Frequency Assignment Algorithms

Radiocommunications Agency Agreement
Ref. RCCM 070

Final Report Year 2

1996/97

S.U. Thiel and S. Hurley
Dept. of Computer Science
University of Wales Cardiff
PO Box 916
Cardiff CF2 3XF

D.H. Smith
Div. of Mathematics and Computing
University of Glamorgan
Pontypridd
CF37 1DL

April 21, 1997

Executive Summary

Frequency Assignment Algorithms

Radiocommunications Agency Agreement RCCM 070

Year 2, 1996/97

The research contract consisted of an investigation into the use of combinatorial optimisation techniques for discrete channel assignment in irregular networks, and the development of lower bounds to assess algorithm performance and to assist in the actual assignment process. The project was funded by the Radiocommunications Agency for the period 1^{st} April 1996 to 31^{st} March 1997 and builds on the work done in Year 1 (April 1995 to March 1996).

The work carried out during Year 2 extends the research of the first year. In particular, the system FASOFT has developed into a state-of-the-art channel assignment system. It contains solution techniques based on meta-heuristics including simulated annealing, tabu search and genetic algorithms. The system also includes sequential (greedy) assignment methods and exact algorithms. The most successful techniques, for both minimum span and fixed spectrum frequency assignment use hybrid sequential/simulated annealing and sequential/tabu search algorithms. When used in conjunction with the assignment of a critical subgraph (often identified by the lower bounding techniques) the techniques produce optimal frequency assignment plans for several benchmark problems which have previously remained unsolved for many years. The system uses a constraint matrix or channel separation matrix to model the network interference.

A Windows interface has been developed for FASOFT using Delphi 2.0. This should allow the system to be more accessible than the previous command line version, particularly to Radio Engineers.

Initial empirical research has been started on the classification of problems and on how to assess the spectrum efficiency of any assignment.

The project has proved very successful and future work will proceed on two fronts. The first will look at the development of heuristic algorithms for clique detection in the constraint graph. This will help in the identification of *critical subgraphs* which, in turn, will help in the classification of problems. Secondly, the problem of *area coverage* in a geographical region will be investigated.

Contents

1 Introduction

The research agreement *Study of Problems Comparing Various Frequency Assignment Algorithms* started in 1st April 1995 and currently continues until 31st March 1997. The project involves the development of methods for the assignment of radio frequencies and the development of a software system to implement these techniques. The investigation is a collaboration between the Universities of Cardiff and Glamorgan and is being funded by the Radiocommunications Agency under agreement RCCM 070.

1.1 The Frequency Assignment Problem

The radio frequency assignment problem concerns the allocation of frequencies to transmitters, with the aim of avoiding or minimising interference. The radio spectrum is an important natural resource. The increasing demands on the radio spectrum due to modern communication needs are outpacing the expansion of the useable spectrum due to technological advances. Many services, both civil and military, require access to the spectrum to function, and the demand for frequencies is increasing year by year. Consequently, efficient management of the spectrum requires that frequencies be assigned, for a particular service, in an optimum or near-optimum manner. The problem is thus of vital importance, but is known to be computationally intractable. In practice, this assignment of frequencies is often done using greedy heuristics which mimic the way the process may be done manually. These methods produce assignments quickly (even for large problems) but inevitably use more spectrum than is necessary. Exact methods based on graph theory give the optimum solution but are computationally unrealistic for problems involving, approximately, fifty transmitters or more.

It is important that methods be found for assessing the performance of these heuristic techniques. Simply comparing one heuristic with another gives no information about the actual quality of the assignments obtained using the heuristics. One approach is the use of standard lattice assignments [12, 30, 29], when the optimal assignment may be known. Another approach is the use of lower bounding techniques [13, 41]. These bounds are sometimes weak, but for frequency assignment problems typical of those that arise in practice, good bounds can often be obtained.

For many lower bounding methods the best lower bound is found not from the complete problem, but by picking out a subproblem of the complete problem. We shall formulate the radio frequency assignment problem as a problem in a constraint graph and identify the subproblem with a subgraph of the constraint graph. The basic idea that we shall explore is that the subgraph which gives the best available lower bound should be assigned first, and the assignment can then be extended to the complete problem. We will describe how this approach can be achieved and describe results which give both significant improvements to the best obtained from any heuristic directly, and in some cases optimal results.

1.2 Interference and the Constraint Matrix

In order to model interference, constraints are imposed on the assignment. When a transmitter can cause interference with a receiver, a constraint is implied on the frequencies assigned to this transmitter and the transmitter serving the receiver. Interference can occur when the assigned frequencies are the same or close together. This can happen when equipment is at the same location or within a few tens of metres of each other *(co-site interference)*, or when equipment is at a distance of several kilometres or more *(far-site interference)*.

Co-channel constraints: This is the most important factor in the consideration of far-site interference. A pair of transmitters located at different sites must not be assigned the same frequency, unless they are sufficiently geographically separated. If f_i and f_j are the frequencies assigned to transmitters i and j then this gives rise to constraints of the form:

$$f_i \neq f_j$$

Adjacent channel constraints: When a transmitter and a receiver are tuned to similar frequencies (normally within three channels of each other), there is still the potential for interference. Therefore a number of constraints arise of the following form:

$$|f_i - f_j| > m$$

for some value of m, where m is the required number of channels separation.

Co-site frequency separation: Any pair of frequencies at a site must be separated by a certain fixed amount, typically, for a large problem, 250 kHz or 5 channels. If a frequency is to be used by a high power transmitter then its frequency separation should be larger, say 500 kHz or 10 channels. The constraint can therefore be of the form:

$$|f_i - f_j| \geq m$$

where m refers to the number of channels separation required between transmitters i and j.

For a given problem containing N transmitters, numbered from 1 to N, these interference constraints are represented by a $N \times N$ constraint matrix corresponding to a constraint graph with N vertices. For each pair of transmitters, (i, j), with $1 \leq i < j \leq N$ there is a value $c_{ij} \geq 0$ which measures the minimal frequency separation required for transmitters i and j. Each c_{ij} labels an edge (ij) in the constraint graph. We denote by C the number of c_{ij} which are non-zero, where C is the number of non-trivial constraints and clearly

$$C \leq \frac{N(N-1)}{2}$$

Two types of frequency assignment problem arise in practice, fixed spectrum and minimum span. Both types of problem can be represented by a constraint matrix.

1.2.1 Fixed Spectrum Model

In this problem there is a fixed set of F_{fixed} frequencies denoted by:

$$D = \{d_1, d_2, \ldots, d_{F_{fixed}}\}$$

It is required to determine N values

$$(f_1, f_2, \ldots, f_N) \in D^N$$

such that, for all i, j with $1 \leq i < j \leq N$

$$|f_i - f_j| > c_{ij}$$

There are $(F_{fixed})^N$ possible frequency assignments and because it may be difficult, or even impossible, to determine one which imposes no interference we determine an assignment where the total interference is minimal. If $f = (f_1, f_2, \ldots, f_N)$ denotes a frequency assignment and $E(f)$ denotes the total interference, the problem is to determine an $f \in D^N$ for which $E(f)$ is minimal or near-minimal. FASoft can also solve problems where the set D varies from transmitter to transmitter, and problems where some of the transmitters have a fixed frequency.

1.2.2 Minimum Span Model

In minimum span frequency assignment, the set of available frequencies is not fixed, and could for example, be represented by the set of consecutive integers

$$D = \{1, \ldots, K+1\}.$$

It is required to determine N values

$$(f_1, f_2, \ldots, f_N) \in D^N$$

such that, for all i, j with $1 \leq i < j \leq N$

$$|f_i - f_j| > c_{ij}$$

and K is minimal over all feasible assignments. This minimum value of K is called the *minimum span* and is denoted $sp(G)$, where G is the constraint graph.

1.3 The Frequency Assignment Software

FASOFT is a frequency assignment software package for any problem described by a constraint matrix. The algorithms used in FASOFT are based on meta-heuristics, but also include exact and sequential (greedy) methods, and lower bounding techniques which aid in the assignment process. The system can be used to solve *minimum span* or *fixed spectrum* problems. Minimum span frequency assignment involves assigning radio frequencies to a number of transmitters subject to a number of constraints, such that no interference is suffered (no constraints violated), and the span of frequencies used i.e. the difference between the largest and smallest frequency used, is minimised. In contrast, fixed spectrum frequency assignment involves assigning frequencies from a fixed set (fixed span) of frequencies. Usually this set is too small to enable all constraints to be satisfied, therefore some alternative measure is minimised such as the number of constraint violations. The techniques studied in the project have included sequential and exact algorithms, together with simulated annealing, tabu search and genetic algorithms.

The system has proved successful on three fronts:

- It is capable of carrying out state of the art frequency assignment by one, or a combination, of these techniques, given a constraint matrix.
- It is a research tool capable of exploring and evaluating the effect of varying parameters in the algorithms, the value of new enhancements to the algorithms and the possibility of constructing hybrids of the algorithms.
- The system has proved an invaluable research tool in evaluating the difficulty of particular frequency assignment problems. A combination of theory, developed in the course of the project, and the use of the algorithms, sometimes produces better results than the use of the algorithms alone. The theory may give useful lower

bounds to the number of frequencies required. The configurations used to obtain these lower bounds, usually but not exclusively involving sets of mutually interfering transmitters, may sometimes be assigned either manually or by alternative methods. The assignments can then be extended to assignments of the complete problem using the system.

In particular, the advances in the second year were as follows:

- The software produced at the end of the first year did not have a user friendly interface. Consequently, to enhance its usability an interface that runs under Windows 95, Windows NT and Windows 3.11 has been developed using Borland Delphi. The command line interface is still being supported, in particular for Unix type operating systems.
- Two new methods have been created, which incorporate an efficient and easy to use minimum span frequency assignment algorithm. Simulated Annealing and Tabu Search were optimised for this procedure.
- A starting file, important for minimum span frequency assignment problems, can also be used to significantly improve the performance for fixed spectrum problems.
- The approach of combining theoretical lower bounds with the algorithmic approach is further developed. The aim of this approach is to substantially enhance the performance of the local search algorithms in dealing with global difficulties, and to provide measures of how good the solutions are. One manifestation of these difficulties is the clique. A clique is a maximal set of mutually interfering transmitters. Cliques can be defined for several levels of interference. If the large cliques are not assigned frequencies in an efficient way, then the whole assignment cannot be made efficiently. Clique detection software has been incorporated into the system, together with other software that can determine structures that are similar to cliques. The system has been modified so that the clique structures can be assigned independently of the complete problem. The assignment can then be extended to the complete problem. For the purpose of determining lower bounds it is possible to capitalise on well developed software used in Operational Research to solve moderately sized cases of the so called *travelling salesman problem* (which involves finding a path which connects all nodes (or cities) and which has minimum distance).
- Some empirical results for the classification of problems have been obtained. A theoretical outline has been presented and preliminary results in classifying constraint matrices into three groups have been encouraging.

1.4 Recent Publications

Full details of the work carried out in the first year of the contract are contained in the reports by Smith, Hurley and Thiel [43, 44]. Also, the second year interim report is available [45].

Preliminary results of our work have been presented in Philadelphia at the ACM symposium on Applied Computing [24].

A complete overview of lower bounding techniques in frequency assignment has been presented in [41]. This paper is to appear in the journal *Discrete Mathematics*. Continuing this theme, a paper describing how lower bounds and critical subgraphs can be used to improve heuristics for the frequency assignment process has been submitted for publication [42]. Also, a paper describing the importance of the initial ordering of vertices in the maximum clique algorithm has been submitted [49].

Finally, a paper describing FASoft and its functionality as a frequency assignment system [22] has just been accepted for publication in Radio Science subject to some revisions.

2 The Frequency Assignment Algorithms

In this section the various frequency assignment algorithms are introduced. All the algorithms are incorporated in FASoft [22, 46].

2.1 Sequential Assignment Methods

Sequential assignment methods are greedy methods which consist of three main steps or *modules* based on the work which appeared in [20]. First, the transmitters are listed in some specified order and then the first transmitter is assigned to frequency one. Second, the next transmitter to be assigned is selected (which may differ from the initial ordering). Finally, the selected transmitter is assigned to a selected frequency. Several options are available for each module. By using different options in each module a total of 64 different sequential methods can be obtained for assignment.

- Initial Ordering.
  - *Largest degree first (LF1).* The transmitters are listed in decreasing order of their degree (i.e. number of incident edges in the constraint graph).
  - *Largest degree first (LF2).* Again the transmitters are listed in decreasing order of their degree but this time the calculation of the degree excludes the transmitters that have already been ordered (and therefore have been removed from the constraint graph together with all incident edges).
  - *Smallest degree last (SL).* This time the transmitters of smallest degree are removed from the constraint graph. When all the transmitters have been removed the list is then reversed to give the final ordering. A number of references have reported that the smallest last ordering requires fewer frequencies than the largest first ordering.

  All of the above orderings use the degree of a transmitter (i.e. the number of incident edges of the corresponding vertex in the constraint graph) to calculate an ordering. Additional orderings can be formed by using, instead, the generalised degree of each transmitter. This takes into account the weight of the edges incident to any transmitter i.e. it takes into account the severity of the constraints. The generalised degree of a transmitter is calculated as the sum of all weights on all edges incident with the transmitter. Therefore, we have the following additional initial orderings

  - *Generalised largest first (GLF1).*

  - *Generalised largest first (GLF2).*
  - *Generalised smallest last (GSL).*

In situations where each transmitter i, $(i = 1, \ldots, N)$ has its own frequency domain D_i (with cardinality $|D_i|$), an alternative way to order the transmitters is to consider the domain sizes. Let the union domain, D be defined as $D_1 \cup D_2 \cup \ldots \cup D_N$. Two options are available:

  - *Size of domain (SD).* The transmitters are listed in increasing order of their domain cardinality.
  - *Size of domain and generalised degree (SDG).* Associate with each transmitter i a value x_i where

$$x_i \;=\; |D| - |D_i| + \text{generalised degree of transmitter } i$$

    The transmitters are listed in decreasing order of the values x_i, $(i = 1, \ldots, N)$.

- Selecting the next transmitter.
  - *Sequential (S).* Select the next unassigned transmitter in the list generated by the initial ordering.
  - *Generalised saturation degree (GSD).* Let V be a set of transmitters and V_c be the transmitters of V already assigned frequencies. Frequency n is said to be denied to the unassigned transmitter v if there is a transmitter u in V_c assigned to frequency n such that transmitter v and u would interfere i.e. assuming an edge exists between u and v in the constraint graph then there is insufficient frequency separation between them. If frequency n is denied to transmitter v, the *influence* of frequency n, denoted by I_{nv}, is the largest weight of any edge connecting v to a transmitter assigned to frequency n. The number

$$\sum I_{nv}$$

    (where the sum is taken over all frequencies n denied to v) is called the *generalised saturation degree* of v. The technique for selecting the next transmitter is as follows: Select a transmitter with maximal generalised saturation degree (break ties by selecting the transmitter occurring first in the initial ordering).

    This technique attempts to assign frequencies to a large weighted clique first and can only be used when all frequency domains are equal.

- Selecting a frequency.
  - *Smallest acceptable (SAF).* The simplest technique is to assign the selected transmitter v to the smallest acceptable frequency i.e. the lowest numbered frequency to which v can be assigned without violating any constraints.

– *Acceptable occupied (AOF).* The selected transmitter is assigned any acceptable occupied frequency (no ordering of the occupying frequencies occurs), if there is no acceptable occupied frequency, assign the transmitter to the smallest acceptable frequency.
– *Smallest acceptable occupied (SAOF).* This technique attempts to minimise the number of frequencies used in the assignment. The selected transmitter is assigned to the smallest acceptable occupied frequency, if there is no acceptable occupied frequency, assign the transmitter to the smallest acceptable frequency.
– *Smallest acceptable most heavily occupied (SAHOF).* The selected transmitter is assigned to the smallest acceptable most heavily occupied frequency, if there is no acceptable occupied frequency assign the transmitter to the smallest acceptable frequency.

Sequential methods are useful in that they provide solutions to the minimum span problem within a reasonable amount of time (typically seconds or minutes). Even though the span of the assignments is usually suboptimal, in some cases they are surprisingly good. However, a sequential algorithm that performs well on one problem does not necessarily give the best span (or even a reasonable span) on another problem. It is also difficult to assess which sequential algorithm will perform well on a specified problem.

Therefore we find that the best way to proceed is to try all possible combinations of options within each module. (This gives 48 possible sequential algorithms when all frequency domains are the same). The assignment with the best span is then chosen.

More importantly for the work presented in this paper, sequential methods provide a zero violation assignment which is used as the starting point to the minimum span algorithms based on simulated annealing and tabu search.

2.2 Exhaustive Search

The incorporation of exhaustive search techniques into an heuristic system is important to enable verification of the meta-heuristics on small problems. Exhaustive search techniques are able to find solutions given a certain frequency span or alternatively prove that such a solution does not exist. For large problems (large not only refers to the number of transmitters but also the density and complexity of constraints) exhaustive search techniques are impractical. In our experience some 30 transmitter problems are already too difficult, whereas sometimes 50 transmitter problems are relatively easy to solve.

For a very small problem size, about 20 transmitters or less, the exhaustive search techniques are sometimes even quicker in providing a solution than heuristic algorithms. An added bonus is that exhaustive search techniques can prove optimality.

Generally, a non-intelligent exhaustive search algorithm would be required to check

$$(\textit{number of frequencies}) \text{ to the power of } (\textit{number of transmitters})$$

assignments. Exhaustive search techniques try to reduce this number, while keeping low overheads. In the following, the two techniques, which are incorporated into FASOFT, *backtracking* and *forward checking* are described. The implementation of the methods follows the structure outlined in [36].

Backtracking

Backtracking is the simplest exhaustive search technique. Sequentially the transmitters are assigned frequencies [1]. The first frequency d_1 from the domain D will be assigned to transmitter i, i.e. $f_i = d_k,\ k = 1$. The backtracking algorithm then checks for constraint violations with all already assigned transmitter frequencies $f_j,\ \forall\ j < i$. If there is no violation, the algorithm moves on to the next transmitter. This is called a forward move. If there is a violation, the algorithm checks the next frequency in the domain. This means that the next frequency is selected from the domain and assigned to transmitter i, i.e. $f_i = d_{k+1}$. If all frequencies from the domain D have been unsuccessfully assigned the forward move *fails* and the algorithm *backtracks* from the previous transmitter $i - 1$.

If the algorithm, in its simplest form, backtracks to the first transmitter and all frequencies for the first transmitter have already been tried, the algorithm terminates, therefore proving that an assignment with the specified frequencies and constraints is not possible.

Forward Checking

In contrast to backtracking, forward checking moves forward only when it already knows that this move will not violate any previous assigned transmitters. For this purpose it requires the use of a current domain D^c instead of the usual domain D. The current domain is initially set to the usual domain, $D^c = D$. When one frequency f_i is selected from the current domain, all constraints with future transmitters have to be examined and violating frequencies will be deleted from the current domains $D^c_j, j > i$. This assures that when assigning a frequency to a transmitter it does not violate any constraints with already assigned transmitters.

[1] The initial ordering of the transmitters can affect the runtime of both the backtracking and forward checking algorithms. In FASOFT the transmitters can be ordered using any of the techniques outlined in section 2.1

A backward move is necessary, when a transmitter assignment empties the current domain of a future transmitter, $D_j^c = \emptyset,\ j > i$. The algorithm *backtracks* exactly as with the above backtracking technique, however the current domains must be restored to the state before the rejected frequency was assigned. This requires that during the forward move the deleted frequencies must be stored in a stack to be retrieved later.

In comparison, forward checking is more work intensive than backtracking, however it *fails* earlier and thus greatly reduces the number of checked assignments compared with the simpler backtracking technique.

2.3 Simulated Annealing

Simulated annealing (SA) is a stochastic computational technique derived from statistical mechanics for finding near globally-minimum-cost solutions to large optimisation problems. In many instances, finding the global minimum value of an objective function with many degrees of freedom subject to conflicting constraints is an NP-complete problem, since the objective function will tend to have many local minima. A procedure for solving optimisation problems of this type should sample the search space in such a way that it has a high probability of finding the optimal or a near-optimal solution in a reasonable time. Over the past decade or so, simulated annealing has shown itself to be a technique which meets these criteria for a wide range of problems.

The method itself has a direct analogy with thermodynamics, specifically with the way that liquids freeze and crystallise, or metals cool and anneal. At high temperatures, the molecules of a liquid move freely with respect to one another. If the liquid is cooled slowly, thermal mobility is restricted. The atoms are often able to line themselves up and form a pure crystal that is completely regular. This crystal is the state of minimum energy for the system, which would correspond to the optimal solution in a mathematical optimisation problem. However, if a liquid metal is cooled quickly i.e. quenched, it does not reach a minimum energy state but a somewhat higher energy state corresponding, in the mathematical sense, to a suboptimal solution found by iterative improvement or hill-climbing.

In order to make use of this analogy with thermodynamical systems for solving mathematical optimisation problems, one must first provide the following elements:

1. A description of possible system configurations, i.e. some way of representing a solution to the minimisation (maximisation) problem, usually this involves some *configuration* of parameters $\mathbf{X} = (x_1, x_2, ..., x_N)$ that represent a solution.

2. A generator of random changes in a configuration; these changes are typically solutions in the *neighbourhood* of the current configuration, for example, a change in one of the parameters, x_i.

3. An objective or cost function $E(\mathbf{X})$ (analogue of energy) whose minimisation is the goal of the procedure.

4. A control parameter t (analogue of temperature) and an *annealing schedule* which indicates how t is lowered from high values to low values e.g. after how many random changes in configuration is t reduced and by how much?

Metropolis, Rosenbluth, Rosenbluth, Teller and Teller [32] first introduced principles of these kinds into numerical minimisation. Given a succession of *moves* (i.e. neighbouring configurations), a simulated thermodynamical system was assumed to change its configuration from energy E_{old} to energy E_{new} with probability

$$prob \ = \ e^{(-(E_{new}-E_{old})/Bt)}$$

(the so-called Boltzmann distribution) for a fixed t and B known as the Boltzmann constant. If $E_{new} < E_{old}$ then the new configuration has a lower energy state than the old one and the system always accepts this move. If $E_{new} > E_{old}$ then the new configuration may still be accepted with probability $prob$ ($0 < prob < 1$) and thus help the system jump out of a local minimum. This general scheme, of always taking a downhill step while sometimes taking a uphill step is known as the Metropolis Algorithm.

The simulated annealing procedure of Kirkpatrick, Gelatt and Vecchi [26] uses the Metropolis Algorithm but *varies* the temperature parameter t from a high value (system at "melting point" i.e. accept most new configurations) to a low value (system at "freezing point" i.e. accept no new configurations). The full SA procedure for minimisation is described in Figure 1 (for maximisation set $E = -E$):

where NUM_{loop} is the number of random changes in configuration at each temperature t and is chosen so that the configuration has reached a minimum energy state for the current temperature. The variable *random* is a randomly generated number in the range [0,1].

2.3.1 SA Implementation for Frequency Assignment

Cost (Fitness) Function

The fitness function, E, minimised by the simulated annealing procedure is formulated so that several factors can be minimised:

```
Initialise t
Generate random configuration X_old
WHILE t > t_min DO
    FOR i = 1 to NUM_loop DO
        generate new configuration, X_new, from X_old
        calculate new energy, E_new
        calculate ΔE = E_new − E_old
        IF ΔE < 0 or random < prob = e^(−ΔE/t) THEN
            X_old ⇐ X_new
            E_old ⇐ E_new
        END IF
    END FOR
    reduce t (e.g.  t = 0.9t)
END WHILE
```

Figure 1: The SA procedure

- number of violated constraints, e_{vio}
- the sum of the amounts by which each constraint is violated, e_{sum}
- the difference between the largest frequency, f_{large}, and the smallest frequency, f_{small}, used
- number of distinct frequencies used, e_{order}
- the largest frequency used, f_{large}
- the largest of the constraint violations, l_{vio}

Therefore we minimise

$$E = \mu_1 e_{vio} + \mu_2 e_{sum} + \mu_3 (f_{large} - f_{small}) + \mu_4 e_{order} + \mu_5 f_{large} + \mu_6 l_{vio}$$

where the μ_i are weights, which can include 0, that reflect the relative importance of the various factors. For example, a solution which has four violations i.e. $e_{vio} = 4$ with the sum of the constraint violations equal to four i.e. $e_{sum} = 4$ (one for each of the violated constraints) would be considered more useful than a solution which has one violation i.e. $e_{vio} = 1$ but a constraint violation sum of 4 i.e. $e_{sum} = 4$. The practical analogy is that in the former case the resulting plan will have an assignment which has a small amount of interference for a small number of nodes, while the latter will contain perhaps two nodes which severely interfere with each other. In addition to the μ_i weights, the different types of constraint can also be weighted to reflect their relative importance, e.g. satisfying the co-site frequency separation constraints is more important than satisfying the adjacent channel constraints and so would be given a higher weight.

This cost function is used as the basis for evaluating the quality of solutions in all the algorithms used in FASoft. Generally the first two factors are the most important.

Representation of an Assignment

A frequency assignment $f = (f_1, \ldots, f_N)$ is represented using an array of indexes $[x_1, \ldots, x_N]$ where $f_j = d_{x_j}$ for $1 \leq j \leq N$. To illustrate this consider a problem which has six transmitters (numbered 1,2,...,6) and three frequencies are available $\{d_1, d_2, d_3\}$. The assignment (3,1,2,3,3,1) would indicate that transmitter 1 is assigned frequency d_3, transmitter 2 is assigned frequency d_1 and so on up to transmitter 6 which is assigned frequency d_1.

Generation of New Configurations

Three different generators can be used to produce a new assignment from the current assignment.

Single Move

Here the neighbours of f are those assignments where the array of indexes differs in precisely *one* component. Thus if f' is represented by $[x'_1, \ldots, x'_N]$ then $f' = (f'_1, \ldots, f'_N)$ is a neighbour of f if there exists j, $1 \leq j \leq N$, such that $x'_j \neq x_j$, and for all $i = 1, \ldots, N$ with $i \neq j$ we have $x'_i = x_i$.

Any assignment f has $N(F_{fixed} - 1)$ neighbours. Each neighbour corresponds to a pair (j, x'_j) with $1 \leq j \leq N$, $1 \leq x'_j \leq F_{fixed}$, $(x'_j \neq x_j)$.

Double Move

Here the neighbours of f are those assignments where the array of indexes differs in precisely *two* components. If f' is represented by $[x'_1, \ldots, x'_N]$ then $f' = (f'_1, \ldots, f'_N)$ is a neighbour of f if there exists j_1 and j_2, $1 \leq j_1, j_2 \leq N$, such that $x'_{j_1} \neq x_{j_1}$ and $x'_{j_2} \neq x_{j_2}$, and for all $i = 1, \ldots, N$ with $i \neq j_1$ or j_2 we have $x'_i = x_i$.

Any assignment f has $\frac{N!(F_{fixed}-1)^2}{2!(N-2)!}$ neighbours. Each neighbour corresponds to two pairs (j_1, x'_{j1}) and (j_2, x'_{j2}) with $1 \leq j_1, j_2 \leq N$, $1 \leq x'_{j1}, x'_{j2} \leq F_{fixed}$, ($x'_{j1} \neq x_{j1}$ and $x'_{j2} \neq x_{j2}$).

Restricted Single Move

Let t_v be the number of transmitters in an assignment f which are assigned frequencies that violate one or more of the constraints. Denote such transmitters as *violating transmitters*. The *restricted single move* generator involves randomly selecting a violating transmitter i and setting $x_i = x'_i$ where $1 \leq x'_i \leq F_{fixed}$.

Starting and finishing temperatures

The starting temperature is determined by first setting $t_0 = 1$ and performing 100 iterations of the FOR loop from the main loop of the above algorithm. If the acceptance ratio, χ, defined as the number of accepted trial assignments divided by 100 (NUM_{loop}), is less than 0.9, double the current value of t_0. Continue this procedure until the observed acceptance ratio exceeds 0.9 (with χ reinitialised to zero prior to starting the FOR loop).

The algorithm terminates when the temperature, t_k, falls below t_{min} (user specified) or the number of *frozen temperatures* exceeds a user specified value, usually 10. A frozen temperature occurs when no new assignments are accepted for a given temperature t_k (i.e. after NUM_{loop} iterations of the FOR loop).

Annealing Schedule (reduction of t_k)

Three annealing schedules are included in FASoft. In all three cases the parameter NUM_{loop} is set to N, the number of transmitters.

Cooling 1 (geometric):

With a simple cooling scheme t_k is reduced according to the formula:

$$t_{k+1} = \alpha t_k, \qquad \text{with } \alpha \in [0, 1]$$

Cooling 2 (Costa [8])

The geometric scheme (cooling 1) decreases the temperature by the specification of the parameter α; the number of iterations at each temperature is fixed at N. In the cooling 2 scheme, the temperature reduction is again specified by the parameter α, however the number of iterations at each temperature is specified by

$$N_{k+1} = \lceil N_k/\alpha \rceil$$

(up to a maximum of $2000N$), where $N_0 = N$, the number of transmitters.

As the temperature decreases the number of trial assignments tested at each temperature increases. This scheme was found to be particularly effective in [8].

Cooling 3 (Hurley and Smith [23]):

A more complex scheme (slower but gives better results) is when the parameter t_k is calculated using:

$$t_{k+1} = t_k \cdot \left(1 + \frac{\ln(1+\delta) \cdot t_k}{3\sigma(t_k)}\right)^{-1}$$

where δ is set to 0.1 and

$$\sigma(t_k) = \frac{1}{NUM_{loop}} \cdot \sqrt{\Phi}$$

where

$$\Phi \; = \; NUM_{loop} \cdot \sum_{i=1}^{NUM_{loop}} \left({E_i}^k\right)^2 - \left(\sum_{i=1}^{NUM_{loop}} {E_i}^k\right)^2 + 0.5$$

and where ${E_i}^k$ is the cost function value for the assignment obtained at iteration i, at temperature t_k.

2.4 Tabu Search

Tabu search (TS) was first suggested by Glover [15] and since then has become increasingly used. It has been successfully applied to obtain optimal or sub-optimal solutions to such problems as scheduling, timetabling, travelling salesman and layout optimisation.

The basic idea of the method, described by Glover, Taillard and de Werra [16], is to explore the *search space* of all feasible solutions by a sequence of *moves*. A move from one solution to another is the best available. However, to escape from locally optimal but not globally optimal solutions and to prevent cycling, some moves, at one particular iteration, are classified as forbidden or *tabu* (or *taboo*). Tabu moves are based on the short-term and long-term history of the sequence of moves. A simple implementation, for example, might classify a move as tabu if the reverse move has been made recently or frequently. Sometimes, when it is deemed favourable, a tabu move can be overridden. Such *aspiration criteria* might include the case which, by forgetting that a move is tabu, leads to a solution which is the best obtained so far.

Formally, tabu search is applied to an optimisation problem. Suppose h is the real-valued objective function on a search space S and it is required to find $s \in S$ such that $h(s)$ has minimal value. For combinatorially hard problems, this requirement needs to be relaxed to finding $s \in S$ such that $h(s)$ is close to the minimal value (a *sub-optimal* value). Sub-optimal problems may be obtained by halting when a certain threshold for an acceptable solution has been achieved or when a certain number of iterations have been completed.

A characterisation of the search space S for which tabu search can be applied is that there is a set of k moves $M = \{m_1, ..., m_k\}$ and the application of the moves to a feasible solution $s \in S$ leads to k, usually distinct, solutions $M(s) = \{m_1(s), ..., m_k(s)\}$. The subset $N_{set}(s) \subseteq M(s)$ of *feasible* solutions is known as the *neighbourhood* of s.

The method commences with a (possibly random) solution $s_0 \in S$ and determines a sequence of solutions $s_0, s_1, ..., s_n \in S$. At each iteration, $s_{j+1}(0 \leq j < n)$ is selected from the neighbourhood $N_{set}(s_j)$. The process of selection is first to determine the tabu set $T_{set}(s_j) \subseteq N_{set}(s_j)$ of neighbours of s_j and the aspirant set $A_{set}(s_j) \subseteq T_{set}(s_j)$ of tabu neighbours. Then s_{j+1} is the neighbour of s_j which is either an aspirant or not tabu and for which $h(s_{j+1})$ is minimal; that is, $h(s_{j+1}) \leq h(s')$ for all $s' \in (N_{set}(s_j) - T_{set}(s_j)) \cup A_{set}(s_j)$. A more formal description of tabu search can be found in Figure 2.

> generate initial solution s
>
> WHILE *not finished*
>
> > Identify $N_{set}(s) \subset S$. (Neighbourhood set)
> >
> > Identify $T_{set}(s) \subseteq N_{set}(s)$. (Tabu set)
> >
> > Identify $A_{set}(s) \subseteq T_{set}(s)$. (Aspirant set)
> >
> > Choose $s' \in (N_{set}(s) - T_{set}(s)) \cup A_{set}(s)$, for which $h(s')$ is minimal.
> >
> > $s = s'$.
>
> END WHILE

Figure 2: The TS procedure

Note that, it is possible, and even desirable to avoid convergence at a local minimum, that

$h(s_{j+1}) > h(s_j)$. The conditions for a neighbour to be tabu or an aspirant will be problem specific. For example, a move m_r may be tabu if it could lead to a solution which has already been considered in the last q iterations (*short-term memory*) or which has been repeated many times before (*long-term memory*). A tabu move satisfies the aspiration criteria if, for example, the value of $h(s')$ with $s' \in T_{set}(s_j)$ satisfies $h(s') < h(s_i)$ for all i, $0 \leq i \leq j$.

2.4.1 Tabu Search Implementation for Frequency Assignment

Here we describe how the tabu search procedure has been implemented for the frequency assignment problem. We explain the representation of assignments, the notions of neighbourhood and move and how short and long term memory are measured.

Representation of an Assignment and Neighbourhood Structure

As in the simulated annealing algorithm a frequency assignment $f = (f_1, \ldots, f_N)$ is represented using an array of indexes $[x_1, \ldots, x_N]$ where $f_j = d_{x_j}$ for $1 \leq j \leq N$.

Full Neighbourhood

This includes all the neighbours of an assignment f produced by using the *single move* generator described for the simulated annealing algorithm. Thus, for tabu search this involves calculating $N(F_{fixed} - 1)$ neighbours at each iteration.

Restricted Neighbourhood

The full neighbourhood is generally too large for it to be practical for it to completely searched for at each iteration. To reduce the neighbourhood, similar steps to those used by Bouju et. al. [5] are incorporated. Therefore, this neighbourhood consists of all neighbours of an assignment f produced by using the *restricted single move* generator described for the simulated annealing algorithm. This increases the efficiency of the algorithm since this restricted neighbourhood will tend to be much smaller than the full neighbourhood ($t_v(F_{fixed} - 1)$ compared to $N(F_{fixed} - 1)$). However, if the restricted neighbourhood consists of moves which are tabu and do not satisfy the aspiration criteria, the full neighbourhood will be selected for this iteration.

Restricted Random Neighbourhood

Here the neighbourhood of an assignment f consists of N neighbours. Each neighbour in the neighbourhood is generated by randomly selecting a violating transmitter and randomly assigning a different frequency.

Definition of a Tabu Move: Short and Long Term Memory

A move to a neighbour (i, x_i') corresponding to changing the assignment of transmitter i to x_i' is said to be *tabu* if it does not satisfy the short term or long term memory condition. These conditions are determined by two numbers: a positive integer r and a real number β, where $0 < \beta < 1$. The short term memory condition specifies that any transmitter, i, cannot be assigned the same frequency over any of the previous r moves. The long term memory condition specifies that the proportion of the number of times transmitter i had been changed over all iterations does not exceed β. Thus, if at iteration j, the frequency $d_{x_i j}$ assigned to transmitter i has changed, then a move at iteration $k+1$ is tabu if either (short term memory)

$$d_{x_i{}^{k+1}} = d_{x_i{}^t} \qquad \text{where} \quad k - r < t \leq k$$

or (long term memory)

$$\frac{1}{k} \sum_{j=1, i_j = i_{k+1}}^{k} 1 \quad > \quad \beta$$

At each iteration, the method selects from the non-tabu neighbours that neighbour $f^{(k+1)}$ of $f^{(k)}$ for which $E(f^{(k+1)})$ is minimal (E as defined above). Note that it is possible that $E(f^{(k+1)}) > E(f^{(k)})$; this allows escape from local minima.

Tabu search allows a tabu move to be selected when certain aspiration criteria are satisfied. In our implementation, a tabu move is selected if the neighbour $f^{(k+1)}$ satisfies

$$E(f^{(k+1)}) \leq E(f')$$

for all neighbours f' of $f^{(k)}$ and

$$E(f^{(k+1)}) < E(f^{(j)}) \qquad \forall\, j, \quad 1 \leq j \leq k.$$

To summarise, if f_{tabu} denotes the best tabu neighbour of $f^{(k)}$ and f_{non} the best non-tabu neighbour, then the rule for determining $f^{(k+1)}$ is

$$\begin{aligned} &\text{if } E(f_{tabu}) < E(f_{non}) \text{ and} \\ &\quad E(f_{tabu}) < E(f^{(j)}) \quad \text{for} \quad 1 \leq j \leq k \end{aligned}$$

then $f^{(k+1)} = f_{tabu}$
else $f^{(k+1)} = f_{non}$

Notice that there are some implicit relations between the parameters r and β which have to be satisfied. If $r > N$ or $\beta < 1/N$, then, after N non-tabu moves, every move is tabu. Also, after k iterations of non-tabu moves, it follows from the recency condition that a transmitter can have changed at most $\lceil k/r \rceil$ times. Hence, the long term memory value, to have any effect, should satisfy

$$k\beta < \lceil k/r \rceil$$

We obtain, therefore, a relation between r and β which is

$$1/N \leq \beta \leq 1/r$$

In practice, we choose

$$\beta = \lambda/r + (1-\lambda)/N$$

for some value of λ with $0 \leq \lambda \leq 1$.

In FASoft, the default values for the short term memory and long term memory conditions are $r = \frac{2*N}{5}$ and $\lambda = 0.5$ respectively.

The algorithm is terminated if zero interference has been obtained or if we reach a predetermined maximum number of iterations. We then produce an assignment with the lowest measure of interference.

2.5 SA and TS Implementation for Minimum Span Frequency Assignment

The above implementations of simulated annealing and tabu search have been described in relation to fixed spectrum frequency assignment i.e. given a fixed number of frequencies, minimise the cost function E. An alternative formulation, which is based on the work of Costa [8], uses hybrid SA/sequential and TS/sequential algorithms for minimum span problems.

Unlike the standard algorithms given earlier for fixed spectrum frequency assignment, which generate a random assignment, the starting point for minimum span assignment is any no-violation assignment usually generated by one of the sequential algorithms. The system then reassigns all transmitters which are assigned to the highest used frequency, to a randomly chosen smaller frequency. That is, it deletes the highest frequency from the frequencies available for use. This step will generally introduce a small number of violations, but preserves some features present in the original assignment. The next step then involves using SA or TS (the components of each algorithm are the same as in fixed spectrum assignment) to find a valid assignment with no violations using the reduced frequency domain. Once this is achieved, the assignment found is taken as the start assignment and the process starts again. The loop terminates, when the heuristic is unable to reduce the number of violations to zero. Therefore, the assignment taken as the start assignment in the previous iteration contains the best zero violation assignment found.

The structure of the minimum span algorithms are as follows:

Step 1:. Generate a no-violation assignment e.g using a sequential method. Say this assignment has span q.

Step 2:. Transmitters assigned with frequency q are randomly assigned a frequency $< q$. This will normally introduce constraint violations.

Step 3:. Perform an SA or TS fixed spectrum algorithm to eliminate the constraint violations introduced in Step 2 i.e. find a zero violation assignment with span $q - 1$. The cost function is defined by $\mu_1 = \mu_3 = \mu_4 = \mu_5 = \mu_6 = 0$ and $\mu_2 = 1$.

Step 4:. If a zero violation assignment is produced in Step 3, set $q = q - 1$ and go to Step 2. If there are constraint violations then output q as the best span obtained and its assignment.

Neighbourhood

As already mentioned, in the minimum span implementation a valid assignment is reduced by one frequency and this generally introduces an assignment with a few constraint violations. It is inefficient to change frequencies for transmitters which are not involved in violations. We therefore limit the choice of transmitters to only those, which violate the assignment. In particular, for all the test examples, given later, we select neighbours using the *restricted single move* for simulated annealing and using the *restricted random neighbourhood* for tabu search. Selecting neighbours in this way increases the efficiency of the algorithms, and also increases the chances that the good features already present in the starting assignment remain intact.

Minimum span frequency assignment results can be found in sections 5.2 and 5.3.

2.6 Genetic Algorithms

Genetic algorithms have been proposed by Holland [21] to mimic some of the processes of natural evolution and selection. In nature, each species needs to adapt to a complicated and changing environment in order to maximise the likelihood of its survival. The knowledge that each species gains is encoded in its chromosomes which undergo transformations when reproduction occurs. Over a period of time, these changes to the chromosomes give rise to species that are more likely to survive, and so have a greater chance of passing their improved characteristics on to future generations. Of course, not all changes will be beneficial but those which are not tend to die out.

Holland's genetic algorithm attempts to simulate nature in the following manner. The first step is to represent a solution to the problem by a string of *genes* that can take on some value from a specified finite range or alphabet. This string of genes, which represents a solution, is known as a *chromosome*. Then an initial population of legal chromosomes is constructed at random. At each generation, the fitness of each chromosome in the population is measured (a high fitness value would indicate a better solution than a low fitness value). The fitter chromosomes are then selected to produce offspring for the next generation, which inherit the best characteristics of both the parents. After many generations of selection for the fitter chromosomes, the result is hopefully a population that is substantially fitter than the original. The theoretical basis for the genetic algorithm is the *Schemata Theorem* [21], which states that the individual chromosomes with good, short, low-order schemata or building blocks (i.e. beneficial parts of the chromosome) receive an exponentially increasing number of trials in successive generations.

All genetic algorithms consist of the following main components:

<u>Chromosomal Representation</u>

Each chromosome represents a possible solution to the problem and is composed of a string of genes. The binary alphabet {0,1} is often used to represent these genes but sometimes, depending on the application, integers or real numbers are used. In fact, almost any representation can be used that enables a solution to be encoded as a finite length string.

<u>Initial Population</u>

Once a suitable representation has been decided upon for the chromosomes, it is necessary to create an initial population to serve as the starting point for the genetic algorithm. This initial population can be created randomly or by using specialised, problem specific, information. From empirical studies, over a wide range of function optimisation problems, a population size of between 30 and 100 is usually recommended.

Fitness Evaluation

Fitness evaluation involves defining an objective or fitness function against which each chromosome is tested for suitability for the environment under consideration. As the algorithm proceeds we would expect the individual fitness of the "best" chromosome to increase as well as the total fitness of the population as a whole.

Selection

We need to select chromosomes from the current population for reproduction. If we have a population of size $2n$, the selection procedure picks out two parent chromosomes, based on their fitness values, which are then used by the crossover and mutation operators (described below) to produce two offspring for the new population. This selection/crossover/mutation cycle is repeated until the new population contains $2n$ chromosomes i.e. after n cycles. In fitness proportional selection [17] the higher the fitness value the higher the probability of that chromosome being selected for reproduction; i.e. for a chromosome x, its probability of selection is given by:

$$\frac{\text{fitness of } x}{\text{total population fitness}}$$

Crossover and Mutation

Once a pair of chromosomes has been selected, crossover can take place to produce offspring. A crossover probability of 1.0 indicates that all the selected chromosomes are used in reproduction i.e. there are no survivors. However, empirical studies have shown that better results are achieved by a crossover probability of between 0.65 and 0.85, which implies that the probability of a selected chromosome surviving to the next generation unchanged (apart from any changes arising from mutation) ranges from 0.35 to 0.15. Three of the standard crossover operators are *one point*, *two point* and *uniform*. Details of the implementation of these operators for frequency assignment will be given later.

If we only use the crossover operator to produce offspring, one potential problem that may arise is that if all the chromosomes in the initial population have the same value at a particular position then all future offspring will have this same value at this position. For example, if all the chromosomes have a 0 in position two then all future offspring will have a 0 at position two. To combat this undesirable situation a *mutation* operator is used. This attempts to introduce some random alteration of the genes e.g. 0 becomes 1 and vice versa. Typically this occurs infrequently so mutation is of the order of about one bit changed in a thousand tested. Each bit in each chromosome is checked for possible mutation by generating a random number between zero and one and if this number is less than or equal to the given mutation probability, e.g. 0.001, then the bit value is changed.

This completes one cycle of the simple genetic algorithm. The fitness of each chromosome in the new population is evaluated and the whole procedure repeated, (Figure 3). In Figure 3, *finished* indicates either an optimal or suitable suboptimal solution has been found or the maximum number of generations has been exceeded.

```
Generate random population
REPEAT
    evaluate fitness of current population
    select chromosomes, based on fitness, for reproduction
    perform crossover and mutation to give new improved population
UNTIL finished
```

Figure 3: The GA procedure

2.6.1 Genetic Algorithm Implementation for Frequency Assignment

Representation of Chromosomes

The length of each chromosome is equal to N, which corresponds to the number of transmitters to be assigned a frequency. The value of each element in the chromosome is an integer corresponding to a frequency,

f_1	f_2	f_3	...	...	f_N

where f_i represents the frequency assigned to transmitter i.

Initial Population

The initial population of chromosomes is set up randomly i.e. for each chromosome each transmitter within the chromosome is assigned a random frequency from the available set of frequencies.

Genetic Operators: Crossover and Mutation

Crossover operators combine characteristics of two parent structures to form two similar offspring. The representation used above for the chromosomes allows the use of standard *one-point*, *two-point* and *uniform* crossover operators.

One-Point Crossover - randomly select a crossover position and crossover the tails:

f_1	f_2	f_3	f_4	f_5	\/	f_6	f_7	parent 1
g_1	g_2	g_3	g_4	g_5	\/	g_6	g_7	parent 2

$$\Downarrow$$

f_1	f_2	f_3	f_4	f_5		g_6	g_7	child 1
g_1	g_2	g_3	g_4	g_5		f_6	f_7	child 2

Two-Point Crossover - similar to one-point only this time two positions are randomly chosen and the middle section is interchanged:

f_1	\/	f_2	f_3	f_4	f_5	\/	f_6	f_7	parent 1
g_1	\/	g_2	g_3	g_4	g_5	\/	g_6	g_7	parent 2

$$\Downarrow$$

g_1		f_2	f_3	f_4	f_5		g_6	g_7	child 1
f_1		g_2	g_3	g_4	g_5		f_6	f_7	child 2

Uniform Crossover - This is the most generalised crossover operator and generates a mask for each application of the operator. This mask determines from which parent the genetic material (frequency) is taken for each gene (transmitter number):

f_1	f_2	f_3	f_4	f_5	f_6	f_7	parent 1
g_1	g_2	g_3	g_4	g_5	g_6	g_7	parent 2
1	0	1	1	1	0	1	mask

$$\Downarrow$$

f_1	g_2	f_3	f_4	f_5	g_6	f_7	child 1
g_1	f_2	g_3	g_4	g_5	f_6	g_7	child 2

In addition to the three standard crossover operators, offspring can be generated by using *heuristic crossover* operators i.e. by using simulated annealing, tabu search or hill-climbing.

The operation of the heuristic crossover operators is straightforward:

1. Two parent chromosomes are selected using the selection routine. Say these are represented by:

$$\begin{array}{cccccc} f_1 & f_2 & f_3 & f_4 & \dots & f_N \\ g_1 & g_2 & g_3 & g_4 & \dots & g_N \end{array}$$

2. Next, depending on which option was selected, a standard simulated annealing, tabu search or hill-climbing procedure is carried out to produce the best or near best single offspring, where the domain of each transmitter $i, (i = 1, ..., N)$, is restricted to the two frequencies assigned to transmitter i in the parents, i.e. $\{f_i, g_i\}$. Thus, one of the offspring is the best or near best chromosome where each transmitter frequency is restricted to one used by its parents. The second offspring is the complement of the first offspring.

Mutation acts as a background operator and introduces new genetic material into the population of chromosomes. Mutation is at random, according to a prescribed rate; an element of a chromosome is replaced by a random, allowable, frequency.

In order to give the genetic operators an added effectiveness, a *quick hill-climb* (see section 2.8) can be performed on each offspring after the crossover and mutation. The quick hill-climb procedure modifies the current offspring to a neighbouring offspring with lower cost. In contrast to *1-opt hill-climb* (see section 2.8) it stops immediately after a neighbour that has a lower cost is found. This does not guarantee to find the best neighbour, but it is significantly faster than the 1-opt hill-climb and is a useful improvement compared with the original offspring.

2.7 Pyramid Genetic Algorithm

This pyramid genetic algorithm (PGA), the basis of which was developed at the University of Limberg [47], starts with a population size which is a power of two. It sorts the elements of the population in nonincreasing order of their fitness (i.e. cost function value) and applies the crossover to the pairs of solutions consisting of an odd element and the next even element in this ordering. Only the best offspring of every crossover is transferred to the next population. Therefore the size of the population is halved in every generation of the algorithm. The algorithm stops when there is only one solution in the population. This solution is the output of the algorithm.

The crossover operators available are the same as those outlined in section 2.6.1 for the standard genetic algorithm, including the heuristic crossover operators. The only difference is that in the pyramid genetic algorithm the population elements are ordered and the parents are selected as indicated above.

2.8 Hill Climb Algorithms

1-Opt Hill-Climb

The starting point of the procedure is generated by assigning each transmitter a randomly chosen frequency from D. The cost function value (interference) of this assignment is then calculated. A neighbouring assignment is generated by assigning the first transmitter the frequency d_1 (provided it was not initially assigned this frequency) and the interference calculated. If it is lower than the previous assignment it is kept and the next frequency d_2 is tried (provided it was not assigned initially). This is continued until all frequencies have been assigned and tested. At this point the second transmitter is assigned the frequency d_1 (again provided it was not assigned this frequency initially), and the interference calculated and compared. This procedure continues until finally the last transmitter is tried with the frequency $d_{F_{fixed}}$ (provided it was not assigned initially).

The whole procedure is started again. Termination of the algorithm occurs when no improvement is made in the interference level on one complete pass through all the transmitters i.e. a local minimum has been located. Formally it can be described as follows:

Step 1. Randomly assign each transmitter i, $(i = 1, ..., N)$ a frequency, f_i, that is uniformly distributed from the set of available frequencies D. Call this assignment Z and calculate $E(Z)$, the interference as calculated from the cost function E. An alternative starting point is an assignment, $\mathbf{Z}$, obtained from a different technique e.g. a genetic algorithm.

Step 2. If $E(Z) = 0$, stop, the current assignment is feasible. Otherwise, set $i = 1$ and set $E_{best} = E(Z)$.

Step 3. Set $j = 1$.

Step 4. If $f_i = d_j$ goto Step 5. Otherwise generate an assignment Z' by assigning transmitter i the frequency d_j. Calculate $E(Z')$, if $E(Z') < E(Z)$ replace Z with Z' and set $E(Z) = E(Z')$.

Step 5. If $E(Z) = 0$, stop, the current assignment is feasible. If $j < F_{fixed}$, set $j = j + 1$ and goto Step 4.

Step 6. If $i < N$, set $i = i + 1$ and goto *Step 3*. If $i = N$ and $E(Z) < E_{best}$ goto Step 2. Otherwise, stop: The current assignment, Z, is locally minimal with cost E_{best}.

Quick Hill Climb

The *quick hill climb* procedure is obtained if the above procedure is terminated as soon as an improved solution has been found (with a maximum of a single pass through the 1-opt procedure i.e. with Steps 1-6 executed only once). The procedure starts with a randomly chosen transmitter and executes a complete pass of the transmitters starting from this point.

2.9 Hybrid Methods

All of the methods presented can be used to solve frequency assignment problems. However, FASoft has been developed so that methods can be combined in a variety of ways to form hybrids. In particular the output from one method can form the starting point of a different method. For example, the best assignment from a genetic algorithm could be refined using a local search algorithm such as simulated annealing and tabu search. Also, we have already discussed the use of heuristic crossover based on SA or TS, in genetic algorithms. This usually outperforms the standard crossover operators but has the drawback of being more computationally intensive.

The combination of sequential methods and SA or TS, for minimum span problems described in section 2.5, is a particular effective combination. However, this form of hybrid can also be used for fixed spectrum problems. Assume, for example, it was required to minimise the amount of interference in a network which has available $q + 1$ frequencies. The following procedure could be used to produce an assignment:

Step 1. Perform the SA or TS minimum span algorithm (section 2.5). This produces a zero violation assignment with span p. If $p \leq q$ then stop, this assignment is within the required span; otherwise goto Step 2.

Step 2. Starting from the solution found in Step 1, reassign each transmitter assigned a frequency $> q + 1$ with a random frequency $\leq q + 1$. Use the fixed spectrum simulated annealing algorithm (section 2.3.1) or the fixed spectrum tabu search algorithm (section 2.4.1) to minimise the number of constraint violations; after termination the best solution found is output as the best assignment with span, q. This step is automated in FASOFT in the flexible use of a start file (section 4.4).

This procedure generally produces an assignment with a smaller number of constraint violations than would have been obtained by minimising the number of constraint violations using only the fixed span available from the start (see section 5.4).

3 Lower Bounds and Clique Methods

For some problems only a relatively small number of transmitters limit the assignment to its lowest possible span. These transmitters can be used to obtain lower bounds and guide the frequency assignment process.

3.1 Lower Bounds

The formulation of the frequency assignment problem requires that each transmitter be assigned a frequency. The frequencies are grouped into a number of equally spaced discrete channels and each pair of potentially interfering transmitters must be separated by a given number of channels, in order to avoid interference. It is usually stated as a generalised graph colouring problem [9, 10, 19, 27, 34, 37, 38, 40, 52]. The transmitters are vertices of a graph and the frequencies are treated as a set of colours.

A *constraint graph* G is a finite, simple, undirected graph with each edge labelled with an integer in the set {0,1,...,L}. Let $V = V(G)$ denote the vertex set of G.

Let $T_0, T_1, ..., T_L$ be sets of non-negative integers with $0 \in T_0$ and $T_0 \subseteq T_1 \subseteq ... \subseteq T_L$.

Definition 1 *A* frequency assignment *in G is a mapping $f : V \to F$ (where F is a set of consecutive integers 1,2,...,K+1) such that if edge $(v_j v_k)$ is labelled i then*

$$|f(v_j) - f(v_k)| \notin T_i.$$

The elements of F are referred to as frequencies. K is referred to as the *span* of the assignment. In this paper we shall only consider the case when T_i is a set of consecutive integers {0,1,...,i}, although other cases can sometimes arise.

Definition 2 *If K is a minimum over all feasible assignments then the assignment is a* minimal assignment. *This minimal value of K is called the* minimal span, *and denoted* sp(G).

A fuller account of lower bounds for frequency assignment problems can be found in [41]. Further bounds for cellular problems can be found in [13]. We describe the simpler bounds from [41] and refer the reader to [41] for proofs.

Definition 3 *A* clique *in G is a maximal complete subgraph of G. If C_p is a clique in G with the label of each edge of the clique being at least p, then C_p is called a level-p clique.*

Proposition 1 *If G is a level-p clique C_p then*

$$sp(G) \geq (p+1)(|V(C_p)| - 1).$$

Construct a weighted complete graph G' with the vertices of G as vertices. The weights $w(v_i v_j)$ of the edges are defined as follows:

$$w(v_i v_j) = 0 \text{ if } (v_i v_j) \text{ is not an edge of } G,$$

$$w(v_i v_j) = s \text{ if the edge } (v_i v_j) \text{ has a label } s-1 \text{ in } G \ (s = 1, 2, ...).$$

Let $H(G')$ denote the length of the shortest Hamiltonian path in G'.

Proposition 2

$$sp(G) \geq H(G').$$

Let $S(G')$ denote the total weight of a minimal weight spanning tree in G'. A Hamiltonian path is a spanning tree and so we have $H(G') \geq S(G')$.

Proposition 3

$$sp(G) \geq S(G').$$

Proposition 3 is normally weaker than Proposition 2, but has the advantage that $S(G')$ can be calculated by a greedy algorithm [35].

Other bounds, which are relaxations of the bound given by Proposition 2 can sometimes be obtained by integer programming methods [25]. Currently they are never stronger than the bound of Proposition 2, but they are easier to calculate for cellular frequency assignment problems.

It is important to realise that the bounds of Proposition 3 and Proposition 2 may be very weak when applied to the whole constraint graph. Often they give a lower bound of 0. Typically they are applied to a level-p clique for some p or to a subgraph obtained from such a clique by adding a small number of additional vertices. As a valid assignment of

G gives a valid assignment of any subgraph of G, the minimal span of G cannot be less than the minimal span of a subgraph G' of G [41].

A number of distinct types of frequency assignment problem arise in practice. There may be a large number of low powered, relatively close transmitters, possibly arranged in a cellular system. There may be one transmitter per cell or there may be a demand vector describing the number of transmitters located at the centre of each cell. A second type of problem is concerned with a network of fixed links, typically of length 20km., with directional antennae. A third type of problem consists of a small number of high powered transmitters, with gaps in coverage filled by a larger number of low powered transmitters. All of these problems can be represented by the formulation in definition 1. However, in the case of a regular lattice of hexagonal cells with one transmitter per cell it may not be sensible to work with a constraint graph alone, and ignore information on the regularity of the geometry. Methods more appropriate to this case can be found in [30, 29]. On the other hand, when there is more than one transmitter per cell and the number of transmitters per cell is not constant, we shall see that the methods described here may be the more appropriate.

It is known from the theory of graph colouring that there exist constraint graphs with no large cliques (in fact with no triangles) with arbitrarily large span [11, 33]. However, in a typical frequency assignment problem of one of the types described above, large cliques tend to occur. Such cliques, and related structures which we shall refer to as "near cliques", may or may not determine the minimal span of the constraint graph.

3.2 Heuristic Algorithms for Frequency Assignment

FASoft [22, 46] is capable of finding good frequency assignments by a number of heuristic and other methods. Problems which are sufficiently small can be solved exactly by backtracking or forward checking [36] with a number of different orderings of transmitters possible (section 2.2). Assignments can always be found quickly with one of a number of different sequential (i.e. greedy) algorithms (section 2.1). The orderings of transmitters and of frequency channels proposed by Hale [20] are incorporated. Better assignments can usually be found by heuristic methods. The methods incorporated include a number of variations of simulated annealing (section 2.3 and 2.5), tabu search (section 2.4 and 2.5) and genetic algorithms (section 2.6 and 2.7).

A key feature we will need for the approach presented here is the ability to specify a starting assignment. This may either be complete, with all transmitters assigned a frequency, or partial. Additionally, transmitters which have been assigned a frequency may have the frequency fixed. One advantage of this facility is that the system can be interrupted at any time and the best assignment found so far is stored. This assignment can then be used as a starting assignment for the same or a different method. Thus hybrids of the various methods are easily constructed (section 2.9). Our concern here is with a different use of the starting assignment facility. Subgraphs of the constraint graph can be assigned first. The assigned frequencies are then fixed and the assignment is extended to an assignment of the full constraint graph. We shall see that in some circumstances a much better assignment can be found using this approach than is possible using any of the heuristics directly.

3.3 Using Cliques and "Near Cliques"

We use the largest level-p clique for some p as a starting configuration. The clique is assigned using FASOFT, the assignment is fixed and an attempt is made to extend the assignment to the full constraint graph G. If the span of the assignment of G is not close to the span of the clique, vertices are successively added to the clique to create a "near clique" and the process is repeated. Let t denote a threshold value, representing the maximum acceptable difference between the span of the assignment of G and the span of the subgraph assignment. Let $U(X)$ denote the subgraph of G induced by a set X of vertices. Let $L(C)$ denote the best lower bound found from a subgraph C and let $s(C)$ denote the span of the assignment of C. Then we have the procedure described in Figure 4.

We now amplify the description of the various steps in the procedure. In order to carry out step 1 we find the maximum level-p clique in the constraint graph for each $p \in \{0, 1, \ldots, L\}$. FASOFT contains an implementation of the maximum clique algorithm due to Carraghan and Pardalos [6]. Although the maximum clique problem is NP-complete (see [14] for a heuristic algorithm), we have found that the Carraghan and Pardalos algorithm works well for typical frequency assignment problems. For example, the largest level-p cliques for $p \in \{0, 1, 2, 3\}$ in a 726 transmitter problem can each be found in at most a few minutes on a 133MHz Pentium PC, although it is important that a good ordering of vertices is used or the run time can be excessive (section 3.4).

The value of p such that the largest level-p clique should be chosen for C_0 is usually best selected by choosing the level-p clique that gives the best lower bound available. In order to do this, and to compute lower bounds later in the procedure, either Proposition 3 or Proposition 2 can be used. An implementation of Prim's Algorithm [35] allows the bound of Proposition 3 to be found immediately. The more accurate bound of Proposition 2 requires the solution of an open symmetric travelling salesman problem. Let C' be the weighted graph obtained from C as described after Proposition 1. We have used the method of Volgenant and Jonker [50] applied to C'. In most cases we have studied, the

```
Choose a starting clique C_0;                                - -STEP 1
OUTPUT lower bound L(C_0);
Assign clique C_0 heuristically;                             - -STEP 2
Fix the assignment of C_0 and
extend it heuristically to an assignment of G;               - -STEP 3
OUTPUT s(G);
σ_old := s(G) − s(C_0);
C := C_0;
σ_new := s(G) − s(C);
WHILE σ_new > t AND σ_new ≤ σ_old LOOP
    σ_old := σ_new;
    C := U(V(C) ∪ W) where W is a set of
    one or more vertices selected to add to C;               - -STEP 4
    Assign C heuristically;                                  - -STEP 5
    OUTPUT lower bound L(C);
    Fix the assignment of C and
    extend it heuristically to an assignment of G;           - -STEP 6
    OUTPUT s(G);
    σ_new := s(G) − s(C);
END LOOP;
OUTPUT best assignment of G;
```

Figure 4: The procedure for improving heuristic results using cliques

Volgenant and Jonker method either gives the exact bound of Proposition 2, or a narrow range within which the bound lies. Thus in this latter case it gives a bound which may be slightly weaker than the bound of Proposition 2.

As already mentioned FASOFT contains all the necessary facilities for carrying out steps 2, 3, 5 and 6. For step 4 we use the following method of adding vertices to C_0 or to C, in order to create a subgraph which we shall refer to as a *near clique*.

Definition 4 *Let Q be a subgraph of G. Consider a fixed domain F of frequencies and let $f : V(Q) \to F$ be an assignment of Q. Let this assignment of Q be fixed. If $v \in V(G) \setminus V(Q)$ then certain frequencies in F cannot be assigned to v because of constraints represented by edges vw with $w \in Q$. We refer to the set of frequencies that* can *be assigned to v as the* reduced frequency domain *of v (with respect to the assignment of the subgraph).*

Given the clique C_0 (or current near clique C) determine the reduced frequency domain of each vertex. Choose the vertex (or vertices) with the smallest reduced frequency domain and add it (or them) to the current clique or near clique.

Ideally the procedure terminates with $\sigma = 0$ and $s(G) = L(C)$. Sometimes $s(G)$ becomes larger than can be obtained directly using heuristic methods. In this case the procedure is not useful for determining an assignment, but may still lead to a useful lower bound. If the clique is not a significant determinant of the span, fixing the assignment of the clique may reduce the freedom of the heuristic algorithm with no consequent benefit.

In section 5.3 we will give some examples in which optimal assignments can be found using this approach, where they cannot be obtained by a direct application of any heuristic we have tried.

3.4 The Use of Orderings for the Maximum Clique Problem

The *maximum clique problem* is to find the largest maximal complete subgraph of G. The maximum clique problem has been shown to be NP-complete [6]. Carraghan and Pardalos proposed an exact algorithm [6]. They suggested an initial ordering of the vertices of the graph, but recommended using the ordering only for graphs with an edge density greater than 40%. The graphs used in their evaluation contained up to 3000 vertices and edge densities ranging from 10% to 90%. All graphs were randomly generated.

Here, we use the Carraghan and Pardalos algorithm and investigate the effect of different initial orderings on the clique produced and the run time of the algorithm. In contrast to [6] the algorithm was tested on data generated from real frequency assignment problems [49].

Different initial orderings have been used in the sequential (greedy) assignment of radio frequencies for some time and, generally, different orderings can produce results which vary considerably. We will use the same orderings as those presented by Hale [20] and already introduced earlier (section 2.1):

LF1, LF2, SL, GLF1, GLF2, GSL

The ordering used by Carraghan and Pardalos [6] was *generalised smallest first*, which is GSL in reverse order. We will consider twelve different initial orderings, the six given above together with the reverse orderings in each case. All the orderings have been compared with each other and also with the use of no ordering.

For the implementation of any exhaustive search the notion of *failing* [36] is important: *Failing* occurs when the algorithm is unable to include any more vertices and therefore *prunes* the search tree. This removes the last included vertex from the clique.

In any exhaustive search it is desirable that the algorithm should *fail* as early as possible, because then the pruning is more effective and a large part of the search tree can be deleted. The aim of a maximum clique algorithm is to find a *largest* clique. It is beneficial to first check vertices having few edges, as the algorithm is then able to *fail* early. Therefore, it can be expected that reverse ordering, where vertices with smaller degrees are listed first, would outperform non-reverse ordering in terms of run time.

We have found in our tests [49], that the type of ordering is important. It not only influences the run time but also the nature of the clique. Our results agree with Carraghan and Pardalos [6] in the choice of GSL with reverse ordering. Ordering with GSL in reverse has been shown to be the most suitable for large problems. Results have confirmed that using a reverse order outperforms using no ordering and a non reverse ordering.

For small problems however, the differences are not so clear. Different orderings can result in a different clique of same size being found. It can be beneficial, to use LF1 or GLF1 to obtain the clique which has the maximum number of constraints (i.e. edges) with the remainder of the graph. It can be expected, that such a clique is more beneficial in the clique approach, as more constraints with the remainder of the graph should make it easier to extend the clique to the full problem. However, the process of finding the largest clique with LF1 or GLF1 will have a longer run time than with any of the reverse order methods.

A more detailed description of the process and the full results regarding the maximum clique algorithm can be found in [49].

4 Advances to the Software

In the second year, several significant improvements to the frequency assignment software FASoft have been incorporated. Although the first year was mainly concerned with the development of the heuristics used for frequency assignment, ongoing research has meant a continuing updating of FASoft and the development of other miscellaneous support routines to enable work on special types of problems. This continual updating has meant that more time than originally planned was used for development, however this has enabled the research to be performed in a much more effective way. The breakthrough in obtaining a high proportion of optimal minimum span results (section 5.2 and 5.3) would not have been otherwise possible.

The four most significant changes to the software are:

1. The development of a Windows interface.
2. A stronger focus on the assignment of minimum span problems as opposed to fixed spectrum problems.
3. The incorporation of clique oriented routines.
4. More flexible use of the start file.

4.1 The User Interface

The main program of FASoft and the 4 clique orientated routines CLIQUE1, CLIQUE2, CLIQUE3 and COMA2TSP have been converted into a Windows interface.

The Windows interface enables the user to become familiar with the system much more easily. Also, the FASoft user interface makes it possible to provide the user with appropriate help at the click of a button.

The Windows interface requires either Windows 95 or Windows NT or alternatively DOS with Windows 3.11 and the 32 bit extension Win32s. The command line version will continue to be the main interface for UNIX based systems.

The Windows interface is developed using Borland Delphi, with 'C' being the programming language for the command line interface and the program code for the heuristic routines. On UNIX based systems the software was developed using 'C' only.

4.2 Minimum Span Frequency Assignments

FASoft was initially developed from research on fixed spectrum problems. This meant that the work on minimum span problems involved a high amount of user interaction, and executing a large number of runs. Furthermore, the algorithms themselves were optimised towards fixed spectrum problems and therefore the results obtained were sometimes unfavourable when compared with alternative techniques that have appeared in the literature. These two shortcomings have now been eliminated by incorporating into FASoft techniques developed from those presented by Costa [8].

Firstly, two new methods have been created, using simulated annealing (SA) and tabu search (TS) respectively. Apart from the chosen heuristic, the methods are identical in that they incorporate an efficient and easy to use minimum span frequency assignment algorithm. Both methods start with a given zero violation assignment. This start assignment can either be specified using the *start file* capability of the system (section 4.4) or found directly using one or a combination of sequential algorithms. The system then reassigns all transmitters which are assigned to the highest used frequency, to a randomly chosen smaller frequency. Thus, it deletes the highest frequency from the frequencies available for use. This step will generally introduce a small number of violations, but preserves some features present in the original assignment. The next step then involves using one of the heuristic methods (SA or TS) to find a valid assignment with no violations using the reduced frequency domain. Once this is achieved, the assignment found is taken as the start assignment and the process starts again. The loop terminates, when the heuristic is unable to reduce the number of violations to zero. Therefore, the assignment taken as the start assignment in the previous iteration contains the best zero violation assignment found (section 2.5).

Secondly, the heuristics were optimised for performing this procedure. As already mentioned a valid assignment is reduced by one frequency and this generally introduces an assignment with a few violations. It is, of course, not very efficient to change frequencies for transmitters which are not involved in those violations. We therefore limit the choice of transmitters to only those, which violate the assignment. This not only speeds up the assignment process, but also increases the chances that the good features already present in the starting assignment remain intact. One of the heuristics, tabu search, can further be speed optimised by including only a randomly chosen subset of the neighbourhood.

These two major changes have considerably enhanced the system into an even more effective and easy to use frequency assignment system.

4.3 Working with Cliques

For some problems only a relatively small number of transmitters limit the assignment to its lowest possible span. It can be difficult to assign these problems when using FASOFT on the whole problem. This is because, due to their random nature, the heuristics employed in FASOFT do not necessarily concentrate on the transmitters which are difficult to assign.

Generally, the difficult transmitters form a clique (fully connected subgraph). It is therefore important to be able to extract cliques and attempt to assign the transmitters contained in the clique, before trying to solve the whole problem (section 3.3).

We have implemented an exact clique finding algorithm, based on the work of Carraghan and Pardalos [6], in a separate routine CLIQUE1, which also generates a constraint matrix containing only the transmitters contained in the clique. The problem of finding a largest clique is NP-complete [6] and therefore can be impractical, especially for large problems (more than 400 transmitters). It is normally beneficial to order the transmitters in the same way as used for sequential algorithms to speed up the search process. This is now implemented in the clique finding algorithm. With the best ordering chosen, the procedure has found the maximum clique in all problems attempted, with up to 726 transmitters. Run times are normally a matter of minutes.

We have also developed two more executables (CLIQUE2 and CLIQUE3) to enhance the effectiveness of FASOFT. The procedure used is as follows:

1. Use CLIQUE1 to find the largest clique and to generate a sub constraint matrix containing only the transmitters in the clique.
2. The transmitters in the sub constraint matrix should then be assigned. This can be done using FASOFT.
3. CLIQUE2 can then be used to create a start file, where all the transmitters in the clique are fixed to their frequency in the previous assignment (from step 2).
4. FASOFT can then be used to assign the complete problem.
5. If the clique needs to be extended, this can be done with a simple file manipulation by hand. CLIQUE3 can then be used to create the sub constraint matrix containing the additional transmitters. It is important at this stage, that the clique is extended sensibly. Frequency domains can help in this context. Transmitters, which have only a limited number of frequencies available without causing constraint violations, should be included first, forming a "near clique".

6. Continue from step 2, until an acceptable assignment is found.

It should be noted that the difficult transmitters generally provide good lower bounds for the frequency assignment. Therefore, in step 2 the assignment should try to match the best lower bound. CLIQUE1 and CLIQUE3 calculate automatically the minimum spanning tree bound [41], using Prim's algorithm. Other lower bound techniques can also be used at this stage.

Solutions for travelling salesman problems often provide good lower bounds for the frequency assignment problem. There exists a TSP algorithm, developed by Volgenant and Jonker [50], in the public domain[2]. This is a DOS executable called TSP1. We have also written a program called COMA2TSP, which converts a constraint matrix into a format that can then be read by this exact TSP algorithm. It should be noted, that a TSP algorithm can only in very special cases produce a valid frequency assignment. However the solution obtained by an (exact) TSP algorithm is always a lower bound to the minimum span required for a zero violation frequency assignment.

An additional improvement to the system is the incorporation of arc consistency. Arc consistency is the reduction of the frequency domains for individual transmitters, so that frequencies which would result in a constraint violation (i.e. with transmitters with fixed frequencies) are eliminated. Arc consistency is therefore important when there are some transmitter frequencies fixed, as in step 5 above. When arc consistency is performed, the transmitter index and the size of domain is printed out on the screen and this information is also stored in a file, having a default filename consisting of the name of the constraint matrix with 'dsz' as the extension. This is useful as the size of domain is one feature used to determine which transmitters should be added to a clique if step 5 (clique extension) is used.

4.4 Starting Assignments

The starting assignment has increased in importance. It is used extensively when working with cliques and can also be used in the loop for minimum span problems (section 4.2). There are two different formats for the start file; when it is the same format as the assignment file there have been the following changes:

- Transmitter frequencies can now be fixed in this format.

[2]ftp://www.mathematik.uni-kl.de/pub/Math/ORSEP/VOLGENAN.ZIP

- It is possible to use domains, which have a lower cardinality than the number of frequencies given in the start assignment file. This enables the user to find lower span assignments then the one presented in the start file (section 2.9). The transmitters, which have initial frequencies outside of the domain, will be assigned to a random starting frequency and a warning will be printed.
- When not all transmitters are contained in the start file a warning is given. However, for those methods which require a complete start file an error is given and the procedure terminates.

It seems reasonable to propose that globally difficult configurations within the constraint graph should be assigned first; they cannot be satisfactorily handled late in the assignment process. The basic idea we have described in section 3.3 is that the configurations (i.e. subgraphs) used to obtain the lower bounds described in section 3.1 and [41] may be precisely the configurations that should be assigned first. It is important that the configurations correspond to strong lower bounds. If configurations are used that are not determinants of the span of the assignment then the configuration could have negative benefit. The freedom available to the heuristic algorithm could be reduced without attacking the real global difficulty in the problem.

Once a good assignment of a good configuration is found (using sequential methods, or heuristic methods or if the problem is small enough exact methods) the assignment is used as the starting point for any of the heuristic algorithms. We have found that starting configurations which are cliques or built from cliques produce good results. Details of how these starting configurations are generated and how they are extended to an assignment of the full problem were given in section 3.3 and are also described in [42].

The starting assignment can also provide valuable guidance for fixed spectrum assignments. Obviously, when starting from a good initial configuration, the algorithm can be expected to find a good sub-optimal solution more easily. It is for example possible, to take the best assignment for a zero violation minimum span model and then use this as a starting configuration for the fixed spectrum model, occupying a smaller span. The frequencies which were used in the minimum span assignment but which are not present in the frequency domain will then be reassigned randomly an alternative frequency. This will obviously introduce initial assignment violations, but the results when using this as a starting configuration can be much better (section 5.4).

5 Frequency Assignment Results

In this section we present the results obtained on a set of test data with simulated annealing and tabu search. We do not provide results for genetic algorithms or hill climbing, as their performance is poor relative to simulated annealing and tabu search.

5.1 Test Data

Firstly we want to introduce some test data, which we use throughout this results section. Most of these files were already used in the Final Report Year 1 [44] or the Year 2 Interim Report [45].

Test95

This is a 95 transmitter example which is typical of a military radio links problem. The edge density is 25%. This constraint matrix contains variable co-site constraints, which can be set individually for each run. Typical values used here for the co-site constraints are 4, 5 or 10.

Test726

This is a 726 transmitter example from the same test set as Test95. It also contains variable co-site values. We only examined Test726 with a co-site value of 4.

Philadelphia test files, Ph1 - Ph6

This example is based around the area around Philadelphia, PA and initially was presented in [3]. It is a relatively large example, but it has the advantage that it can be described easily. The problem has been dealt with previously by several authors, particularly by Gamst [13] and it has been proposed to use this test set as a benchmark for heuristic algorithms [42]. The Philadelphia problem contains of 21 cells located in a hexagonal geometry as shown in Figure 5.

A requirement vector M is used to describe the demand for frequencies in each cell. Thus, m_i denotes the number of channels required by cell i. There are three different requirement vectors:

$$M_1 = (20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20, 20)$$

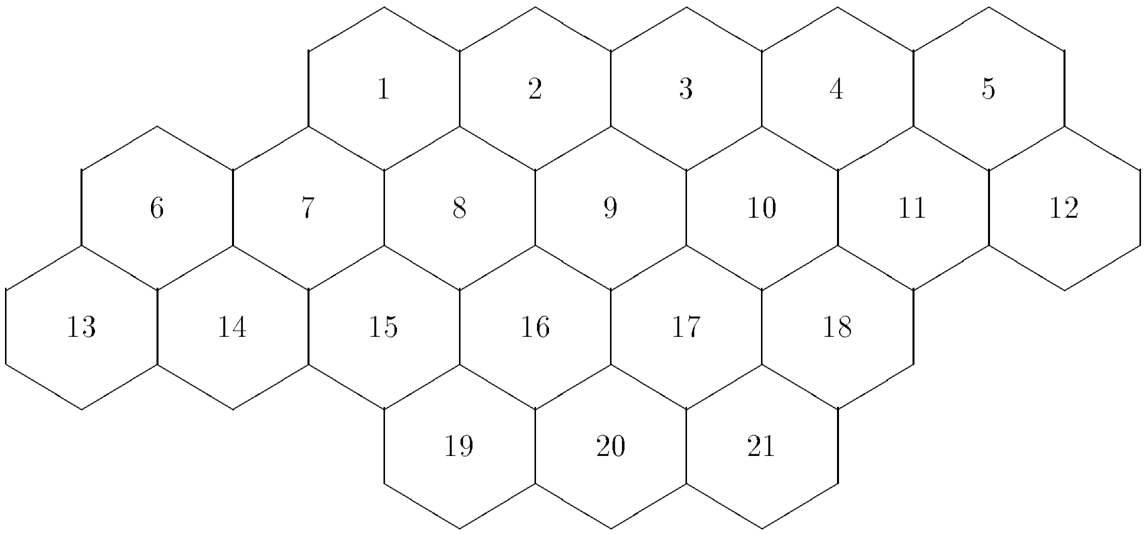

Figure 5: The cellular geometry of the Philadelphia problem

$$M_2 = (5,5,5,8,12,25,30,25,30,40,40,45,20,30,25,15,15,30,20,20,25)$$

$$M_3 = (8,25,8,8,8,15,18,52,77,28,13,15,31,15,36,57,28,8,10,13,8)$$

Transmitters are considered to be located at cell centres and the distance between transmitters in adjacent cells is taken to be 1. Denote by d_k the smallest distance between transmitters which can use frequencies with a separation of k channels. In this example $d_0 = \sqrt{12}$ or $\sqrt{7}$, $d_1 = \sqrt{3}$, $d_2 = d_3 = d_4 = 1$ and $d_5 = 0$, so transmitters within the same cell (co-sited transmitters) must be separated by at least 5 channels.

The three demand vectors together with two different values of d_0 result in six variations of the problem.

Ph1 uses M_1 and $d_0 = \sqrt{7}$
Ph2 uses M_1 and $d_0 = \sqrt{12}$
Ph3 uses M_2 and $d_0 = \sqrt{7}$
Ph4 uses M_2 and $d_0 = \sqrt{12}$
Ph5 uses M_3 and $d_0 = \sqrt{7}$
Ph6 uses M_3 and $d_0 = \sqrt{12}$

Out of these examples Ph6 is most often quoted in the literature [3, 13, 22, 25, 28, 39, 42, 51].

We were also interested in how scaling a problem effects the outcome of the frequency assignment process. We used Ph6 as a starting point and scaled the demand vector M_3. The scaling factors used were 0.2, 0.4, 0.6 and 0.8 to obtain the smaller problems Ph62, Ph64, Ph66 and Ph68. We also created larger problems (Ph6x2 and Ph6x4) by using a scaling factor of 2 and 4. With the larger problems it is also important to observe if the heuristics are still effective because of the increased complexity. Ph6x4 for example contains 1924 transmitters and requires 1,568,246 constraints to be solved. Ph6x4 is the largest problem investigated in this report.

Bt_58_4

This example consists of an irregular network with 58 cells (base stations) each of which has a traffic demand equal to 4 (i.e. 232 transmitters need to be assigned, 4 transmitters per cell). There are co-channel and adjacent channel constraints (2748 in total, giving a constraint density of 10.2%). This example was used by Lochtie and Mehler to test a neural net approach [31].

5.2 Minimum Span Results

The introduction of two new methods as described in section 4.2 has not only made the system more user friendly when working with minimum span problems, but it also improved the effectiveness of the assignment process. In particular, the improvement of the new tabu search loop procedure compared to the *standard* tabu search has been significant.

Bt_58_4

Firstly we present some results for Bt_58_4. A neural network approach was used in [31] and found a zero violation solution with a span of 28. An assignment with a span of 15 can be obtained with sequential assignments, using several combinations available. Proposition 1 (section 3.1) gives a lower bound of 15. Therefore the assignment with span 15, obtained with sequential algorithms, is optimal.

Test726

Secondly we want to present results on a far more difficult problem, Test726. The best bound has span 181 and can be obtained using linear programming methods [2]. The best assignment we achieved has span 229 and is still some way from the best bound.

The best sequential result for Test726 has span 248 and can be improved to 246 with the default SA loop and to 244 with the default TS loop (section 2.5 and [46]). We continued to try to improve the result by using TS. If executed for 10000 runs a 236 assignment can be found. Alternatively increasing the 'number of frozen iterations' to 4000, TS obtained a span of 238 with the default of 10 runs and the best known span of 229 with 4000 runs. A summary of the results can be found in Table 1.

lower bound	181
SEQ	248
SA (default)	246
TS (default)	244
TS with 10000 runs	236
TS with $frozen = 4000$	238
TS with $frozen = 4000$ and 4000 runs	229

Table 1: The results of Test726

A comparison on how TS with the two different values for the 'number of frozen iterations' decreases the span is shown in Figure 6. *Run A* uses 10000 runs and 1000 frozen iterations. *Fail A* is when it unsuccessfully tries to decrease the span by one to 235. *Run B* and *Fail B* are the plots for 4000 runs performing 4000 frozen iterations. The y-axis is the span and the x-axis is the number of assignments tested, to give an indication of the run time.

In Figure 7 the same graphs are plotted, but now in linear scale for the x-axis (number of assignments tested). This gives a much better indication on the time it actually takes to perform such long searches. One dash ($5 \cdot 10^9$ assignments tested) is printed on the x-axis after about 8 days of runtime on an Alpha 500/333 processor. Run A took about 14 hours to find the 236 assignment, failing to improve on it after 16 days. Run B found the 229 span assignment within 21 days and took another 21 days trying to improve on it.

It can be seen that Test726 is quite a large and complex problem. It is difficult to obtain a good bound and it is very time consuming to achieve a good minimum span assignment. Clique methods (section 3.3) have so far been unsuccessful in obtaining a better assignment.

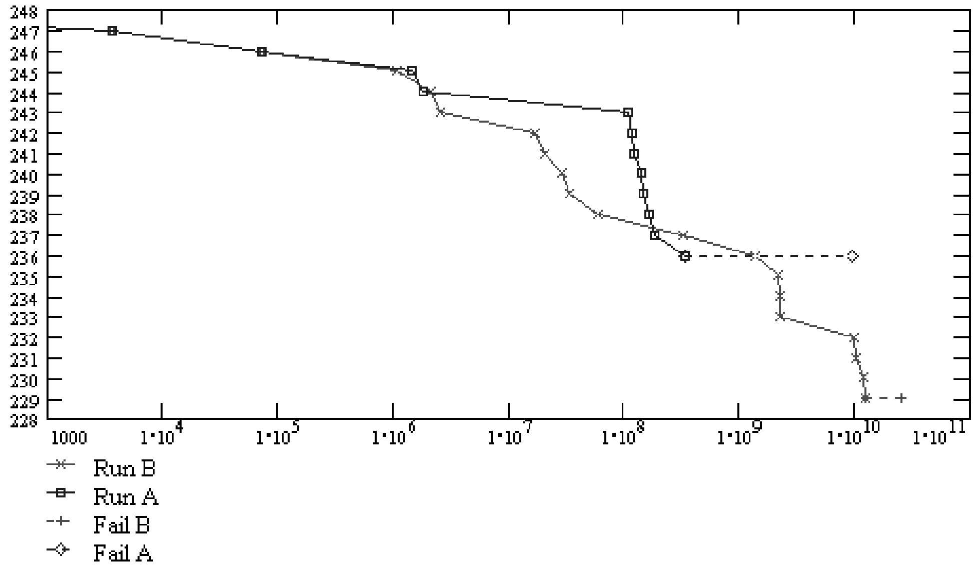

Figure 6: Test726 results in logarithmic scale

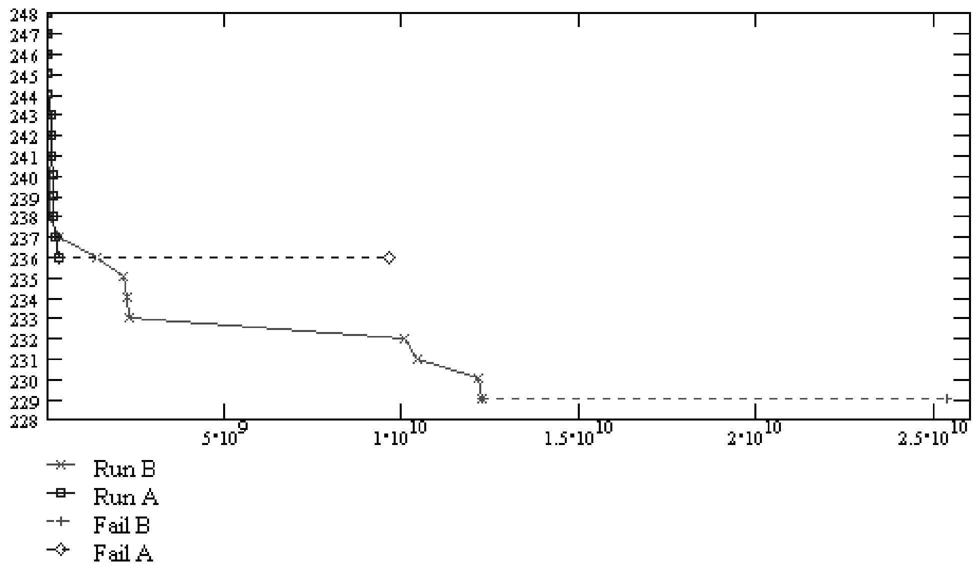

Figure 7: Test726 results in linear scale

5.3 Minimum Span Results using Cliques

Test95

The optimal results for Test95, with co-site values of 10 and 5, can be achieved without cliques. To obtain the bounds however, cliques are needed. Using co-site 10, the span 60 bound is the clique bound (Proposition 1, section 3.3) of 7 co-sited transmitters, which form a level-9 clique. An assignment can be easily obtained with sequential algorithms. The problem is a bit more difficult with co-site 5. The bound, based on Proposition 2 is 48 [44]. At the end of Year 1 [44] we could achieve the optimal assignment only with clique methods. Now the tabu search loop starting from the best sequential assignment of span 54 is able to obtain a span 48 assignment without using cliques.

The bound of 47 for Test95, with co-site value 4 can also be obtained from Proposition 2 when using the TSP algorithm of Volgenant and Jonker [50] on a level-1 clique C_1 with 16 vertices. Thus, we select C_1 and find an assignment of span 47 using one of our heuristic algorithms. However, this assignment, when the frequencies are fixed, does not extend to an assignment of the full constraint graph with span 47. In fact, the assignment of C_1 extends to an assignment with span 50. If we add the vertex of smallest reduced frequency domain to C_1 we obtain a "near clique" C_1' with 17 vertices. Again the Volgenant and Jonker algorithm gives a lower bound for the minimal span of 47 for C_1'. An assignment of C_1' with span 47 can be obtained heuristically. Now the assignment of the near clique C_1', with all frequencies fixed, does extend heuristically to give an assignment of span 47 for the full 95 transmitter problem. This assignment is thus of minimal span.

A summary of the results with Test95 can be found in Table 2.

co-site	4	5	10
bound	47	48	60
SEQ	51	54	*60*
SA	49	52	-
TS	49	*48*	-
with cliques	*47*	-	-

Table 2: The results of Test95 with 3 different values for the co-site separation

Philadelphia test files

A summary of the results can be found in Table 3. Details on how these assignments were achieved are given below, starting with Ph6 to Ph1. This order was chosen because Ph6, as already stated, is the most used Philadelphia example in the literature.

	Ph1	Ph2	Ph3	Ph4	Ph5	Ph6
lower bound	178	239	252	257	426	426
SEQ	230	250	268	284	475	447
loop SA	200	*239*	259	260	438	428
loop TS	188	240	257	269	429	428
with cliques	179	-	*252*	*257*	*426*	*426*

Table 3: The results of the six Philadelphia examples

Ph6

The problem was described by Anderson [3] in 1973 and was also dealt with by Sivarajan, McEliece and Ketchum [39]. Both papers quote a lower bound for the span of 413 and the best assignment of span 444 is quoted in [13]. R. Leese (private communication) found an assignment of span 427 by hand. An improved lower bound was first obtained by Janssen and Kilakos [25] using integer programming methods. They showed that $sp(G) \geq 426$. The result is more easily described using Proposition 2 and the Hamiltonian path that gives the bound of the Proposition. Let G_s denote the subgraph of the constraint graph induced by vertices representing transmitters in cells 2,3,8,9,10,16,17 and consider the graph G'_s. We will show that $H(G'_s) \geq 426$. G'_s is complete with no edges of weight 0. It has 275 vertices so the span is at least 274. There are no edges of weight 1 incident with any vertex from cell 9. Thus the frequencies before and after any frequency assigned to a vertex in cell 9 must be unoccupied. These unassigned frequencies are distinct if the co-site value is at least 3. The minimum value of $H(G'_s)$ occurs when the first and last frequencies are assigned to vertices from cell 9. Thus $H(G'_s) \geq 274+75\times 2+2\times 1 = 426$. Thus the minimal span for the complete problem is not less than 426. The same result holds if the co-site value is 3 or 4 instead of 5 (i.e. $d_4 = 0$ or $d_3 = d_4 = 0$).

In attempting to find a good assignment for this problem we proceed as follows. The maximum clique algorithm [6] finds a maximal level-1 clique C_1 whose vertices correspond to the transmitters in cells 8,9,16. (Clearly the largest level-2 and level-3 cliques are the same and the largest level-4 clique consists of the 77 vertices corresponding to cell 9. Proposition 2 gives a lower bound of $76 \times 5 = 380$). An assignment of C_1 of span 414 can be found heuristically. If an attempt is made to extend this assignment the best span found has span 451. A near clique can be built from C_1 by adding successively the vertices with smallest reduced frequency domain, which turn out to be the vertices corresponding to cells 17, 2 and 15. An assignment of this 275 vertex near clique of span 426 can be found heuristically. This can then be extended using a heuristic algorithm to an assignment of the complete problem of span 426, which we know is minimal. The full assignment is shown in [42]. Typically a direct application of the heuristic techniques in FASoft without consideration of cliques gives an assignment of span 428 (which is itself an improvement on previously published results).

Ph5

Here the method we have used for Ph6 to find the lower bound remains valid, so the span is again at least 426. As the constraints have been weakened, the same assignment is still valid and we again have an assignment with a minimal span of 426. In either case, it is possible to reduce the co-site value from 5 to 4 or 3 and, as previously mentioned, the same bound still holds. These changes represent weakened conditions so the same assignment is still valid and the minimal span is again 426.

It should be noted however, that as we can see in Table 3 both simulated annealing and tabu search when applied directly to Ph5 obtain a higher span assignment than with Ph6, which is more constraint problem.

Ph4

The level-0 clique corresponding to cells 2,3,4,8,9,10,11,16,17,18,20,21 gives a lower bound $sp(G) \geq 257$ with Proposition 1. However, it turns out that in this example much of the difficulty is with the cell of highest weight (cell 12), and it is better to start with the same subgraph as in Ph3 below rather than the level-0 clique. An assignment of the subgraph of span 255 can be found easily. This can be fixed and extended to an assignment of span 259 for the full constraint graph. Finally we found that if all fixed transmitters are unfixed at this stage the span of the assignment can be reduced to 257, which is also the lower bound. The full assignment is shown in [42]. The best assignment we were able to find without clique methods had span 260.

Ph3

In this example, our methods show that the minimal span is exactly 252. Let G_s denote the subgraph of the constraint graph induced by vertices representing transmitters in cells 4,5,10,11,12,18 and consider the graph G'_s. The same method as used with Ph6, with the central cell 11 in place of cell 9 shows that $H(G'_s) \geq (8 + 12 + 40 + 40 + 45 + 30 - 1) + 38 \times 2 + 2 \times 1 = 252$. Note that the same argument applied to cells 3,4,9,10,11,17,18 would give $sp(G) \geq 245$. Because these two bounds are fairly close, it turns out to be beneficial to start with the union of the two sets of transmitters (those in cells 3,4,5,9,10,11,12,16,17,18), instead of a single clique. For this subgraph, one of the heuristics will find an assignment of span 253, but cannot find an assignment of span 252, even if left to run for two days. In this case a manual intervention is required. The method used to find the lower bound shows that the first and last frequency must be assigned to a transmitter in cell 11. In our first attempt the assignment did not start in cell 11. This was remedied by reducing the frequency assigned to each transmitter by a fixed value (in order to start the assignment in cell 11), fixing the frequencies of the transmitters in the allowable range and reassigning all transmitters outside of the allowable range. An assignment of span 252 was obtained. If the assignment is fixed it is then easy to extend the assignment of span 252 to the complete constraint graph. The full assignment is shown in [42]. The best assignment that we were able to find without clique methods had span 257.

Ph2

The optimal assignment of span 239 (level-0 clique) can be obtained using the simulated annealing loop, starting with the best sequential assignment.

Ph1

Here it is quite easy to obtain a bound of span 177 [25]. It is possible to improve this bound by one. The bound of 177 is effectively a Hamiltonian Path bound (Proposition 2), for example using cells 2,3,8,9,10,16,17. It can then be proven that it is not possible to add all the remaining transmitters to achieve a valid assignment. We believe that the bound is actually 179 and not 178, but this is very hard to prove.

An assignment of span 179 can be easily constructed by hand, however to achieve the assignment of span 179 using heuristic methods is far more difficult. All 20 transmitters from cell 9 have to be assigned and fixed to frequencies $1, 10, \ldots, 172$. This assignment can then be extended to the cells 2,3,8,9,10,16,17. With this assignment fixed the full problem can then be assigned to the span of 179.

Ph6, scaled

For the scaled versions of Ph6, the results can be found in Table 4. With the scaled down examples we have not tried to obtain optimal results using cliques, as Ph62 and Ph66 already provided optimal results with the heuristics alone and the assignments for Ph64 and Ph68 were only one off the best possible span.

Both simulated annealing and tabu search provide already very good assignments for Ph6x2 and Ph6x4. The optimum can be obtained when using cliques. The strategies used were similar to those already mentioned regarding Ph1 to Ph6.

Of course, the knowledge that Ph6x2 and Ph6x4 are scaled versions of Ph6 can also be used to produce assignments. Repeating twice the span 426 assignment for Ph6 with

	Ph62	Ph64	Ph66	Ph68	Ph6x2	Ph6x4
lower bound	82	169	254	341	855	1713
SEQ	88	183	269	359	894	1729
loop SA	*82*	171	*254*	342	858	1724
loop TS	*82*	170	*254*	343	858	1724
with cliques	-		-		*855*	*1713*

Table 4: The results of the scaled Philadelphia examples

a co-site gap of 5 is immediately a valid span 857 assignment for Ph6x2. Similarly an assignment of span 1719 for Ph6x4 can be achieved by repeating the Ph6 assignment 4 times. Alternatively the 855 span assignment of Ph6x2 can be used twice to produce a 1715 assignment for Ph6x4. However no such previous knowledge was used to produce the results in Table 4.

5.4 Fixed Spectrum Results

Sometimes, when the available number of frequencies is limited to a certain span, lower than the minimum bound required by the problem, the heuristics can take the fixed spectrum as input and try to minimise the violations or errors (section 2.3.1).

Here we want to minimise the errors, which gives the sum of the amount of each violation, as opposed to simply the number of violations. The errors resemble more closely an interference measure as required by a radio engineer.

We used the three test files Test95, Test726 and Ph6 to obtain results for data of varying size and difficulty. Previously when minimising the errors or violations the starting configuration used was always a random assignment [44]. This is also the default in FASOFT. However this is not the only choice possible, for example [5] starts with an assignment with all transmitters assigned to the same frequency. A third possibility of a starting assignment is to use a known zero violation assignment of higher span and then randomly reassigning the transmitters having frequencies outside the fixed spectrum available. This is very similar to the minimum span approach.

The five different scenarios tested were:

1. Test95, co-site 5, best bound and best known assignment has span 48. The fixed spectrum was limited to a span of 39.

2. As (1.) above with a fixed spectrum of span 24.

3. Test726, co-site 4, best bound has span 181, best known assignment has span 229. The fixed spectrum was limited to a span of 174.

4. Ph6, best bound is 426 and best known assignment has span 426. The fixed spectrum was limited to a span of 399.

5. As (4.) above with a fixed span of 199.

In each scenario we used three different starting configurations: a random assignment, an assignment with span 0 (using a single frequency) and the best (available) span assignment. Simulated annealing and tabu search were used as the heuristics, with 2 different sets of parameters each. The first parameter set for SA and TS were the default parameters [46]. In the second parameter set for SA the starting temperature *t_start* was set to 0.1 to prevent SA from using high temperatures. In the second parameter set for TS the number of frozen iterations was increased to 4000, because of the large search space.

The results are summarised in Tables 5 to 9 and contain the errors of the best run and also the average over all 10 runs. (10 is the default parameter for number of runs). Also, in order to give an indication of the complexity of the problem and the different run times for SA and TS the number of assignments tested were also printed. In real processor times, the scenarios with Test95 take about 1 to 5 minutes on a 133MHz Pentium, whereas the runtime with Test726 may take 1 to 5 hours.

Starting Assignment		Random	One frequency	Best
SA default	Errors	18	18	16
	Average	19.8	20.3	20.1
	Tested	6 831 610	6 853 195	6 766 405
SA $t\_start = 0.1$	Errors	19	21	19
	Average	22.6	24.2	21.2
	Tested	5 877 430	5 702 585	5 715 532
TS default	Errors	19	17	18
	Average	21.1	21.0	20.0
	Tested	2 169 645	2 496 461	1 666 620
TS $frozen = 4000$	Errors	18	18	17
	Average	19.2	20.2	18.5
	Tested	7 377 035	6 675 700	6 488 749

Table 5: Test95, span 39

Starting Assignment		Random	One frequency	Best
SA default	Errors	92	94	93
	Average	96.4	98.9	97.9
	Tested	8 385 896	8 158 903	8 145 193
SA $t\_start = 0.1$	Errors	97	94	98
	Average	101.9	101.0	102.3
	Tested	7 193 457	7 199 585	7 243 124
TS default	Errors	92	90	95
	Average	98.1	96.3	98.9
	Tested	2 411 706	2 301 615	1 844 276
TS $frozen = 4000$	Errors	91	88	89
	Average	92.5	93.4	92.9
	Tested	9 639 700	6 923 947	8 155 315

Table 6: Test95, span 24

Starting Assignment		Random	One frequency	Best
SA default	Errors	352	377	359
	Average	377.3	380.8	377.1
	Tested	71 479 384	67 831 904	71 440 928
SA $t\_start = 0.1$	Errors	383	386	321
	Average	397.7	404.6	344.2
	Tested	58 505 944	58 639 815	63 540 387
TS default	Errors	386	361	337
	Average	408.3	397.4	346.0
	Tested	58 626 216	64 022 143	34 704 826
TS $frozen = 4000$	Errors	332	335	307
	Average	358.6	351.5	318.3
	Tested	164 463 322	158 110 736	117 925 701

Table 7: Test726, span 174

Starting Assignment		Random	One frequency	Best
SA default	Errors	40	40	42
	Average	44.1	44.6	45.8
	Tested	25 896 827	24 610 280	26 026 482
SA $t\_start = 0.1$	Errors	51	53	31
	Average	57.5	57.6	32.0
	Tested	21 170 077	21 870 392	25 889 948
TS default	Errors	63	61	31
	Average	69.8	68.2	32.9
	Tested	26 498 251	23 304 291	12 037 220
TS $frozen = 4000$	Errors	59	57	30
	Average	65.2	62.7	32.3
	Tested	73 690 753	71 215 459	32 401 910

Table 8: Ph6, span 399

Starting Assignment		Random	One frequency	Best
SA default	Errors	618	622	621
	Average	628.6	631.2	628.2
	Tested	42 223 315	40 540 514	42 406 954
SA $t\_start = 0.1$	Errors	632	630	612
	Average	644.4	639.5	616.4
	Tested	35 835 664	35 989 576	38 383 248
TS default	Errors	625	626	612
	Average	647.2	654.7	618.2
	Tested	31 471 088	28 661 348	18 578 689
TS $frozen = 4000$	Errors	627	622	606
	Average	633.0	632.2	612.0
	Tested	99 176 269	92 013 844	67 213 265

Table 9: Ph6, span 199

We can see that the results when starting from a good zero violation assignment are almost always better than starting from a random assignment or from an assignment where all transmitters are assigned the same frequency. In simulated annealing it obviously makes sense to leave the default starting temperatures for random assignments, so that annealing can slowly decrease the temperature from a high starting temperature. However this is counter productive when a good starting configuration is used. In this case, the annealing should start with a low value such as $t\_start = 0.1$, which is the default [46] for the minimum span annealing algorithm.

The results indicate that tabu search outperforms simulated annealing, almost always providing the best result. This sometimes meant a longer run time than with simulated annealing, especially with Test726. It seems doubtful the simulated annealing even with more runs or a longer annealing schedule, could improve significantly and achieve the performance of tabu search. However, the occasional poor result of tabu search when starting from a random assignment as in Figure 8 should be noted.

The errors indicated in the results table give a good comparison of simulated annealing and tabu search using different starting configurations. The results are probably not the best fixed spectrum assignments which can be achieved. For example the result on Test95, span 39 can be improved from 17 to at least 15 when changing some tabu search parameters, resulting in an increased run time. However, the challenge is to obtain better results within a similar run time, testing a similar number of configurations.

We also compared fixed spectrum results obtained with FASOFT with results by Castelino and Stephens using a surrogate constraint & tabu threshold algorithm (SCTT) [7]. The example problems are Test726 and 5 smaller instances from the same scenario. The span for all examples was set to 49. A summary of the results is presented in Table 10, where the numbers refer to the minimum number of violations and not to errors as with Tables 5 to 9. A violation is a constraint, which has not been met, irrespective by the amount it has been violated (error). It can be seen that FASOFT is able to produce better results (having less violations) without exception using run times which were at least 50% shorter.

5.5 Guidance on the use of the Heuristic Algorithms

The results presented here give guidance on the use of the heuristic algorithms. If possible, we always recommend starting from a good assignment. This is especially true for minimum span assignments, but the benefits for fixed spectrum frequency assignment are also apparent. Tabu search is normally the better heuristic, however the differences between simulated annealing and tabu search are not great. This is especially true for

	FASoft	SCTT
Test252	9	20
Test282	114	117
Test410	361	365
Test450	130	171
Test490	585	615
Test726	1466	1472

Table 10: A comparison of results, span 49

fixed spectrum problems, whether any one of the algorithms is better depends mainly on the problem.

The default parameters for both simulated annealing and tabu search work very well with a large range of problems. The only parameters we recommend changing for simulated annealing are the starting temperature (set to 0.1 if a start file is used) and possibly the number of runs. With an increase in the number of runs the search can be more diversified, the overall run time increases or decreases proportionally. Similarly we recommend a change in the number of runs for tabu search, if the time is available. The only other parameter we recommend changing for tabu search is the 'number of frozen iterations'. Set to 1000, which is the default, it is possible to do several runs in a very short time. For large problems however this value should be increased to 4000 to take account of the increase in search space. All the available parameters are listed and explained in detail in the FASoft User Manual [46].

The clique approach for minimum span frequency assignment problems is fairly automated (section 3.3 and [46]) and quite easy to perform. Sometimes however, in order to obtain the very best minimum span assignment a certain amount of understanding of the problem structure is required. A novice in frequency assignment might be unable to perform such a task, as some knowledge in graph theory is essential.

6 The Classification of Problems

There are several approaches to classification and we examined two different ways. One possible way to achieve classification is to use statistical features. Another way is to use more problem specific information contained within the constraint matrix, to achieve a general description of problem classification.

All of the classification approaches considered here try to group problems, which are defined using constraint matrices, into mutually exclusive groups. An alternative approach of assigning a value of difficulty to a constraint matrix poses many more problems and it is doubtful if the approach can be realised.

It should be stressed that all classification should be able to be done without significant expensive computational work. Also it should be possible to classify a constraint matrix before the full assignment process has started. This is important as it is hoped that the classification can guide the user on which methods to try and use in order to obtain good assignments.

6.1 Classification of Problem Difficulty

In this report we have seen that the methods developed have proved very successful in dealing with many real problems of frequency assignment. At the same time, there remain some problems for which there is a large measure of uncertainty about how good the assignments obtained are. The proportion of such problems can be expected to increase as the number of transmitters increases above 800-1000.

It seems sensible to classify in terms of the methods described in this report. Whereas the frequency assignment problem itself can be classified as NP-Hard, many problem instances of moderate size can now be solved in a reasonable time. Any classification in these terms which is method independent seems very hard to obtain. Note that here we are only concerned with describing difficulty using the constraint matrix formulation.

The principal advances in frequency assignment have been based on

1. A successful hybrid of sequential algorithms with tabu search (or simulated annealing);
2. Methods based on detecting a clique or near-clique, assigning it, fixing the assignment and extending it;

3. Generating lower bounds, particularly the TSP bound.

It can be observed that some problems are more amenable to this approach than others. Indeed, even if the hybrid algorithm (item 1) is used without item 2, some problems prove easier than others to obtain a reasonable solution.

It is not completely clear how difficulty should be classified. We might consider:

1. Problems for which an optimal solution can be found and proved. Test95 and most of the Philadelphia problems fall within this category. However, even in this case there can be a considerable variation in the ease with which an optimal assignment can be found and proved.
2. Problems for which a near optimal solution can be found and proved. Certainly for one of the variants of the Philadelphia problem (Ph1, section 5.1), it is very easy to find an assignment that is within one channel of optimality. However closing the gap appears to need a very difficult enumeration to improve the lower bound. Strict optimality may be of more interest to the mathematician than the radio engineer.
3. Problems for which there is a large uncertainty as to how far from optimal the best assignment that can be found, such as for example Test726. However, this case can be subdivided into problems for which an apparent reasonable solution can be found easily and problems for which it seems hard to find any reasonable assignment.

One clear observation from our experience is that for cellular problems, or at least problems for which the constraint values are very strongly correlated with distance, the methods described in this report are successful. This is true either if there is a reasonably dominant cluster of transmitters, or in the case of a regular layout, there is a demand vector that describes a reasonably dominant cluster of demands. It has to be said that some parts of the method may become less satisfactory above, say, 1000 transmitters.

On the other hand, for radio links problems where terrain effects and directionality of antennae make the correlation with distance less strong, it may be much more difficult to prove optimality. In particular, the clique structure may be much more complex in such problems, making the methods harder to apply.

It should also be noted that the TSP bound appears remarkably successful for cellular problems. It may be much less suitable for other problems.

6.2 Sequential Algorithms and the Classification of Problems

During the course of this work different modules for sequential algorithms have been developed (section 2.1). It is often best to use all possible sequential algorithms, in order to achieve a good sequential assignment which can be used as an effective starting configuration for heuristic algorithms. This is important as the result of the sequential algorithms (i.e. the span of an assignment) can vary considerably depending on the chosen module. In this section we investigate the relationship between the *order transmitter* module and the type of problem.

We have run 24 different sequential algorithms on all the problems available to us. The 24 different sequential algorithms were the six different initial orderings combined with the 4 different modules for *selecting a frequency*. The second module *selecting the next transmitter* was not used as it is not defined for all problems, such as those defined in CELAR Format [1], and also the run time can be prohibitive for large problems. The test problems we used were:

- 4 test cases we obtained from other researchers, for example Bt_58_4 (section 5.1).
- 6 constraint matrices derived from a problem provided to us by the Radiocommunications Agency.
- 16 test cases, which stem from military radio links problems. Test95 and Test726 (section 5.1 are contained in this set.
- regular hexagonal grid problems up to $d_o^2 = 12$ (20 problems).
- the 6 Philadelphia problems plus six scalings of the Philadelphia Ph6 problem (section 5.1).
- 40 problems, where the constraint matrix was produced by a random number generator.
- 5 cliques, the three level-0, level-1 and level-2 cliques of Philadelphia Ph6 and the level-0 cliques for Test95 and Test726.

This data set comprises all test data available to us and is more extensive than the typical examples introduced in the section 5.1.

We have run all the 24 different sequential algorithms in the order provided by FASoft $(600, 601, 602, 603, 500, \ldots, 103)$ [46] and noted the *Initial Ordering* module (section 2.1) which first found the best sequential assignment. We limited it only to the first best assignment, as most of the time only one module produced the best assignment. The *Initial Ordering* module refers to the numbers as follows: 6 - *GLF1*, 5 - *LF1*, 4 - *GSF*, 3 - *SL*, 2 - *GLF2*, 1 - *GLF2*.

We obtained the following results.

- Of the 4 test cases obtained from other researchers, one problem used ordering 6, the others used ordering 3 and 4. The problem using ordering 6 is a very small 30 transmitter problem. The other 3 problems are larger and appear to us more difficult in the sense that we are not able to find a sub constraint matrix which can be used for the clique method (section 3.3).
- Of the 6 test cases, 4 used ordering 4, 2 used ordering 6.
- The 16 test cases used only ordering 2,4 and 6.
- The 20 hexagonal problems all used either 3 or 4 as ordering.
- The random problems used all orderings, however ordering 1 was used twice and ordering 5 only once.
- All Philadelphia problems used ordering 2.
- The 5 cliques used orderings 2 and 6.

In summary and by looking more closely at the different results from orderings of problems of a similar type type (or origin), we can note:

- **ordering 1 and 5**: These 2 orderings were only best with random data, and then were only used 3 out of 40 times. These 2 orderings can be regarded as not very good for frequency assignment problems.
- **ordering 3 and 4**: When the best ordering was 3 and 4 it does not seem to be feasible to extract a sub problem which defines the overall problem. These orderings appeared for all hexagonal problems, and also, for example, for Test726.
- **ordering 2**: On problems using ordering 2 it can be expected that a sub problem can be extracted. Ordering 2 was best on Test95 and all Philadelphia problems.
- **ordering 6**: With ordering 6 the problem is often easy to solve, for example the problem with all constraints equal or some of the cliques.

In conclusions we find a classification into 3 groups:

- Orderings 1 and 5 are probably never good. This is a result concerning the development of sequential algorithms and does not help in the classification process.

- **The hexagonal group:**
  Ordering 3 and 4 provide the best result, when the clique approach is unlikely to work and the problems are similar or based upon hexagonal grid problems. If a solution is desired it is recommended to let FASOFT run directly on the problem. Also, if the detailed structure of the origin of the constraint matrix is known, it can be useful to use approaches similar to those use in the Successive Maximum Repetition algorithm (SMR) as introduced by Leese [30].

- **The direct group:**
  Problems with the best ordering 6 are more likely to be easy problems, where the direct approach with FASOFT might already provide a very good assignment and possibly even the sequential assignments themselves are already very good.

- **The clique group:**
  On problems with the best ordering 2 it is beneficial to try the clique approach.

We have found here some guidelines which provide help when a good assignment is sought on a new problem. The only work which needs to be done, to obtain the classification into one of the three groups is the run of several sequential algorithms. This is done before any run of the heuristics and is not time consuming. This is recommended in order to produce a good starting configuration for subsequent use by one of the heuristics in any case.

6.3 Statistical Classification

Classification is often performed in two steps. The first is the extraction of meaningful features and the second is the classification into different groups by means of those features. The most important aspect in this strategy is the extraction of features. Without them, any classification is impossible.

In frequency assignment, these features have to relate to the constraint matrix. This immediately introduces some difficulties regarding the available features.

- The problem which is immediately apparent is the need for features which are invariant to the size of the matrix. Obviously, a frequency assignment problem which has twice or four times the size of another problem (for example Ph6x2 and Ph6x4 in respect to Ph6, see section 5.1) is more difficult to solve heuristically, as the search space dramatically increases. This increase in difficulty however is not based on any intrinsic change of the type of problem. These problems can still be solved in the same way, the only change being a considerable increase in the run time.

- A second problem is also connected with scaling. For example, simply multiplying all values in the matrix (e.g. by multiplying all constraints by two) should not result in a different classification result.
- Furthermore a simple change in ordering of the transmitters in the constraint matrix just recreates the same problem. Several statistical classifiers could however not cope with such a change.

There are not many statistical features which are invariant against the above mentioned changes. For example the *Energy*, *Entropy*, *Inertia*, *Local Homogeneity* or *Correlation* [48] which are often used in image processing (either directly or indirectly with a dependency matrix) to classify texture (i.e. the coarseness, smoothness, regularity), cannot be used here. Also the use of first and second order *moments* applied directly to the matrix would not be invariant against the problems outlined above. However moments can be used to describe the variations of constraints in respect to transmitters as follows:

The generalised degree, G_i, is the sum of the amount of all constraints involving transmitter i. The variance is then defined as:

$$\mu_2 = \frac{1}{n}\sum_{i=0}^{n}(G_i - m)^2$$

with n the number of transmitters and m the average value for G_i:

$$m = \frac{1}{n}\sum_{i=0}^{n} G_i$$

Another simple feature, which is invariant but has an influence in the problem difficulty, when using heuristic algorithms, is the constraint density. If c is the number of constraints with n the number of transmitters, the density D is then defined as:

$$D = \frac{2c}{n(n-1)}$$

We used these two features μ_2 and D to give guidance on the classification of problems. By using only two features, it is possibly to assign any frequency assignment problem, defined as a constraint matrix, a point in the two dimensional $D\mu_2$ space.

Plotting a $D\mu_2$ diagram reveals that all the regular hexagonal grid examples are concentrated in almost the same location, and also the 16 radio link problems (Computer Science data) and the constraint matrices based upon the Philadelphia data are concentrated at destinct places (Figure 8).

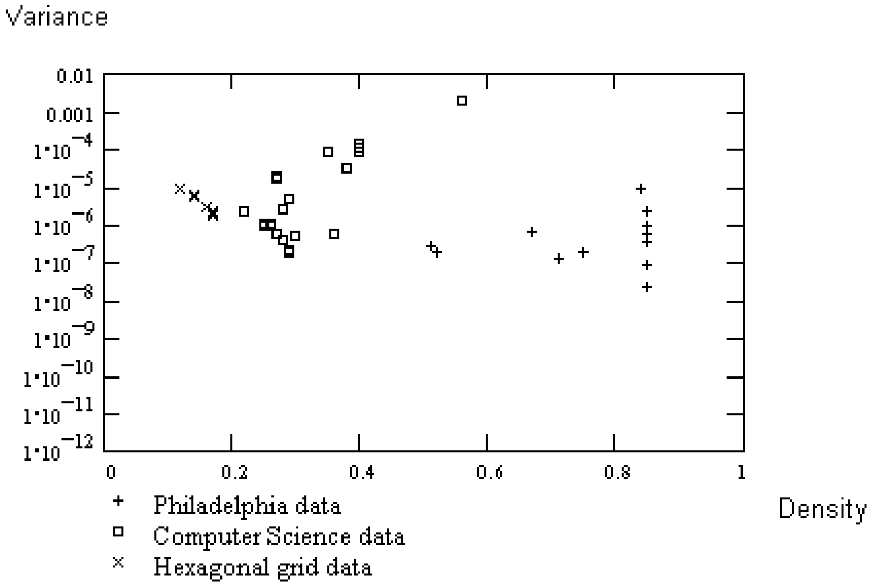

Figure 8: Plot of density against transmitter variance for constraint matrices

This is a small success, however the data is insufficient to be able to classify into groups as in section 6.2. To classify more accurately, much more test data is needed and possibly more invariant but meaningful features have to be extracted.

7 The Assessment of the Spectrum Efficiency of an Assignment

The radio frequency assignment problem concerns the assignment of frequencies to transmitters, with the aim of avoiding or minimising interference. Most work on this problem has focussed on the **process** of assigning frequencies, with relatively little consideration given to the question of how good a particular assignment is. Indeed, there does not seem to be a great deal of agreement about exactly what it is that should be measured.

Here we will make some tentative suggestions as to how this issue could be approached and show how the system FASOFT and its associated lower bounding software can be used for this purpose. We will consider only the case where the available spectrum consists of a number of discrete equally spaced channels. However, we will include fixed spectrum problems in our considerations, where there is a fixed number of channels available which may be less than the indicated minimum required number of channels.

We will also give some consideration to the question of whether it is possible to give some indication of the merit of an actual allocation of spectrum for a particular service.

The thrust of our treatment will be on how a system such as FASOFT can help in these matters. We will leave discussion of how interference should be assessed at a general level.

7.1 Construction of Constraint Matrices

FASOFT, in common with most frequency assignment systems, starts from a *constraint matrix* or its equivalent. The constraint matrix is a $N \times N$ matrix (where N denotes the number of transmitters). The element in the i'th row and j'th column of the constraint matrix indicates the required number of channels separation between the frequencies assigned to transmitters i and j. A constraint matrix would normally be obtained using protection ratios. This involves a calculation of the wanted signal received by a receiver in the service area of a wanted transmitter, and of the interfering signal received by the receiver from the unwanted transmitter. This gives a signal-to-interference ratio. In order that the required protection ratio be obtained, a number of channels of required separation is imposed between the frequencies to be assigned to the wanted and unwanted transmitters. It is these required separations that are recorded as the elements of the constraint matrix.

Definition 5 *Suppose that the wanted and unwanted signals are assessed and the number of channels of required separation are calculated according to standard recommendations. If an assignment is then found with no constraint violations (i.e. so that all of the required separations are obtained), then we shall refer to that assignment as an* ideal assignment.

For fixed spectrum problems, where a fixed number of consecutive channels are available, an ideal assignment may not be possible. Different users proceed in different ways. Some simply try to minimise the number of constraint violations, or the sum of the number of channels by which the required separations are missed, or the maximum number of channels by which any required separation is missed. Other users attach so called "soft constraints" to the transmitter/receiver pairs, which indicate the relative importance of the required separation being missed. A weighted sum of the constraint violations is then minimised. An alternative, and perhaps preferable, approach is to increase the allowable interference in some way and then recalculate the constraint matrix. A new attempt is then made to find an assignment which meets the (possibly modified) required separations. This process continues iteratively until a zero constraint violation assignment is obtained, as long as the allowable interference used can still be considered to give an acceptable grade of service.

Definition 6 *Suppose that the wanted and unwanted signals are assessed and the number of channels of required separation are calculated according to a weakening of the conditions for an ideal assignment, such that the required separations can still be considered to give an acceptable grade of service. If an assignment is then found with no constraint violations (i.e. so that all of the required separations are obtained), then we shall refer to that assignment as an* acceptable assignment.

We shall measure the spectrum efficiency of an assignment in terms of these definitions. Thus our measure assumes that assignments are found using constraint matrices. If it subsequently turns out that better assignments can be found by circumventing the use of constraint matrices then our approach would require some modification.

7.2 The Calculation of Spectrum Efficiency of an Individual Assignment

The measure that we will use for spectrum efficiency is one given by Berry [4]:

$$\frac{\text{spectrum space used by an ``ideal'' system}}{\text{spectrum space used by the system being evaluated}}.$$

If the channels are treated as positive integers then the span is the difference between the largest channel and the smallest channel, so we can modify the above definition very slightly for simplicity to:

$$\frac{\text{minimum span}}{\text{actual span}}.$$

Berry's measure is a dimensionless number between 0 and 1. In order to deal with fixed spectrum problems we will allow our measure to exceed 1. Specifically, the measure is defined to be:

$$\frac{\text{minimum span of an ideal assignment}}{\text{actual span of an acceptable assignment being evaluated}}.$$

Thus if an ideal assignment of minimum span is evaluated the measure is 1. If an ideal assignment of greater than minimum span is evaluated, the measure is less than 1, the smaller the measure the poorer the assignment. If a nonideal but acceptable assignment is evaluated, the span may be less than the minimum span of an ideal assignment and then the measure may exceed 1. This can be considered an improvement from the point of view of the spectrum authority, as long as the quality of service really is acceptable to the operator. However, if no acceptable assignment can be found within a given allocation of spectrum then the measure is not even defined. In this case the lower bounding software on the span may be used to resolve a debate about the adequacy of the given allocation for the service in question.

In calculating the measure when there is an ideal or acceptable assignment being evaluated, the denominator is given immediately as the span of the assignment. We need only consider the calculation of the numerator. If for the constraint matrix corresponding to an ideal assignment FASoft gives an assignment with span equal to the best lower bound on the span found, then the numerator is known exactly. If these numbers are not equal but the difference is small compared with the span, a good estimate of the spectrum efficiency measure is still obtained. The difficulty comes when there is a large gap between the span of the best ideal assignment obtained and the best lower bound that can be determined. Of course in this case there is no firm knowledge and the measure can only be determined accurately in such cases after further research. However, we incline to the view that the best heuristics in FASoft are better developed than the lower bounding techniques. We would incline to use an estimate much closer to the best span obtained than to the lower bound. Strictly then, we should in this case replace our measure by:

$$\frac{\text{an estimate of the minimum span of an ideal assignment}}{\text{actual span of an acceptable assignment being evaluated}}.$$

We will illustrate these points in the examples which follow.

Example 1 *For the "standard" Philadelphia problem* FASoft *gives an assignment of span 426 and a lower bound of 426 on the span can be determined. Thus* FASoft *gives* *an ideal assignment of spectrum efficiency 1. Many authors have only been able to obtain sub-optimal assignments with spectrum efficiency less than 1. If only 400 consecutive channels were available it might be possible (say by relaxing certain second adjacent channel constraints to adjacent channel constraints) to obtain an acceptable assignment of span 399. The spectrum efficiency of this acceptable assignment is 426/399.*

Example 2 *We are aware of a 726 transmitter problem typical of a military radio links problem for which the best span given by* FASOFT *is 229. However, the best lower bound on the span obtained by methods described in this report is 175. This was obtained by applying the Volgenant and Jonker travelling salesman software to a near clique (in fact a clique plus one vertex). There is no reason why the Hamiltonian path bound should be exact here. There may be many constraint violations between nonconsecutive vertices in the path. Out efforts to improve the Hamiltonian path bound for the frequency assignment problem are not yet sufficiently developed to deal with this case. Thus we would incline to a best estimate of the minimum span in the range 215 to 225, giving a quality of between 215/233 and 225/233. Further research is needed to improve this estimate or to improve our confidence in it. Very recently new techniques have improved the lower bound on the span from 175 to 181 [2].*

7.3 The Spectrum Efficiency of an Allocation

Having dealt with the estimation of the spectrum efficiency of a single assignment, it is then useful to ask if an assessment can be made of the spectrum efficiency of an Allocation made to an operator for a given service. Although this is conceptually harder, it could possibly be tackled as follows. Suppose that a number of typical instances of assignment problems for the **entire service** were generated. An efficient Allocation should allow an acceptable assignment to be found for all of the problem instances. However, no assignment should have a spectrum efficiency which is very much less than 1.

The suggestions made here for the assessment of spectrum efficiency require further debate. However, they do put into a firmer framework the way in which FASOFT and its associated lower bounding software may be used to assess quality.

8 Conclusions and Further Work

8.1 The Frequency Assignment Software

We have described a system, FASoft which can be used for state of the art frequency assignment. Fixed spectrum or minimum span problems can be solved using a variety of techniques, the most successful of which are simulated annealing and tabu search.

Given a constraint matrix defining the required channel separation between transmitters, the system can be used as a practical tool for frequency assignment. Alternatively, it can be used as a research tool capable of exploring and evaluating the effect of varying parameters in the algorithms, and hence their effect on a frequency plan.

The results we have given show that the best heuristic in FASoft generally performs very well, giving an assignment fairly close to best possible. Sometimes, the assignment can be improved by using cliques. This has the additional advantage that a good lower bound can be generated. It should be remarked that the clique method is most effective if it remains exploratory. More rigid rules governing the selection of an initial clique or the building of a near clique may prevent the very best solution from being found.

Although for many types of problem the clique method is effective, we are aware of problems for which it is not. Fixing the assignment of a clique may reduce freedom and actually make it harder to find an assignment for the full constraint graph. For cellular problems clique methods are useful if the geometry is small or if the demands are quite variable. For demands with little variation over extensive geometries the methods described here may be ineffective. For example, for many hexagonal lattice problems with one transmitter per cell the minimal span may be known exactly [30, 29]. For an extensive region the heuristics cannot discover the lattice assignment and generally perform poorly. Clique methods are unlikely to help.

Finally, further research is needed to develop tight lower bounds for fixed spectrum frequency assignment, either based on the number of constraint violations or some other measure. At present comparisons can only be made on the relative performance of algorithms without reference to their absolute performance. Also the performance of genetic algorithms is disappointing in comparison with simulated annealing and tabu search. Research is needed to determine why this is the case and to increase the effectiveness of genetic algorithms for frequency assignment.

In comparison with results published elsewhere, FASoft almost always gives the best results, the only condition is that the problem has to be defined using a constraint matrix. Several results published in the literature can be improved using FASoft, sometimes problems are very easy to solve, for example Bt_58_4 which appeared in [31] or the Philadelphia problems with a higher co-site separation than 5 [39].

FASoft has a number of features which make it unique:

- It has a user friendly interface, but can also run in command line mode or perform batch runs.
- Both minimum span and fixed spectrum assignments can be solved.
- Various input formats for the constraint matrix, including the CELAR format [1] can be used.
- It is flexible regarding frequency domains (available frequencies). Every transmitter can be allocated a unique frequency domain.
- Subgraphs can be assigned first, and this result can be used to assign the whole problem. (It also provides tools for finding *critical* subgraphs.)
- Starting assignments can be used in a flexible way.
- It incorporates sequential (greedy) algorithms, exhaustive search (Backtracking and Forward Checking), and several heuristics (with simulated annealing and tabu search being the most efficient and effective).
- Finally, FASoft produces excellent frequency assignment results.

8.2 The Classification of Problems

In section 6, we have outlined the theoretical aspects of problem classification and also used two approaches to achieve classification results. The statistical classification using the constraint density and a measure of variance is not enough to obtain a meaningful separation between groups of problems. More classification features and test data are needed for this process.

However the classification based on sequential algorithms is useful. Obviously, more tests are needed to fully understand the achieved separation into different groups and the implication of the grouping on the problem difficulty. But all the data sets available so far support the theory that the classification into the three groups (hexagonal group, direct group and clique group) is reliable. Further work could involve more features which would enable the three groups to be split into subgroups to more accurately classify problems.

8.3 The Constraint Matrix

Most of the work carried out to date has taken the constraint matrix as a starting point. Recent work at Oxford [18] has demonstrated that the choice of constraint matrix for a particular frequency assignment problem is not unique, and that the actual choice of constraint matrix may be important. Work at Royal Holloway, University of London aims to apply the theory of Constraint Satisfaction to determine the best set of (not necessarily binary) constraints to use for a particular problem.

Throughout this report, the starting point for a frequency assignment problem was the constraint matrix. The constraint matrix is, and will be for the foreseeable future, the main tool in the modelling process for frequency assignment problems. Radio engineers prefer the constraint matrix, as it gives them a unique constraint between all pairs of transmitters. However, it may well be that a move to non-binary constraints, as proposed in the Royal Holloway work in collaboration with the Cardiff/Glamorgan group, would have substantial benefits. The evaluation is currently only just beginning, but if non-binary constraints are used there would be significant but limited implications for the work described in this report.

8.4 Finding Good Initial Orderings for Sequential Assignment

Sequential assignment methods are greedy methods which mimic the way a frequency assignment problem may be solved manually. The transmitters are ordered in some way. Difficult to assign transmitters are given a higher priority than the less difficult to assign ones. Difficulty is usually assessed by the degree or generalised degree of the transmitter. Given this initial ordering transmitters are then selected one by one and assigned a frequency such that no constraints are violated.

The most popular way of generating an initial ordering of transmitters is to use the degree or generalised degree of a transmitter (vertex in the constraint graph) and produce an ordered list based on *largest first, smallest last, generalised largest first* or *generalised smallest last.*

An alternative way of generating the initial ordering is to use meta-heuristics. In particular, starting from a random ordering of the transmitters successive refinements (which differ depending on the heuristic used) are made to improve the ordering. To assess the fitness (i.e. goodness of a given transmitter ordering) frequencies are sequentially assigned to the transmitters in the order they appear in the list such that the smallest allowable frequency is used and no constraints are violated.

Results are available on using a genetic algorithm to generate the initial ordering and work is currently underway on a simulated annealing algorithm. The success of the GA is mixed. On the standard Philadelphia problem the optimal span of 426 is obtained, whereas on the 726 transmitter radio links problem a span of 243 is obtained (which is an improvement over the best sequential result of 248, but is some way short of the best known span of 229).

8.5 Further Work

Experience with using FASoft is beginning to lead to an understanding of which problems are difficult and how problem difficulty should be defined. This report has begun a study of how problems should be classified. This activity is in its infancy and needs to continue.

At the same time, some of the methods developed will become less effective as problem size increases from several hundred to several thousand transmitters. The maximum clique algorithm used, for example, is already approaching the limit of its applicability at 800 transmitters. Development of the methods to deal with larger problems will need to continue in the future.

In particular, it is proposed to continue work on two fronts:

1. At present the application of clique based procedures in the generation of a frequency assignment is not fully automated and requires a certain amount of manual intervention to produce successful assignments. Consequently, work will be done to deskill the use of the clique method in the assignment process to a level similar to that currently required to apply the heuristics directly. This would enhance the use of the assignment algorithms by Radio Engineers.

   Related to this will be the development of an heuristic algorithm for the identification of the clique or *critical subgraph*. This critical subgraph is the subgraph of the constraint graph that gives the best lower bound (of some easily calculated type). It is expected to be related to cliques and near cliques and would supersede the use of near cliques. The critical subgraph will, without manual intervention, determine the best starting configuration for assignment heuristics. An additional advantage will be that the current limitation on size forced by the exact maximum clique algorithm [6] will be removed. The critical subgraph should also lead to a useful classification of problems.

2. Currently a constraint matrix (or channel separation matrix) models the potential interference between pairs of transmitters. If the channel separation between each

pair of transmitters is satisfied in an assignment then it is considered that the interference in the network is minimised. The constraint matrix model seems to be particularly suitable for fixed links (or point to point) networks, however, it is unclear whether this model is appropriate for *area coverage* problems. The principal criteria that must be satisfied at every reception point in the coverage areas are:

(a) A specified signal-to-interference ratio between the wanted signal and any interfering signal must be attained.

(b) The field-strength must not be less than a certain minimum.

Consequently, research will be carried out to consider the most effective representation for frequency assignment in area coverage problems. The advantages of new representations relative to the constraint matrix representation will be assessed. Efficient solution of problems expressed in the new representation by modifications of existing methods will be studied.

References

[1] Celar data. ftp://ftp.cs.city.ac.uk/pub/constraints/csp-benchmarks.

[2] S.M. Allen and N. Dunkin. Frequency assignment. Technical report, 1997.

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

[4] L.A. Berry. Spectrum metrics and spectrum efficiency: proposed definitions. *IEEE Trans. on Electromagnetic compatibility*, EMC-19:313–319, 1977.

[5] A. Bouju, J.F. Boyce, C.H.D. Dimitropoulos, G. vom Scheidt, and J.G. Taylor. Tabu search for the radio links frequency assignment problem. In *Conference on Applied Decision Technologies: Modern Heuristic Methods*, pages 233–250, Brunel University, 1995.

[6] R. Carraghan and P.M. Pardalos. An exact algorithm for the maximum clique problem. *Operations Research Letters*, 9:375–382, 1990.

[7] D.J. Castelino and N.M. Stephens. A surrogate constraint tabu thresholding implementation for solving the frequency assignment problem. *Annals of Operations Research*, to appear 1997.

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

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

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

[11] B. Descartes. Solution to advanced problem number 4526. *Amer. Math. Monthly*, 61:352, 1954.

[12] A. Gamst. Homogeneous distribution of frequencies in a regular hexagonal cell system. *IEEE Transactions on Vehicular Technology*, 31:132–144, 1982.

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

[14] M. Gendreau, P. Soriano, and L. Salvail. Solving the maximum clique problem using a tabu search approach. *Annals of Operations Research*, 41:385–403, 1993.

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

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

[17] D.E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning.* Addison Wesley, 1989.

[18] R.A.H. Gower and R.A. Leese. The sensitivity of channel assignment to constraint specification. In *Proceedings of EMC97 Symposium, Zurich*, pages 131–136, 1997.

[19] W.K. Hale. Frequency assignment: theory and applications. *Proc. IEEE*, 68:1497–1514, 1980.

[20] W.K. Hale. New spectrum management tools. *Proc. IEEE International Symposium on Electromatic Compatibility Record*, pages 47–53, 1981.

[21] J.H. Holland. *Adaption in natural and artificial systems.* University of Michigan Press, Ann Arbor, 1975.

[22] S. Hurley, D. H. Smith, and S. U. Thiel. FASoft: A software system for frequency assignment. *Radio Science, to appear.*

[23] S. Hurley and D.H. Smith. Fixed spectrum frequency assignment using natural algorithms. In *Proceedings of the First Conference on Genetic Algorithms in Engineering Systems*, pages 373–378, Sheffield, September 1995.

[24] S. Hurley, S.U. Thiel, and D.H. Smith. A comparison of local search algorithms for radio link frequency. In *ACM Symposium on Applied Computing*, pages 251–257, Philadelphia, 1996.

[25] J. Janssen and K. Kilakos. Polyhedral analysis of channel assignment problems: (i) tours. Technical Report CDAM-96-17, London School of Economics, 1996.

[26] S. Kirkpatrick, C.D. Gelatt, and M.D. Vecchi. Optimisation by simulated annealing. *Science*, 220(4598):671–680, May 1983.

[27] T.A. Lanfear. Graph theory and radio frequency assignment. Technical report, NATO EMC Analysis Project No. 5, NATO Headquarters, 1110 Brussels, 1989.

[28] R. Leese. Tiling methods for channel assignment in radio communication networks. In *3rd ICIAM Congress*, 1996.

[29] R. Leese. A unified approach to problems in radio channel assignment, Oxford, 1994. In *Applications of Combinatorial Mathematics.* Oxford, 1997.

[30] R. Leese. A unified approach to the assignment of radio channels on a regular hexagonal grid. *IEEE Transactions on Vehicular Technology*, to appear.

[31] G.D. Lochtie and M.J. Mehler. Channel assignment using a subspace approach to neural networks. In *IEE Antennas and Propagation*, pages 296–300, 1995.

[32] N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller. Equations of state calculations by fast computing machines. *Journal of Chemical Physics*, 21(6):1087–1092, 1953.

[33] J. Mycielski. Sur le coloriage des graphs. *Colloq. Math.*, 3:161–162, 1955.

[34] R. Peeters. The frequency assignment problem as a weighted graph colouring problem. Master's thesis, Eindhoven University of Technology, 1991.

[35] R.C. Prim. Shortest connection networks and some generalizations. *Bell System Technical Journal*, 36:1389–1401, 1957.

[36] P. Prosser. Hybrid algorithms for the constraint satisfaction problem. *Computational Intelligence*, 9(3):268–299, 1993.

[37] A. Raychaudhuri. *Intersection assignments, T-colourings and powers of graphs.* PhD thesis, Rutgers University, 1985.

[38] F.S. Roberts. T-colorings of graphs: recent results and open problems. *Discrete Mathematics*, 93:229–245, 1991.

[39] K.N. Sivarajan, R.J. McEliece, and J.W. Ketchum. Channel assignment in cellular radio. In *Proceedings of 39th Conference, IEEE Vehicular Technolgy Society*, pages 846–850, 1989.

[40] D.H. Smith. Graph colouring and frequency assignment. *Ars Combinatoria*, 25(C):205–212, 1988.

[41] D.H. Smith and S. Hurley. Bounds for the frequency assignment problem. *Discrete Mathematics*, to appear.

[42] D.H. Smith, S. Hurley, and S.U. Thiel. Improving heuristics for the frequency assignment problem. *Submitted to European Journal of Operational Research.*

[43] D.H. Smith, S. Hurley, and S.U. Thiel. Frequency assignment algorithms. Technical report, Interim Report Year 1, Radiocommunications Agency Agreement Ref. RCCM 070, October 1995.
http://www.cs.cf.ac.uk/User/Steve.Hurley/Ra/interim1/int_rep1.html.

[44] D.H. Smith, S. Hurley, and S.U. Thiel. Frequency assignment algorithms. Technical report, Final Report Year 1, Radiocommunications Agency Agreement Ref. RCCM 070, March 1996.
http://www.cs.cf.ac.uk/User/Steve.Hurley/Ra/year1/year1.html.

[45] D.H. Smith, S. Hurley, and S.U. Thiel. Frequency assignment algorithms. Technical report, Interim Report Year 2, Radiocommunications Agency Agreement Ref. RCCM 070, October 1996.
http://www.cs.cf.ac.uk/User/Steve.Hurley/Ra/interim2/interim2.html.

[46] D.H. Smith, S. Hurley, and S.U. Thiel. *FASoft User Manual*, 2.00 edition, April 1997. Frequency assignment software Radiocommunications Agency agreement Ref. RCCM 070.

[47] G.D. Smith, A. Kapsalis, V.J. Rayward-Smith, and A. Kolen. Report 2.1: Implementation and testing of genetic algorithm approaches. Technical report, EUCLID CALMA Radio Link Frequency Assignment Project, 1995.

[48] S.U. Thiel. *The use of image processing techniques for the automated detection of blue-green algae*. PhD thesis, University of Glamorgan, November 1994.

[49] S.U. Thiel, S. Hurley, and D.H. Smith. The use of orderings in exact algorithms for the maximum clique problem. *Submitted to Journal of the Operational Research Society*.

[50] T. Volgenant and R. Jonker. A branch and bound algorithm for the symmetric traveling salesman problem based on the 1-tree relaxation. *European Journal of Operational Research*, 9:83–89, 1982.
http://www.mathematik.uni-kl.de/~wwwwi/WWWWI/ORSEP/contents.html.

[51] W. Wang and C. Rushforth. An adaptive local search algorithm for the channel assignment problem (cap). *IEEE Transactions on Vehicular Technology*, 45(3):459–466, 1996.

[52] J.A. Zoellner and C.L. Beall. A breakthrough in spectrum conserving frequency assignment technology. *IEEE Trans. on Electromagnetic Compatibility*, EMC-19:313–319, 1977.