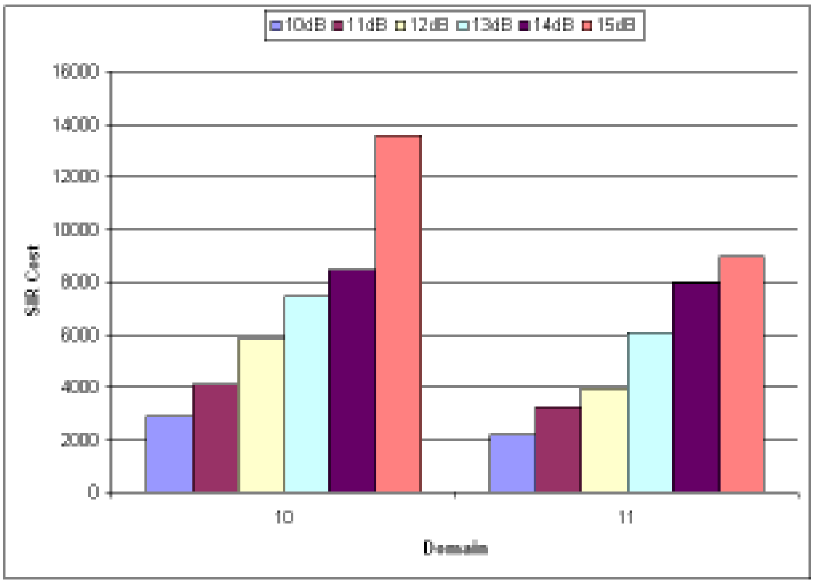

Figure 14.6: 95 transmitter problem evaluated at 9dB using domains of 10 and 11 consecutive frequencies. Results show SIR costs for the binary constraint solving simulated annealing algorithm algorithm. The binary constraints are generated at SIR thresholds of 10dB, 11dB, 12dB, 13dB, 14dB and 15dB.

generated at a threshold of 12dB. However, the results presented in figure 14.7 are good for all binary constraint thresholds used.

In summary, the solutions of the binary constraints showed that the best results were obtained by generating the binary constraints at up to 3dB higher than the evaluation SIR threshold. However, when the pure SIR simulated annealing algorithm was run using these results as starting assignments all of the results produced were generally good.

Analysis of the SIR Algorithms

This section analyses the results from the 95 transmitter problem evaluated at an SIR threshold of 9dB to show which of the SIR algorithms outlined in section 8.7 produced the best results.

Figure 14.8 presents a chart of all of the best results obtained from the algorithms used to solve the 95 transmitter problem. The results are obtained from table 14.3 and table 14.4. It is clear from the chart that the simulated annealing pure SIR algorithm from a binary constraint start produced

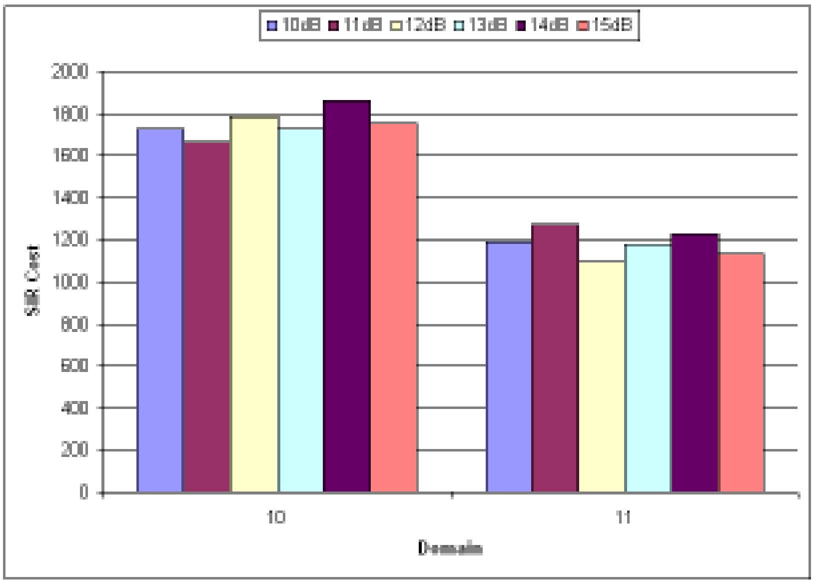

Figure 14.7: 95 transmitter problem evaluated at 9dB using domains of 10 and 11 consecutive frequencies. Results show SIR costs for the pure SIR simulated annealing algorithm algorithm from a binary constraint start. The binary constraints are generated at SIR thresholds of 10dB, 11dB, 12dB, 13dB, 14dB and 15dB.

the best results in both cases, with good results also being produced by the simulated annealing pure SIR algorithm from a random start and the simulated annealing hybrid algorithm. The simulated annealing binary constraint solving algorithm produced the poorest results by a substantial margin.

14.4.2 95 Transmitter SIR Problem, 15dB SIR Evaluation

This set of experiments were performed on the 95 transmitter problem at an SIR threshold of 15dB. Domains of 25 and 26 consecutive frequencies were used for the results presented in table 14.5 and table 14.6. The best results for each method are highlighted in bold. In both cases the best overall result is obtained using the simulated annealing pure SIR algorithm with a binary constraint starting assignment.

Analysis of the Constraint Weight

The results from the 95 transmitter problem evaluated at an SIR threshold of 9dB can be used to provide understanding on the use of a weight on the binary constraint cost function component. The weight values that are being compared are 1, 10 and 100. The constraints were generated at SIR

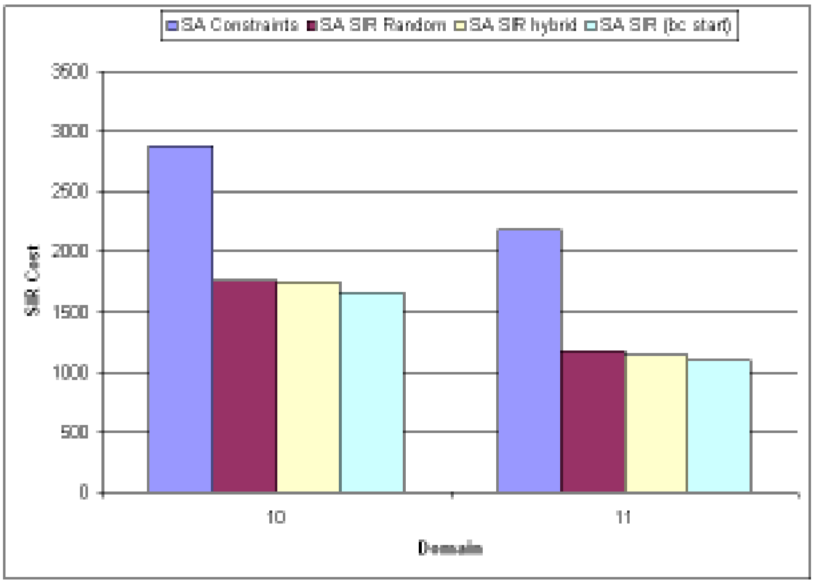

Figure 14.8: 95 transmitter problem evaluated at 9dB using domains of 10 and 11 consecutive frequencies. The chart gives the best results from all of the algorithms for each domain.

thresholds of 16dB, 17dB, 18dB, 19dB, 20dB, 21dB and 22dB.

Figure 14.9 presents a chart of the results obtained using the simulated annealing SIR hybrid algorithm on the 95 transmitter problem, using a domain of 25 consecutive frequencies. Using the results from table 14.5 the average SIR cost can be evaluated for each constraint weight and is presented in table 14.18. The averages show that using a constraint weight of 10 produced the best overall results while using a constraint weight of 100 produced the worst overall results.

Constraint weight	Average SIR Cost
1	7399.19
10	6615.96
100	7671.23

Table 14.18: Average SIR costs for the 95 transmitter problem solved at 15dB using a domain of 25 consecutive frequencies. Results are obtained from the simulated annealing hybrid algorithm.

Figure 14.10 presents a chart of the results obtained using the simulated annealing SIR hybrid algorithm on the 95 transmitter problem, using a domain of 26 consecutive frequencies. Using the results

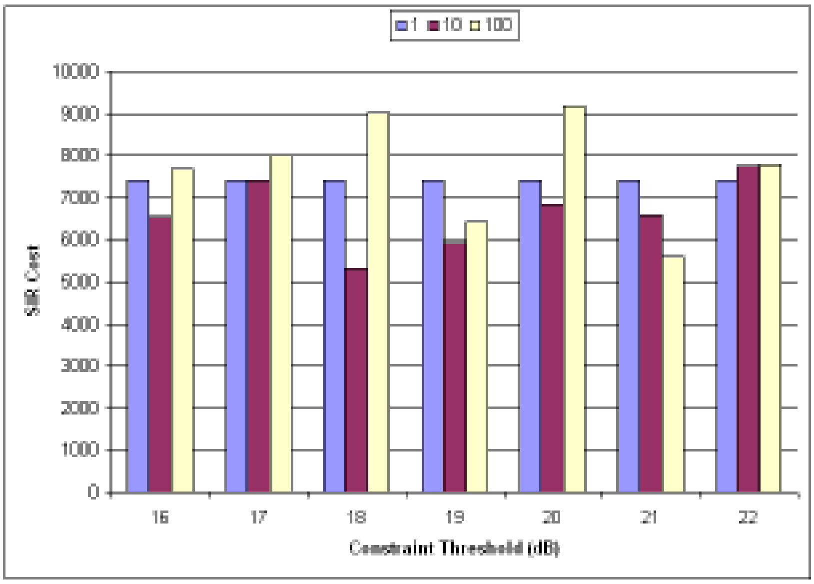

Figure 14.9: 95 transmitter problem evaluated at 15dB using a domain of 25 consecutive frequencies. Results show SIR costs for the hybrid algorithm using constraint weights of 1, 10 and 100.

from table 14.6 the average SIR cost can be evaluated for each constraint weight and is presented in table 14.19. The averages show that using a constraint weight of 10 produced the best overall results while using a constraint weight of 1 produced the worst overall results.

Constraint weight	Average SIR Cost
1	7532.12
10	6089.27
100	6496.85

Table 14.19: Average SIR costs for the 95 transmitter problem solved at 15dB using a domain of 26 consecutive frequencies. Results are obtained from the simulated annealing hybrid algorithm.

In summary, the analysis of the results in this section has revealed that in this case the best results are obtained for the simulated annealing hybrid algorithm using a constraint weight of 10.

Analysis of the Binary Constraint Generation SIR Threshold

This section analyses the results from the 95 transmitter problem evaluated at an SIR threshold of 15dB to provide an understanding on the use of an appropriate SIR threshold when generating binary

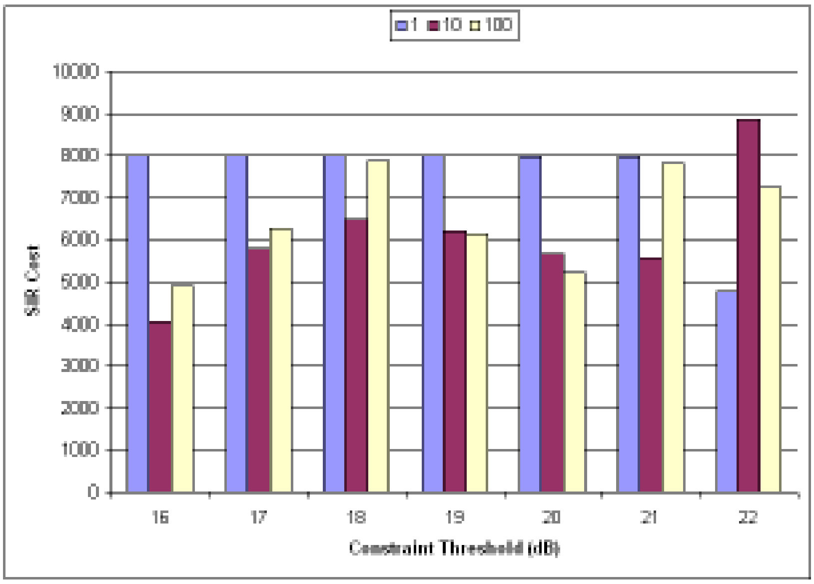

Figure 14.10: 95 transmitter problem evaluated at 15dB using a domain of 26 consecutive frequencies. Results show SIR costs for the hybrid algorithm using constraint weights of 1, 10 and 100.

constraints.

Figure 14.11 presents a chart of the results obtained by solving the 458 transmitter problem using the simulated annealing binary constraint solving algorithm. The chart presents the results for domains of 25 and 26 consecutive frequencies and are obtained from table 14.5 and table 14.6. It is clear from this chart that binary constraints generated at thresholds of 18dB, 19dB, 20dB, 21dB and 22dB give poor results when compared to lower thresholds.

Figure 14.12 presents a chart of the results obtained by solving the 95 transmitter problem using the simulated annealing pure SIR algorithm from a binary constraint start. The results presented in figure 14.11 are used as starting assignments. Figure 14.12 presents the results for domains of 25 and 26 consecutive frequencies. It can be deduced from the chart that when a domain of 25 consecutive frequencies is used the best result is obtained from the binary constraints generated at a threshold of 17dB. When a domain of 26 consecutive frequencies is used the best result is also obtained from the binary constraints generated at a threshold of 17dB.

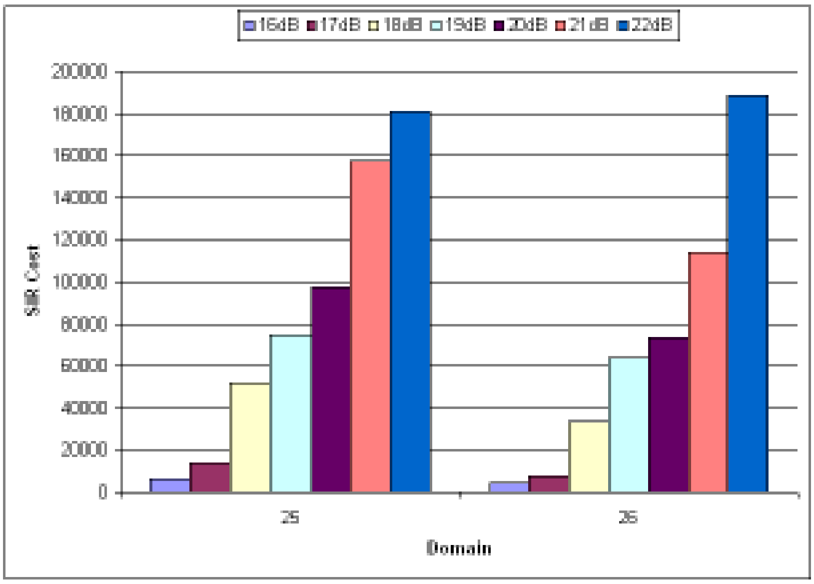

Figure 14.11: 95 transmitter problem evaluated at 15dB using domains of 25 and 26 consecutive frequencies. Results show SIR costs for the binary constraint solving simulated annealing algorithm algorithm. The binary constraints are generated at SIR thresholds of 16dB, 17dB, 18dB, 19dB, 20dB, 21dB and 22dB.

In summary, the analysis performed in this section has shown that binary constraints generated at thresholds more than 3dB above the evaluation SIR threshold produce poor results when compared to lower binary constraint generation thresholds. For these results the binary constraints generated at 17dB gave the best results overall.

Analysis of the SIR Algorithms

This section analyses the results from the 95 transmitter problem evaluated at an SIR threshold of 15dB to show which of the SIR algorithms outlined in section 8.7 produced the best results.

Figure 14.13 presents a chart of all of the best results obtained from the algorithms used to solve the 95 transmitter problem. The results are obtained from table 14.5 and table 14.6. The chart presents results for domains of 25 and 26 consecutive frequencies. It is clear from the chart that the simulated annealing pure SIR algorithm from a binary constraint start produced the best results in both cases by substantial margins. The simulated annealing binary constraint solving algorithm and the simulated annealing pure SIR algorithm from a random start produced the poorest results.

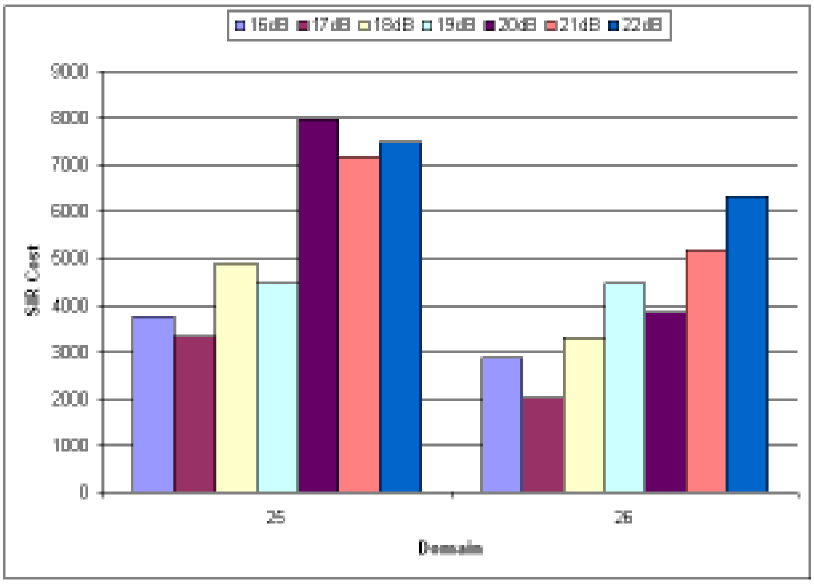

Figure 14.12: 95 transmitter problem evaluated at 15dB using domains of 25 and 26 consecutive frequencies. Results show SIR costs for the pure SIR simulated annealing algorithm algorithm from a binary constraint start. The binary constraints are generated at SIR thresholds of 16dB, 17dB, 18dB, 19dB, 20dB, 21dB and 22dB.

14.4.3 458 Transmitter Problem, 9dB SIR Evaluation

This set of experiments were evaluated at an SIR threshold of 9dB. Domains of 10 and 11 consecutive frequencies are used for the results presented in table 14.7 and table 14.8. The best results for each method are highlighted in bold. In both cases the best overall result is obtained using the simulated annealing pure SIR algorithm with a binary constraint starting assignment.

Analysis of the Constraint Weight

The results from the 458 transmitter problem evaluated at an SIR threshold of 9dB can be used to provide understanding on the use of a weight on the binary constraint cost function component. The weight values that are being compared are 1, 10 and 100. The constraints were generated at SIR thresholds of 10dB, 11dB, 12dB, 13dB, 14dB and 15dB.

Figure 14.14 presents a chart of the results obtained using the simulated annealing SIR hybrid algorithm on the 458 transmitter problem, using a domain of 10 consecutive frequencies. Using the results from table 14.7 the average SIR cost can be evaluated for each constraint weight and is presented in

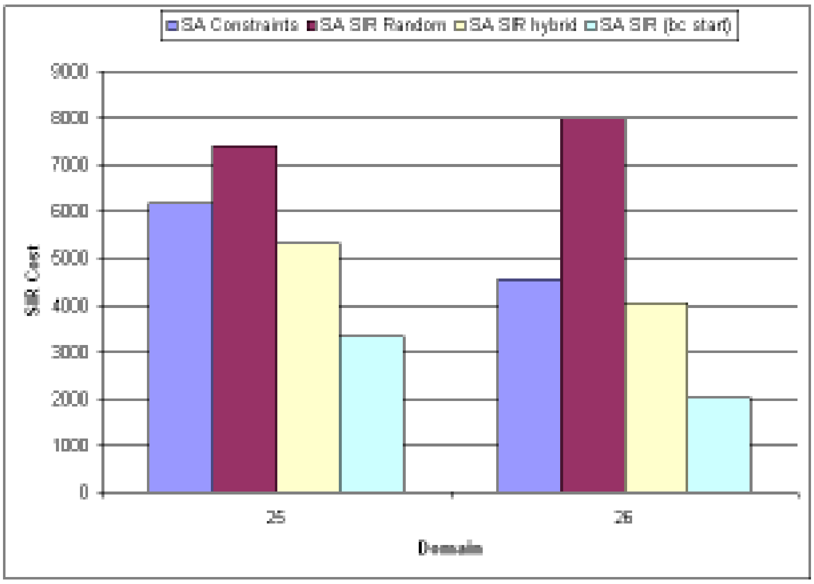

Figure 14.13: 95 transmitter problem evaluated at 15dB using domains of 25 and 26 consecutive frequencies. The chart gives the best results from all of the algorithms for each domain.

table 14.20. The averages show that using a constraint weight of 1 produced the best overall results while using a constraint weight of 100 produced the worst overall results.

Constraint weight	Average SIR Cost
1	2664.71
10	3035.30
100	5610.08

Table 14.20: Average SIR costs for the 458 transmitter problem solved at 9dB using a domain of 10 consecutive frequencies. Results are obtained from the simulated annealing hybrid algorithm.

Figure 14.15 presents a chart of the results obtained using the simulated annealing SIR hybrid algorithm on the 458 transmitter problem, using a domain of 11 consecutive frequencies. Using the results from table 14.8 the average SIR cost can be evaluated for each constraint weight and is presented in table 14.21. The averages show that using a constraint weight of 1 produced the best overall results while using a constraint weight of 100 produced the worst overall results.

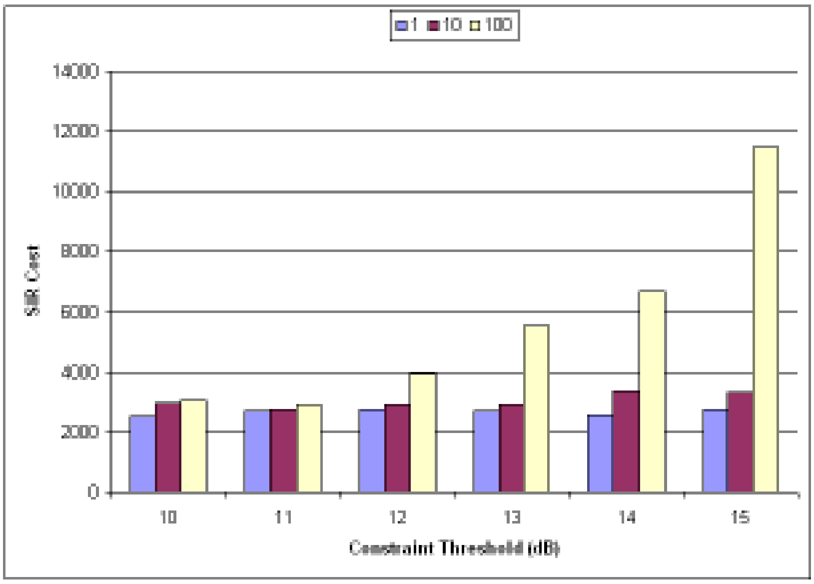

Figure 14.14: 458 transmitter problem evaluated at 9dB using a domain of 10 consecutive frequencies. Results show SIR costs for the hybrid algorithm using constraint weights of 1, 10 and 100.

Constraint weight	Average SIR Cost
1	1031.00
10	1371.53
100	2329.31

Table 14.21: Average SIR costs for the 458 transmitter problem solved at 9dB using a domain of 11 consecutive frequencies. Results are obtained from the simulated annealing hybrid algorithm.

The results of this analysis show that using a constraint weight of 1 in the simulated annealing hybrid algorithm cost function produces the best result on average.

Analysis of the Binary Constraint Generation SIR Threshold

This section analyses the results from the 458 transmitter problem evaluated at an SIR threshold of 9dB to provide an understanding on the use of an appropriate SIR threshold when generating binary constraints.

Figure 14.16 presents a chart of the results obtained by solving the 458 transmitter problem using the simulated annealing binary constraint solving algorithm. The chart presents the results for domains of 10 and 11 consecutive frequencies and are obtained from table 14.7 and table 14.8. It is clear from

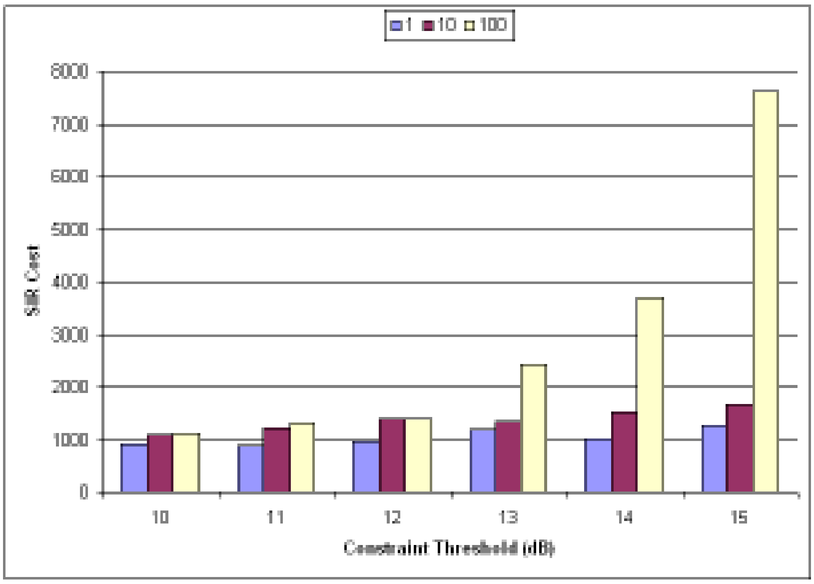

Figure 14.15: 458 transmitter problem evaluated at 9dB using a domain of 11 consecutive frequencies. Results show SIR costs for the hybrid algorithm using constraint weights of 1, 10 and 100.

this chart that binary constraints generated at thresholds of 13dB, 14dB and 15dB give poor results when compared to lower thresholds.

Figure 14.17 presents a chart of the results obtained by solving the 458 transmitter problem using the simulated annealing pure SIR algorithm from a binary constraint start. The results presented in figure 14.16 are used as starting assignments. Figure 14.17 presents the results for domains of 10 and 11 consecutive frequencies. It can be deduced from the chart when a domain of 10 consecutive frequencies is used the best result is obtained from the binary constraints generated at a threshold of 11dB. When a domain of 11 consecutive frequencies is used the best results are obtained from the binary constraints generated at thresholds of 11dB, 12dB and 13dB.

In summary, from the analysis completed in this section it can be deduced that the best results are obtained from binary constraints that are generated at thresholds of no more than 4dB above the evaluation SIR threshold.

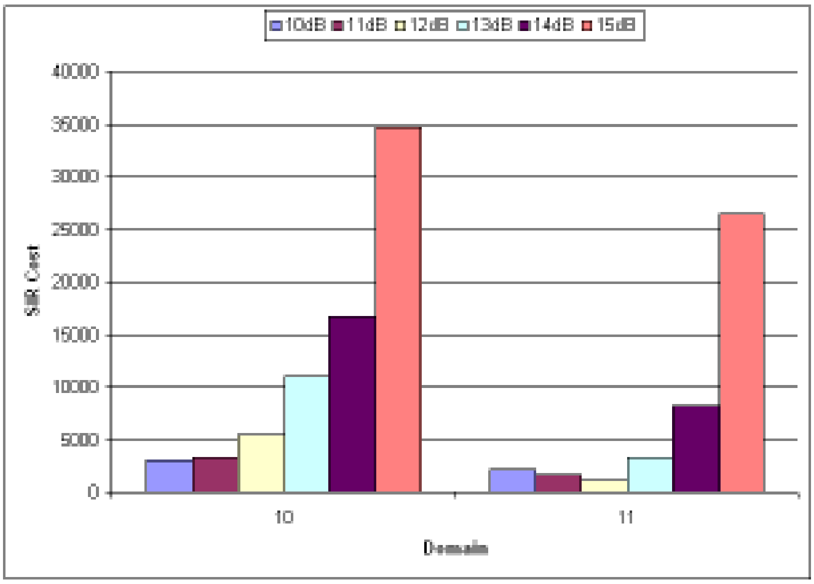

Figure 14.16: 458 transmitter problem evaluated at 9dB using domains of 10 and 11 consecutive frequencies. Results show SIR costs for the binary constraint solving simulated annealing algorithm algorithm. The binary constraints are generated at SIR thresholds of 10dB, 11dB, 12dB, 13dB, 14dB and 15dB.

Analysis of the SIR Algorithms

This section analyses the results from the 458 transmitter problem evaluated at an SIR threshold of 9dB to show which of the SIR algorithms outlined in section 8.7 produces the best results.

Figure 14.18 presents a chart of all of the best results obtained from the algorithms used to solve the 458 transmitter problem. The results are obtained from table 14.7 and table 14.8. The chart presents results for domains of 10 and 11 consecutive frequencies. It is clear from the chart that the simulated annealing pure SIR algorithm from a binary constraint start produces the best results in both cases. The simulated annealing binary constraint solving algorithm produced the poorest results.

14.4.4 458 Transmitter Problem, 15dB SIR Evaluation

This set of experiments were evaluated at an SIR threshold of 15dB. Experiments were run using domains of 24 and 25 consecutive frequencies. The results are presented in table 14.9 and table 14.10, with the best results for each method being highlighted in bold. In both cases the best overall result

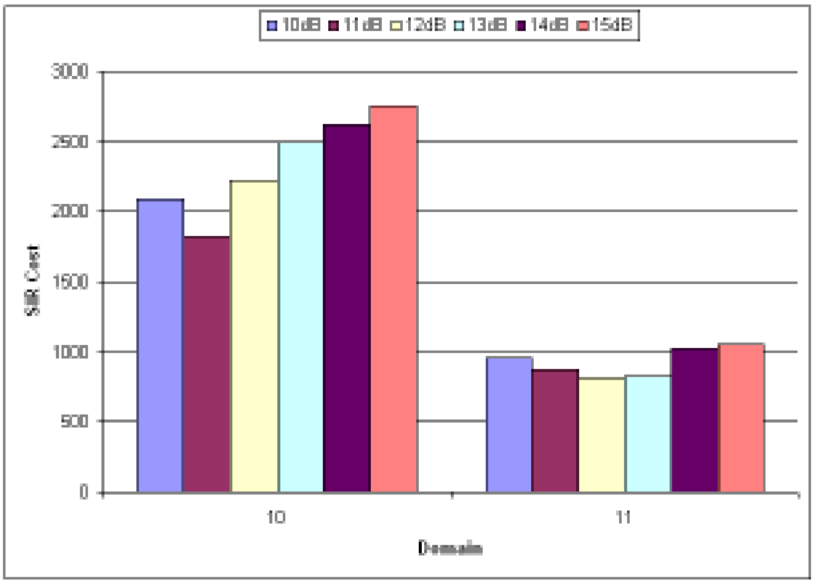

Figure 14.17: 458 transmitter problem evaluated at 9dB using domains of 10 and 11 consecutive frequencies. Results show SIR costs for the pure SIR simulated annealing algorithm algorithm from a binary constraint start. The binary constraints are generated at SIR thresholds of 10dB, 11dB, 12dB, 13dB, 14dB and 15dB.

by a substantial margin is obtained using the simulated annealing pure SIR algorithm with a binary constraint starting assignment.

Analysis of the Constraint Weight

The results from the 458 transmitter problem evaluated at an SIR threshold of 15dB can be used to provide understanding on the use of a weight on the binary constraint cost function component. The weight values that are being compared are 1, 10 and 100. The results are analysed for domains of 24 and 25 consecutive frequencies. The constraints were generated at SIR thresholds of 16dB, 17dB, 18dB, 19dB, 20dB, 21dB and 22dB.

Figure 14.19 presents a chart of the results obtained using the simulated annealing SIR hybrid algorithm on the 458 transmitter problem, using a domain of 24 consecutive frequencies. Using the results from table 14.9 the average SIR cost can be evaluated for each constraint weight and is presented in table 14.22. The averages show that using a constraint weight of 1 produced the best overall results while using a constraint weight of 100 produced the worst overall results.

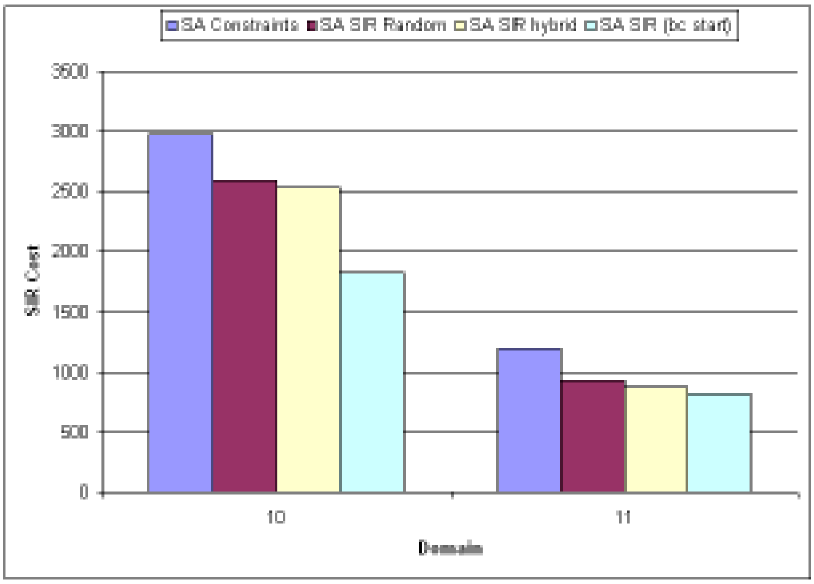

Figure 14.18: 458 transmitter problem evaluated at 9dB using domains of 10 and 11 consecutive frequencies. The chart gives the best results from all of the algorithms for each domain.

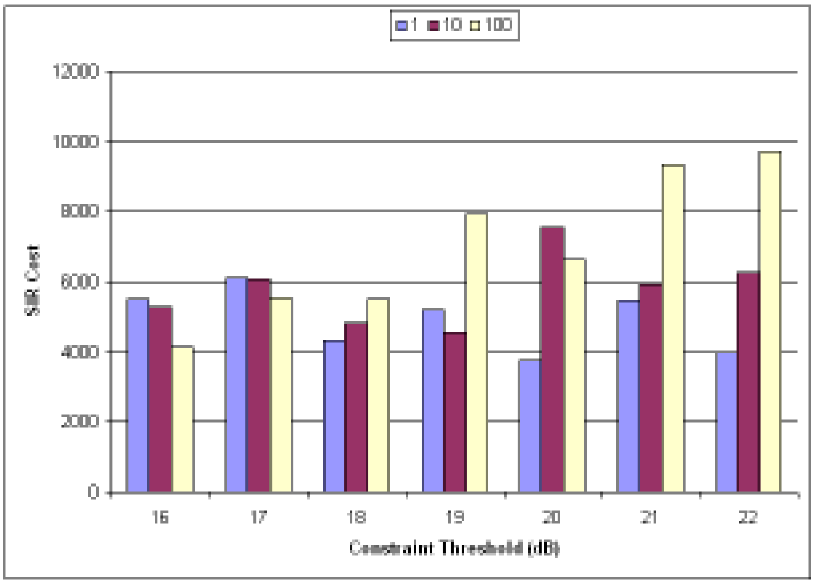

Figure 14.19: 458 transmitter problem evaluated at 15dB using a domain of 24 consecutive frequencies. Results show SIR costs for the hybrid algorithm using constraint weights of 1, 10 and 100.

Constraint weight	Average SIR Cost
1	4909.04
10	5778.95
100	6973.87

Table 14.22: Average SIR costs for the 458 transmitter problem solved at 15dB using a domain of 24 consecutive frequencies. Results are obtained from the simulated annealing hybrid algorithm.

Figure 14.20 presents a chart of the results obtained using the simulated annealing SIR hybrid algorithm on the 458 transmitter problem, using a domain of 25 consecutive frequencies. Using the results from table 14.10 the average SIR cost can be evaluated for each constraint weight and is presented in table 14.23. The averages show that using a constraint weight of 10 produced the best overall results while using a constraint weight of 100 produced the worst overall results.

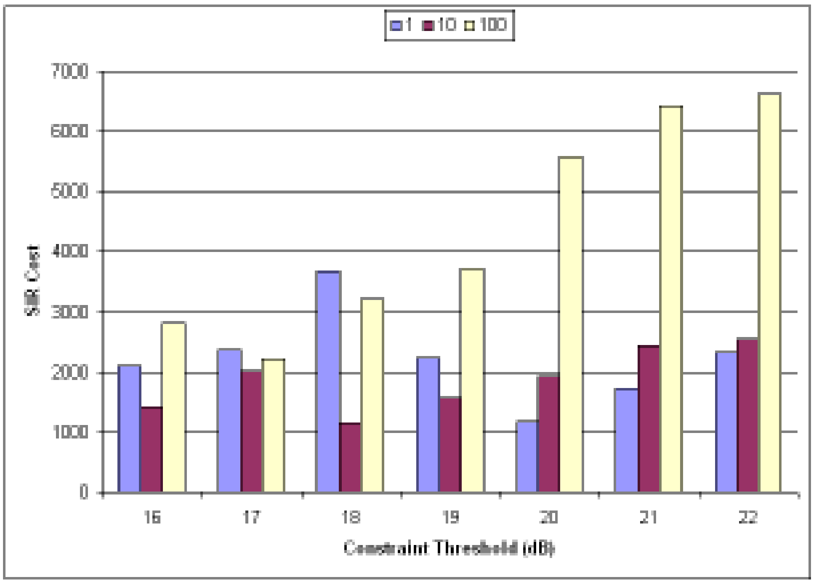

Figure 14.20: 458 transmitter problem evaluated at 15dB using a domain of 25 consecutive frequencies. Results show SIR costs for the hybrid algorithm using constraint weights of 1, 10 and 100.

In summary, the results have shown that a constraint weight of 100 always produced the worst results. While a constraint weight of 1 gives the best results the majority of the time, a constraint weight of 10 gives the best results sometimes. These results indicate that the simulated annealing algorithm produces worse results when the solving of the binary constraints is given much more priority over

Constraint weight	Average SIR Cost
1	2230.42
10	1869.51
100	4365.99

Table 14.23: Average SIR costs for the 458 transmitter problem solved at 15dB using a domain of 25 consecutive frequencies. Results are obtained from the simulated annealing hybrid algorithm.

the reduction of the SIR cost.

Analysis of the Binary Constraint Generation SIR Threshold

This section analyses the results from the 458 transmitter problem evaluated at an SIR threshold of 15dB. This aims to provide an understanding on the use of an appropriate SIR threshold when generating binary constraints, for domains of 24 and 25 consecutive frequencies, with the binary constraints generated at SIR thresholds of 16dB, 17dB, 18dB, 19dB, 20dB, 21dB and 22dB.

Figure 14.21 presents a chart of the results obtained by solving the 458 transmitter problem using the simulated annealing binary constraint solving algorithm. The chart presents the results for domains of 24 and 25 consecutive channels and are obtained from table 14.9 and table 14.10. It is clear from this chart that binary constraints generated at thresholds of 20dB, 21dB and 22dB always give poor results when compared to lower thresholds.

Figure 14.22 presents a chart of the results obtained by solving the 458 transmitter problem using the simulated annealing pure SIR algorithm from a binary constraint start. The results presented in figure 14.21 are used as starting assignments. Figure 14.22 presents the results for domains of 24 and 25 consecutive channels and are obtained from table 14.9 and table 14.10. It can be deduced from the chart that binary constraints generated at thresholds of 16dB, 20dB, 21dB and 22dB generally produce the poorest results when compared to thresholds of 17dB, 18dB and 19dB, with 19dB producing the best results in both cases.

In summary, the results have shown that binary constraints should be generated at a threshold of between 2dB and 4dB above the SIR evaluation threshold. When evaluating an assignment at an SIR threshold of 15dB the best results will be obtained using a set of binary constraints generated at thresholds of 17dB, 18dB or 19dB, with the larger values more appropriate to larger domains.

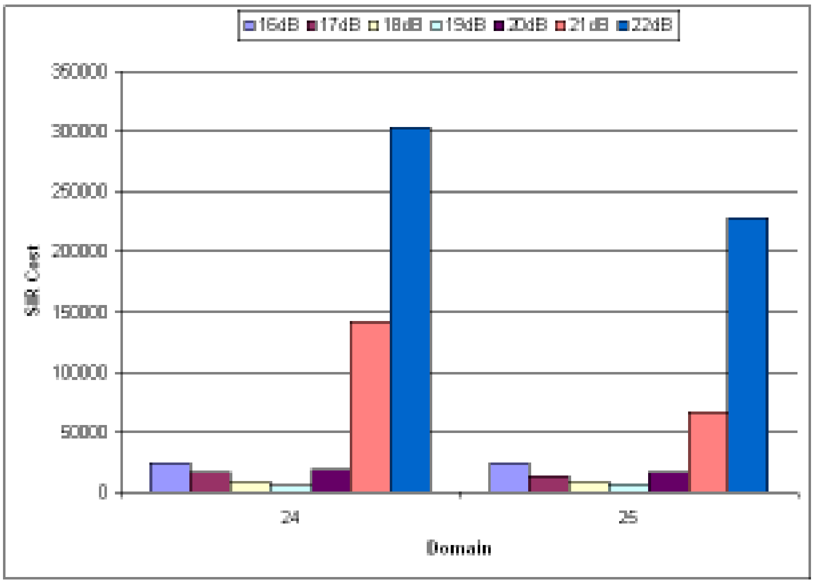

Figure 14.21: 458 transmitter problem evaluated at 15dB using domains of 24 and 25 consecutive frequencies. Results show SIR costs for the binary constraint solving simulated annealing algorithm algorithm. The binary constraints are generated at SIR thresholds of 16dB, 17dB, 18dB, 19dB, 20dB, 21dB and 22dB.

Analysis of the SIR Algorithms

This section analyses the results from the 458 transmitter problem evaluated at an SIR threshold of 15dB to show which of the SIR algorithms outlined in section 8.7 produces the best results.

Figure 14.23 presents a chart of all of the best results obtained from the algorithms used to solve the 458 transmitter problem. The results are obtained from table 14.9 and table 14.10. The chart presents results for domains of 24 and 25 consecutive frequencies. It is clear from the chart that the simulated annealing pure SIR algorithm from a binary constraint start produces considerably better results in both cases. The simulated annealing binary constraint solving algorithm and the simulated annealing pure SIR algorithm from a random start tend to produce the poorest results.

In summary, it is interesting to note that when binary constraints are solved without any SIR calculations they produce the poorest results. This is also true when using a pure SIR algorithm without binary constraints. But when the two methods are combined by initially solving the binary constraints and then using the solution as a starting point for a pure SIR algorithm, the best results are always

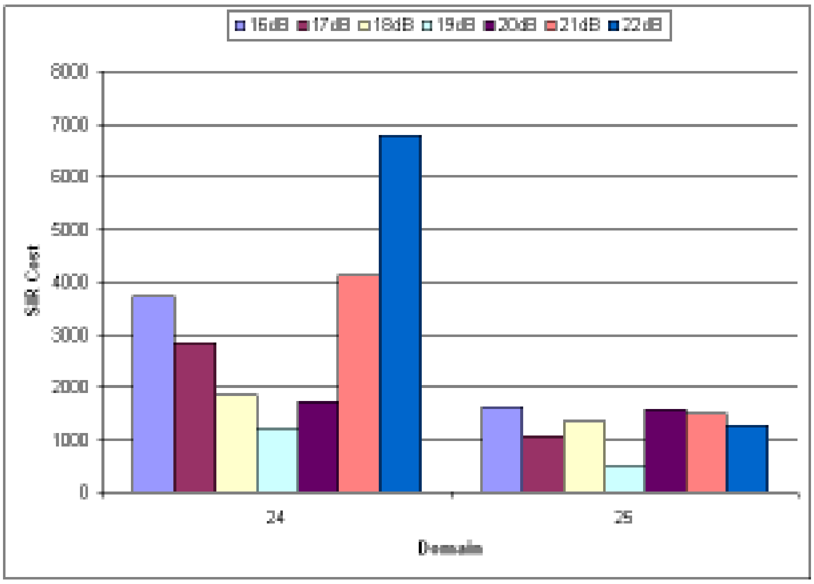

Figure 14.22: 458 transmitter problem evaluated at 15dB using domains of 24 and 25 consecutive frequencies. Results show SIR costs for the pure SIR simulated annealing algorithm algorithm from a binary constraint start. The binary constraints are generated at SIR thresholds of 16dB, 17dB, 18dB, 19dB, 20dB, 21dB and 22dB.

obtained.

14.4.5 Further Results for the 458 Transmitter Problem

Table 14.13 gives further results for the 458 transmitter problem that are presented in [21]. The results provide further evidence that the pure SIR simulated annealing algorithm for a binary constraint start gives the best solutions when compared to a binary constraint solving simulated annealing algorithm, a hybrid simulated annealing algorithm and a pure SIR simulated annealing algorithm from a random start. Runtimes are presented for this set of results and show the difference in time for obtaining solutions for binary constraint based problems and obtaining solutions for SIR problems. Although the runtime differences are particularly large, the results reflect the importance of the SIR calculations. When a binary constraint start is used, good results are obtained much earlier in the run.

The results presented in table 14.13 also provide further evidence that generating the binary constraints at an SIR threshold of 2-4dB higher than the evaluation SIR.

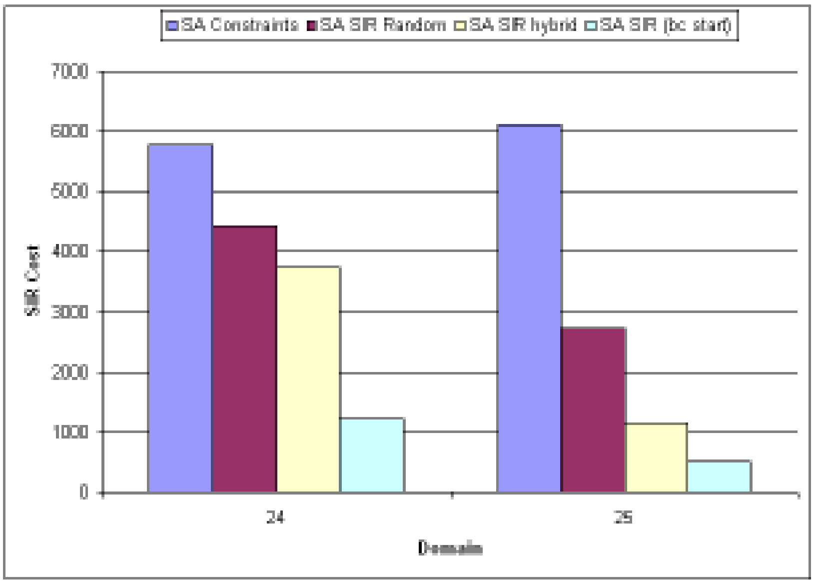

Figure 14.23: 458 transmitter problem evaluated at 15dB using domains of 24 and 25 consecutive frequencies. The chart gives the best results from all of the algorithms for each domain.

14.4.6 Radiocommunication Agency Terrain Database Computed Data

The set of novel terrain based data experiments were evaluated at an SIR threshold value of 15dB. An initial set of runs were performed using the plain SIR simulated annealing algorithm. These were done to establish viable domain sizes in order to obtain satisfactory results. Table 14.11 presents a set of domains with the corresponding start temperature, end temperature, SIR cost and network coverage.

The results presented in table 14.11 show that a large domain is required in order to obtain good results for this problem. The next set of experiments apply a program of algorithms on a particular domain. This will involve generating sets of binary constraints at numerous SIR thresholds above the SIR of 15dB used for evaluation. The algorithms encompass the binary constraint solving simulated annealing algorithm, hybrid simulated annealing algorithm and the plain SIR simulated annealing algorithm using a start file generated by a solution of the binary constraints.

The set of results presented in table 14.12 use a domain of 150 consecutive frequencies. The best results for each method are highlighted in bold. It is clear that the SA SIR (constraint start) algorithm produces the best results for this problem. The binary constraints generated at a threshold of 16dB produce the best results for the SA constraints, SA SIR hybrid and SA SIR (bc start) algorithms.

The SA SIR hybrid algorithm used a constraint weight of 10 which forces the algorithm to focus more on the solving of the binary constraints. This algorithm has the benefit that only one run is required, but it is clear that the separation of this algorithm produces better results for this problem. The pure SA SIR random start algorithm emphasises the need for binary constraints, while the SA constraints algorithm emphasises the need for the SIR. For a hard problem the solution of the binary constraints gives poor SIR results and the need for an SIR algorithm is essential.

14.4.7 Satellite Problem HEX358 Results

Results for the novel HEX358 problem are presented in table 14.14. The results provide further evidence that the pure SIR simulated annealing algorithm for a binary constraint start gives the best solutions when compared to a binary constraint solving simulated annealing algorithm, a hybrid simulated annealing algorithm and a pure SIR simulated annealing algorithm from a random start. Runtimes are presented for this set of results and show the difference in time for obtaining solutions for binary constraint based problems and obtaining solutions for SIR problems. When comparing the runtimes of the pure SIR simulated annealing algorithm from a binary constraint start to those for the SIR simulated annealing hybrid algorithm and the pure SIR simulated annealing algorithm from a random start it is clear that the SIR simulated annealing algorithm from a binary constraint start gives better results with faster runtimes.

The results presented in table 14.14 also provide further evidence that the binary constraints should be generated at an SIR threshold of 2-4dB higher than the evaluation SIR.

14.4.8 Satellite Problem HEX1794 Results

Results for the novel HEX1794 problem are presented in table 14.15. The results show that the pure SIR simulated annealing algorithm from a binary constraint start gives the best results. The best result is produced from a set of binary constraints generated at an SIR threshold of 20dB, which is 5dB above the evaluation SIR. Therefore in some cases it may be required to generate binary constraints at an SIR threshold of up to 5dB higher than the evaluation SIR. This may depend on the size and complexity of the problem and the amount of multiple interference anticipated.

14.5 SIR Runtime Experiments

This section compares the binary constraint solving simulated annealing algorithm to the pure SIR simulated annealing algorithm. A set of binary constraints are generated at 17dB from the 458 transmitter problem. The binary constraints are solved using the binary constraint solving simulated annealing algorithm. The SIR cost and the runtime of obtaining this solution is then recorded. The pure SIR simulated annealing algorithm is then run using the binary constraint solution as a starting assignment. The pure SIR simulated annealing algorithm is run three times, the first run is for 60 seconds, the second run is for 120 seconds and the final run is for 300 seconds. At the end of each run the SIR cost is recorded. The experiments were performed using a domain of 20 consecutive frequencies. The results are presented in table 14.24.

Algorithm	Moves per temp change	Start temp	Min temp	Cooling ratio	SIR Cost	Runtime (seconds)
BC SA	1500	1	10^{-15}	0.987	33985.83	25
SIR SA	1500	1	-	0.987	24524.69	60
SIR SA	1500	1	-	0.987	23718.40	120
SIR SA	1500	1	-	0.987	23674.07	300

Table 14.24: Results for the 458 transmitter problem using the binary constraint solving simulated annealing algorithm and the pure SIR simulated annealing algorithm using the binary constraint solution as a starting assignment. Experiments were performed on a Pentium 4 2.4GHz processor.

The results presented in table 14.24 show that an SIR algorithm can give better results than a binary constraint solving algorithm in short runtimes. A substantial cost improvement of the binary constraint solution is achieved in only 60 seconds by the pure SIR simulated annealing algorithm. The SIR results are unobtainable in such short runtimes without the binary constraint solution, but the SIR results could not be achieved by a binary constraint solving algorithm as the SIR cost can not be improved upon once all of the binary constraints are solved.

The impressiveness of the runtimes of these experiments can be seen when compared to the work performed in [25], [33] and [46], where it was found that SIR experiments can run for as long as 24 hours. Therefore SIR techniques can be extremely quick and effective when combined with binary constraints.

14.6 Frequency Hopping Experiments

The frequency hopping experiments are performed on the 458 transmitter problem with varying numbers of narrow band carriers being replaced by unsynchronised hopping carriers and groups of synchronised hopping carriers. The unsynchronised hopping carriers and groups of synchronised hopping carriers also have varying list length and contiguous block requirements. The objectives of these experiments are to show that the inclusion of frequency hopping carriers improve network performance when compared to the non-hopping case and that the list length cost and the contiguous cost work to improve the quality of the hopping lists as the algorithm progresses. The SIR simulated annealing algorithm will be used to solve the problems presented here.

The experiments will use an SIR threshold of 15dB and a domain of 25 consecutive frequencies. The results will then be compared to the non-hopping case presented in table 14.10. The results for the frequency hopping experiments are presented in table 14.25.

Unsynch Hoppers	Synch Groups	Start Temp	Iterations per Temp	Cool Ratio	Stop Temp	SIR Cost	List Length Cost	Contiguous Cost
0	0	1	1500	0.987	10^{-18}	2717.77	-	-
20	0	1	2000	0.999	0.01	484.33	1	1
20	0	2	2000	0.999	0.1	804.97	0	2
20	3	1	2000	0.999	0.01	189034.39	1240	0
20	3	2	2000	0.999	0.1	189034.39	1240	0
20	5	1	2000	0.999	0.01	6815.66	159	7
20	5	2	2000	0.999	0.1	5890.43	149	14
50	0	1	2000	0.999	0.01	0.00	0	0
50	3	1	2000	0.999	0.01	3617.62	742	26
50	3	2	2000	0.999	0.1	1566.42	434	36
50	5	1	2000	0.999	0.01	15303.22	547	3
50	5	2	2000	0.999	0.1	15146.17	555	3
80	0	1	2000	0.999	0.01	0.00	0	0
80	3	1	2000	0.999	0.01	1504.60	555	48
80	3	2	2000	0.999	0.1	4363.47	1265	34
80	5	1	2000	0.999	0.01	53407.46	851	2
80	5	2	2000	0.999	0.1	53639.52	851	2

Table 14.25: Results for the 458 transmitter problem evaluated at an SIR threshold of 15dB using a domain of 25 consecutive frequencies with the inclusion of unsynchronised hopping carriers and groups of synchronised hopping carriers.

The results show that network performance is greatly improved when large numbers of narrow band carriers are replaced by unsynchronised hopping carriers. These results also show that the majority, or in some cases all, of the list length and contiguous block requirements are met. The experiments that involve groups of synchronised hopping carrier give much poorer results than for the experiments that involve only unsynchronised hopping carriers. An attempt to improve these results using a higher start temperature was unsuccessful. In another attempt to successfully improve the results the weight X_3 (see section 11.2) is given a value of 2500. This value increases the probability of a cost increasing move of a synchronised hopping carrier being accepted. Table 14.26 presents the results of these further experiments.

Unsynch Hoppers	Synch Groups	Start Temp	Iterations per Temp	Cool Ratio	Stop Temp	SIR Cost	List Length Cost	Contiguous Cost
20	3	2	2000	0.987	0.1	5147.64	419	41
20	5	2	2000	0.987	0.1	13193.50	175	25
50	3	2	2000	0.987	0.1	2578.17	446	52
50	5	2	2000	0.987	0.1	12269.6	448	48
80	3	2	2000	0.987	0.1	2330.55	447	65
80	5	2	2000	0.987	0.1	12454.2	386	60

Table 14.26: Results for the 458 transmitter problem evaluated at an SIR threshold of 15dB using a domain of 25 consecutive frequencies with the inclusion of unsynchronised hopping carriers and groups of synchronised hopping carriers. A value of 2500 was used for X_3

The further results presented in table 14.26 show that the increased probability of cost increasing synchronised hopping carrier moves being accepted yields some significant improvements for some of the results, while others do not yield improvements. While unsynchronised hopping carriers do reduce SIR cost, care must be taken if unsynchronised hopping carriers are to be used.

To show how the SIR simulated annealing algorithm works to improve network performance, hopping list length costs and hopping list contiguous block costs, the costs can be recorded each time the frequency assignment is improved. This cost data can then be plotted to show how the costs change as the algorithm progresses. The cost data for this analysis is taken from the experiment involving 80 unsynchronised hoppers and 0 synchronised groups, given in table 14.25.

Figure 14.24, figure 14.25 and figure 14.26 provide a good insight into the behaviour of the SIR simulated annealing algorithm when it is assigning lists of frequencies to hopping carriers. Figure 14.24

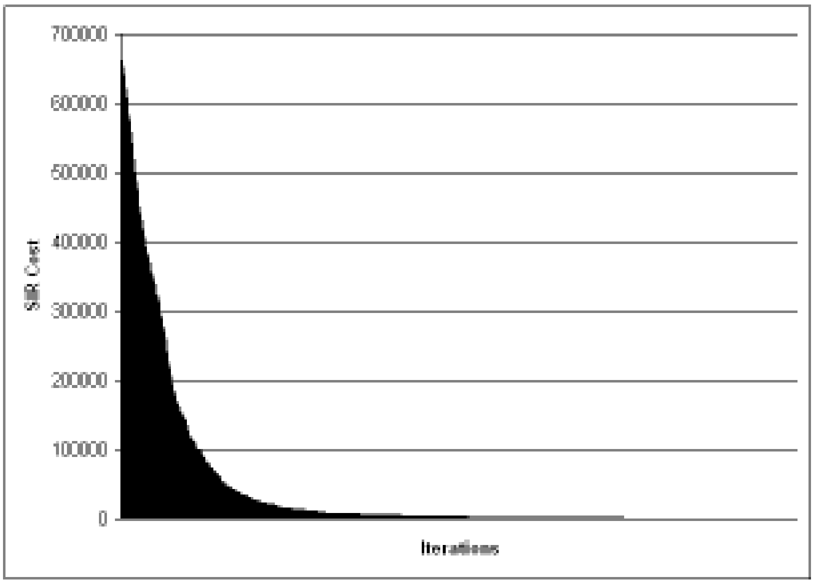

Figure 14.24: Graph showing the reduction in SIR cost on the 458 transmitter problem containing 80 unsynchronised frequency hopping carriers.

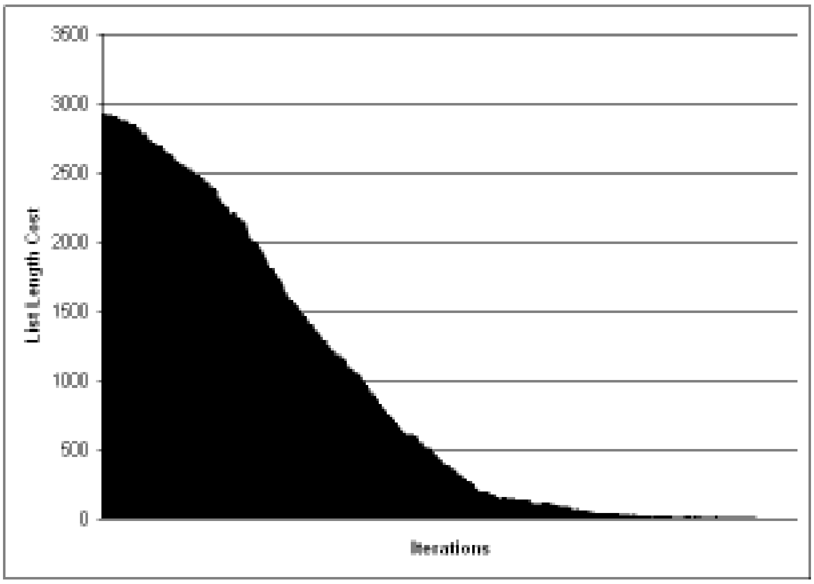

Figure 14.25: Graph showing the performance of the list length cost on the 458 transmitter problem containing 80 unsynchronised frequency hopping carriers.

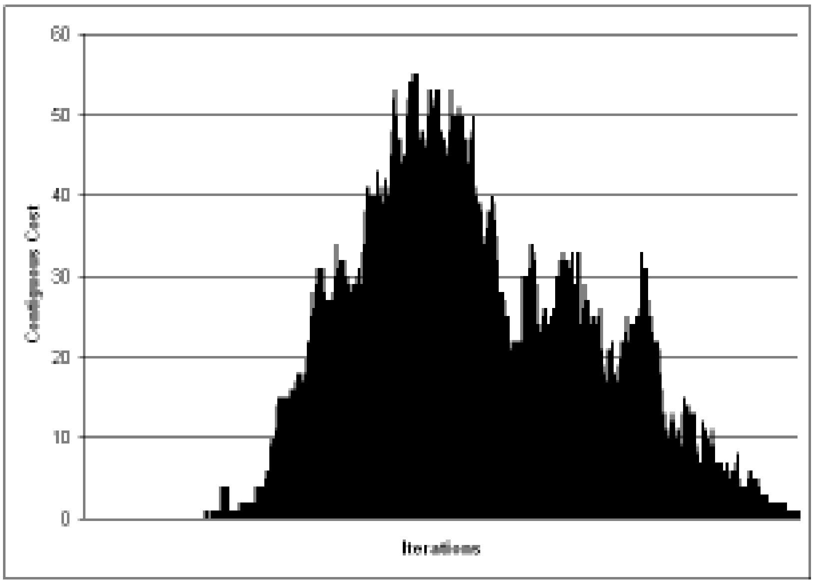

Figure 14.26: Graph showing the performance of the contiguous block cost on the 458 transmitter problem containing 80 unsynchronised frequency hopping carriers.

and figure 14.25 show a consistent improvement in the SIR cost and the list length cost. The SIR cost is initially very high due to the random start, while the list length cost is predictably initially high because all of the hopping carriers begin with a frequency list length of 1. As the algorithm progresses the list lengths will increase, hence improving the SIR cost and the list length cost. Figure 14.26 displays the contiguous block costs, which are initially at zero because each hopping carrier's frequency list consists of 1 contiguous block, i.e. 1 frequency. This cost tends to increase as the frequency list lengths increase, but this cost is then reduced as the frequency lists are optimised by the algorithm. Observation of the general behaviour of the algorithm shows that the SIR cost tends to be considerably reduced initially, the algorithm then improves the list length costs while still improving the SIR cost, and then finally the hopping lists are optimised by improving the contiguous block cost.

Several results have been published for GSM frequency hopping experiments, for example [5], [6] and [35]. The results presented here use a combination of GSM frequency hopping techniques and military frequency hopping requirements. Hence, it is difficult to compare any previously published results.

14.7 Multiple Carrier Type SIR Experiments

The final set of experiments, presented in this section use the 458 transmitter problem with multiple carrier types included. In a similar way to the experiments performed in section 14.1, runtimes will be compared when adding further carrier types to the 458 transmitter problem. The SIR simulated annealing algorithm will be used to solve the problems presented here. The multiple carrier types consist of up to 8 wide band carriers (of which some may be assumed to be stacks of CDMA carriers) presented in figure 14.27, up to 8 Aggregated Channel Carriers (AGCC) presented in figure 14.1 and up to 8 synchronised hopping carriers given in figure 14.28.

Carrier No.	Transmitter Identity	Bandwidth (channels)	Receiver Filter Bandwidth (channels)	β
1	31	5	5	0
2	32	7	5	0
3	33	9	5	0
4	34	5	5	0
5	35	11	5	0
6	36	15	5	0
7	37	17	5	0
8	38	5	5	0

Figure 14.27: Wide band carriers. β values are used in spectral overlap function 2 (refer to section 9.3).

Carrier No.	Transmitter Identity	Desired List Length	Contiguous Block Allowance	Synchronised Group
1	41	10	4	1
2	42	10	4	1
3	43	12	3	2
4	44	12	3	2
5	45	14	3	3
6	46	14	3	3
7	47	14	3	3
8	48	14	3	3

Figure 14.28: Frequency hopping carriers; for further details refer to section 10 and section 11.

The experiments contain combinations of the carrier types. The experiments use the simulated annealing algorithm parameters that were presented section 14.1 and are evaluated at an SIR of 15dB using a domain of 25 consecutive frequencies. The results are presented in table 14.27.

Experiment Rank	Problem	AGCC Groups	Wide Band Carriers	Hopping Carriers	SIR Cost	Runtime (seconds)
1	458 Transmitter	-	-	-	2717.77	926
2	458 Transmitter	-	8	-	9939.98	990
3	458 Transmitter	2	2	2	6834.69	1146
4	458 Transmitter	-	8	8	64526.52	1223
5	458 Transmitter	-	-	8	28255.43	1228
6	458 Transmitter	4	4	4	13786.88	1250
7	458 Transmitter	8	8	-	9942.81	1275
8	458 Transmitter	8	-	-	2770.18	1293
9	458 Transmitter	8	-	8	14068.14	1514
10	458 Transmitter	8	8	8	40145.38	1546

Table 14.27: Results for the 458 transmitter problem with multiple carrier types, comparing runtimes when adding AGCC groups, wide band carriers and hopping carriers. Experiments were performed on a Pentium 4 2.4GHz processor. The results are ranked in increasing order of runtimes.

The SIR costs presented in table 14.27 are displayed to reflect the increasing complexity of the problems. The SIR costs could be reduced by possibly restricting frequencies to certain transmitters, but the objective of these experiments is to compare runtimes only. The results are ranked in increasing order of runtimes, this allows for an analysis of which carrier types cause the most significant impacts on performance. It is clear from the results that the wide band carrier types cause the least significant loss in performance, while it is obvious that the introduction of AGCC groups and synchronised hopping groups cause significant but realistic losses in performance of the algorithm. The reason for the losses in performance when including AGCC groups and synchronised hopping groups is that a change in frequency for either group consists of a change in frequency for every transmitter in the group. The loss in performance is unavoidable but is not unacceptably large.

Experiments of this complexity have not previously been presented in the literature on FAPs. Therefore the results can not be compared to any others presented in the past.

Chapter 15

Conclusion

The aims of this research were to define a common formulation for military frequency assignment problems, encompassing frequency planning in all areas of military frequency assignment (land, air, maritime and satellite).

A generic model has been designed and implemented in software. The model incorporates a weighted linear cost function consisting of binary constraint violations, signal-to-interference (SIR) based cost, frequency hopping list costs, spurious emissions, spurious responses, and intermodulation products.

The algorithms implemented are a hill climbing algorithm and a simulated annealing algorithm. The algorithms can use any linear combination of the first three components of the cost function as listed above, and could optionally include the last three.

Both the generic model and the software implemented allow a propagation model to be applied either internally or externally. The model applied internally (for example purposes) are d^{-4} for terrestrial problems and $|sin(x)/x|^2$ for satellite problems. Any propagation model could be used if applied externally, via the intermediate file.

Multiple carrier types have been considered. These multiple carrier types consist of narrow band homogeneous carrier types, aggregated channel carrier types, wide band heterogeneous carrier types and CDMA carrier types, unsynchronised frequency hopping carrier types and synchronised frequency hopping carrier types. Valid frequencies can be assigned to narrow band homogeneous carrier types, wide band heterogeneous carrier types and CDMA carrier types and aggregated channel carrier types. Valid lists of frequencies can be assigned to individual unsynchronised hopping carriers and groups of synchronised hopping carriers.

Experiments have been performed on a selection of binary constraint problems, consisting of the 95 transmitter problem, the 458 transmitter problem and the satellite HEX1794 problem. The results presented have shown that the software created for this research has comparable performance to established frequency assignment problem solving software. This is not a novel approach, but it is important to demonstrate that binary constraints can be solved quickly and efficiently. Further experiments on binary constraint problems included aggregated channel carrier types and wide band carrier types. The software shows excellent performance when solving binary constraint problems involving multiple carrier types. This is true for all problems attempted.

Extensive experiments have been performed on various problems to evaluate the capabilities of the SIR based techniques developed in this research. The problems consisted of the 95 transmitter problem, the 458 transmitter problem, a Radiocommunication agency terrain database computed data 458 transmitter problem, the satellite HEX358 problem and the satellite HEX1794 problem. The algorithms tested consisted of the pure SIR simulated annealing algorithm from a random start, the binary constraint solving simulated annealing algorithm, the hybrid simulated annealing algorithm and the pure SIR simulated annealing algorithm from a binary constraint start. The objectives of the experiments were to show which of the algorithms gives the best results, to evaluate the ideal threshold for generating binary constraints and to determine the importance of weighting the sum of the binary constraint violations in a hybrid algorithm.

The results showed that in every case the pure SIR simulated annealing algorithm from a binary constraint start out-performed the other algorithms substantially. The binary constraint solutions were generally unsatisfactory when compared with solutions obtained by the other algorithms.

The ideal threshold for generating the binary constraints was somewhat variable. However a general rule can be derived; binary constraints should be generated at a threshold between 2dB and 4dB above the evaluation SIR threshold, with the larger values appropriate when more multiple interference is anticipated.

It is clear that the pure SIR simulated annealing algorithm from a binary constraint start outperforms the simulated annealing hybrid algorithm. However, the hybrid algorithm does allow for variable weighting of the binary constraint violations and the SIR cost. It was initially unclear whether or not giving additional weight to the sum of the binary constraint violations would give better results when compared to the equally weighted case. The results have shown that equally weighted hybrid cost function (i.e. weights 1 and 1) gives the best results in most cases, although in some cases a weight of 10 applied to the binary constraint violations (i.e. weights 10 and 1) can give good results. It is clear that a weight no larger than 10 should be applied to the sum of the binary constraint violations in a hybrid algorithm.

SIR cost based algorithms may appear to be slow when compared with binary constraint solving algorithms. A set of experiments were performed to show that using the pure SIR simulated annealing algorithm from a binary constraint start for short periods of time can substantially improve results. In particular applying an SIR based cost algorithm for only 60 seconds on an assignment produced from a binary constraint solution gives considerable cost improvements. If an SIR based cost algorithm is run from a random start good results take significantly longer to achieve.

A set of experiments were performed on the 458 transmitter problem which had various numbers of narrow band carriers replaced by unsynchronised frequency hopping carriers and groups of synchronised frequency hopping carriers. The main objective of the experiments was to show that the inclusion of frequency hopping carriers improves network performance when compared to the non-hopping case. The results showed that this was true for many cases, while in some cases the problems were difficult to solve due to the large numbers of groups of synchronised hopping carriers, however a technique was presented that reduced the costs of these difficult problems. A secondary objective was to show that the algorithm improved the quality of the frequency hopping lists. This is done using a list length cost and a contiguous block cost. The results have shown that these cost components are extremely important in improving the quality of frequency hopping lists and that they perform well.

The final set of experiments attempted to solve the 458 transmitter problem involving combinations of multiple carrier types using the pure SIR simulated annealing algorithm from a random start. The objective of this set of experiments was to evaluate the loss in performance with the added complexity of the multiple carrier types. The results showed a controlled decrease in performance as the problem became more complex. It can be concluded that the assignment of frequencies to wide band carriers and CDMA carriers have little affect on the performance, while the assignment of frequencies to aggregated channel carrier groups and the assignment of frequency lists to groups of synchronised hopping carriers have a significant effect on the performance. The overall performance is realistic and acceptable, demonstrating that the model of the formulation can be implemented with little loss of performance.

In conclusion, this research has developed a generic model encompassing all areas of military frequency assignment. A novel approach to SIR techniques has been demonstrated that combines the use of binary constraints with SIR techniques. This approach has proved to be more effective than the individual approaches. A system has been implemented that is suitable for large military problems. The performance of the system ranges from excellent to satisfactory. With the constant advances in modern day computing it is clear that SIR calculations will become the standard formulation for frequency assignment problems, although the research presented here shows that binary constraints are still needed.

Bibliography

[1] Aardal K.I., van Hoesel S.P.M., Koster A.M.C.A., Mannino C., Sassano A., *Models and solution techniques for frequency assignment problems,* ZIB-Report 01-40, Berlin, December 2001.

[2] Aftergood S., May 2000, [online], *FAS (Federation of American Scientists) - Intelligence Programs and Systems*, Available from: http://www.fas.org/irp/program/list.htm [Accessed August 2004].

[3] Allen S.M., Smith D.H., Hurley S., *Lower Bounding Techniques for Frequency Assignment,* Discrete Mathematics, Nos. 197/198, pp. 41-52, 1999.

[4] Allen S.M., Smith D.H., Hurley S., *Generation of Lower Bounds for Minimum Span Frequency Assignment,* Discrete Applied Mathematics, Vol. 119, Nos. 1-2, pp. 59-78, June 2002.

[5] Björklund P., Värbrand P., Yuan D., *Optimal GSM Network Planning with Frequency Hopping,* presented at 5th INFORMS Telecommunication Conference, Boca Raton, USA, March 2000.

[6] Björklund P., Värbrand P., Yuan D., *Optimal Frequency Planning in Mobile Networks with Frequency Hopping,* preprint, 2002.

[7] Bouju A., Boyce J.F., Dimitropoulos C.H.D., vom Scheidt G., Taylor J.G., *Tabu search for radio links frequency assignment problem,* in Conference on Applied Decision Technologies: Modern Heuristic Methods, Brunel University, Uxbridge, England, pp. 233-250, 1995.

[8] Cabon B., de Givry S., Lobjois L., Schiex T., Warners J.P., *Radio Link Frequency Assignment,* Constraints, Vol. 4, pp. 79-89, 1999.

[9] Capone A., Trubian M., *Channel assignment problem in cellular systems: a new model and tabu search algorithm,* IEEE Transaction on Vehicular Technology, VT-48 No. 4, pp. 1252-1260, July 1999.

[10] Castellino D.J., Hurley S., Stephens N.M., *A tabu search algorithm for frequency assignment,* Annals of Operations Research, Vol. 63, pp. 301-319, 1996.

[11] Correia (ed.) L.M., *Wireless Flexible Personalised Communications;* COST 259: European Co-operation in Mobile Radio Research, Chichester, England: John Wiley & Sons Ltd., 2001.

[12] Dorne R., Hao J.-K., *Tabu search for graph coloring, T-colorings and set T-colorings,* in Meta-Heuristics: Advances and Trends in Local Search Paradigms for Optimization, Voss A., Martellow S., Osman I.H., Roucairol C., Eds. Norwell, MA: Kluwer, pp. 77-92, 1999.

[13] Dueck G., Scheuer T., *Threshold Accepting: a general purpose optimization algorithm appearing superior to simulated annealing,* Journal of Computational Physics, Vol. 90, No. 1, pp. 161-175, 1990.

[14] Dunkin N., Allen S.M., Smith D.H., Hurley S., *Frequency assignment problems: Benchmarks and Lower Bounds,* Technical Report UG-M-98-1, University of Glamorgan, 1998.

[15] Dunkin N.W., Jeavons P.G., *Expressiveness of binary constraints for the frequency assignment problem,* DIAL-M Workshop, 3rd Annual ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM'97), Budapest, October 1997.

[16] Duque-Anton M., Kunz D., Ruber B., *Channel assignment for cellular radio using simulated annealing,* IEEE Transaction on Vehicular Technology, Vol. 42, pp. 14-21, February 1993.

[17] EADS SPACE Services, No date, [online], *Military Communications*, Available from: http://www.space.eads.net/web1/home/index.asp [Accessed August 2004].

[18] Eisenblätter A., Koster A., June 2000, [online], *FAP web - A website about Frequency Assignment Problems*, Available from: http://fap.zib.de/ [Accessed November 2004].

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

[20] Gamst A., Rave W., *On frequency assignment in mobile automatic telephone systems,* Proceedings of GLOBECOM'82, IEEE, pp. 309-315, 1982.

[21] Graham J.S., Montemanni R., Moon J.N.J., Smith D.H., *Frequency Assignment, Multiple Interference and Binary Constraints,* submitted for publication.

[22] Hale W.K., *Frequency Assignment: Theory and Applications,* in Proceedings of IEEE, Vol. 68, No. 12, pp. 1497-1514, December 1980.

[23] Hale W.K., *New spectrum management tools,* in Proceedings of IEEE International Symposium on Electromagnetic Compatibility, IEEE Press, Piscataway, N.J., pp. 47-53, 1981.

[24] Hao J.-K., Dorne R., Galinier P., *Tabu search for frequency assignment in mobile radio networks,* Journal of Heuristics, Vol. 4, pp. 47-62, 1998.

[25] Hurley S., Whitaker R.M., Smith D.H., *Channel assignment in cellular networks without channel separation constraints,* in Proceedings of IEEE Vehicular Technology Conference, pp. 1714-1718, Fall 2000.

[26] Hurley S., Smith D.H., Thiel S.U., *FASoft: A system for discrete channel frequency assignment,* Radio Science, Vol. 32, No. 5, pp. 1921-1939, 1997.

[27] Kirkpatrick S., Gelatt C., Vecchi M., *Optimization by simulated annealing,* Science. 220(4598), pp. 671-680, 1983.

[28] Metropolis N., Rosenbluth A., Rosenbluth M., Teller A., Teller E., *Equations of state calculations by fast computing machines,* Journal of Chemical Physics, Vol. 21, No. 6, pp. 1087-1092, 1953.

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

[30] MOD (Ministry of Defence), No date, [online], *Defence Procurement Agency - Integrated Project Teams*, Available from: http://www.mod.uk/dpa/ipt/index.html [Accessed August 2004].

[31] Montemanni R., Moon J.N.J., Smith D.H., *An improved tabu search algorithm for the fixed spectrum frequency assignment problem,* IEEE Transaction on Vehicular Technology, Vol. 52, No. 4, pp. 891-901, July 2003. (ISSN 0018-9545).

[32] Montemanni R., Smith D.H., Allen S.M., *Lower bounds for fixed spectrum frequency assignment,* Annals of Operations Research, Vol. 107, pp. 237-250, 2001.

[33] Montemanni R., Smith D.H., Allen S.M., *An ANTS algorithm for the minimum-span frequency assignment problem with multiple interference,* IEEE Transactions on Vehicular Technology VT-51(5), pp. 949-953, September 2002.

[34] Montemanni R., Smith D.H., Allen S.M., *An improved Algorithm to Determine Lower Bounds for the Fixed Spectrum Frequency Assignment Problem,* European Journal of Operational Research, 156, pp. 736-751, August 2004.

[35] Moon J.N.J., Hughes L.A., Smith D.H., *Assignment of Frequency Lists in Frequency Hopping Networks,* IEEE Transactions on Vehicular Technology, May, 2005, to appear.

[36] Pender T., *UML Bible,* 1st edition, Wiley, June, 2003.

[37] Proakis J., *Digital Communications,* 4th Edition, McGraw Hill, 2001.

[38] Pujante A., de Haro L., *Optimisation of satellite frequency plans with balanced carriers by simulated annealing,* Electronic Letters, Vol. 33, No. 11, pp. 934-935, May, 1997.

[39] Smith D.H., Allen S.M., Hurley S., *Characteristics of good meta-heuristic algorithms for the frequency assignment problem,* Annals of Operations Research, Vol. 107, pp. 285-301, 2001.

[40] Smith D.H., Hurley S., Thiel S.U., *Improving heuristics for the frequency assignment problem,* European Journal of Operational Research, Vol. 107, No. 1, pp. 76-86, 1988.

[41] Smith D.H., Taplin R.K., Hurley S., *Frequency Assignment with Complex Co-Site Constraints,* IEEE Transactions on Electromagnetic Compatibility, Vol. 43, No. 2, pp. 210-218, May 2001.

[42] Smith G.D., Kapsalis A., Rayward-Smith V.J., Kolen A., *CALMA Radio Link Frequency Assignment Project: Technical Annexe,* EUCLID Project Report, UEA, Norwich, T-2.1:A - Supplementary Report, 1995.

[43] The British Army Vehicles & Equipment, No date, [online], *The Electronic Battlefield,* Available from: http://www.army.mod.uk/equipment/index.html [Accessed August 2004].

[44] Tiourine S.R., Hurkens C.A.J., Lenstra J.K., *An overview of Algorithmic Approaches to Frequency Assignment Problems,* Technical Report, Eindhoven University, The Netherlands, August 1995.

[45] Toftegaard Nielsen T., Wigard J., *Performance Enhancements in a Frequency Hopping GSM Network,* Boston, Kluwer Academic Publishers, 2000.

[46] Watkins W.J., Hurley S., Smith D.H., *Evaluation of Models for Area Coverage,* Radiocommunications Agency Agreement, RCCM070, Final Report, December, 1998.

[47] Westbrook L.D., Calledene T., Berry I., Briscombe N., *Carrier Frequency Allocation on FDMA Military Satcoms,* Proceedings of IEEE Milcom 2001: Communications for Network Centric Operations, McLean VA, October 2001.