FIXED SPECTRUM FREQUENCY ASSIGNMENT USING NATURAL ALGORITHMS

S Hurley And * D H Smith

University of Wales, * University of Glamorgan, UK

email steve@cm.cf.ac.uk, dhsmith@glam.ac.uk

Abstract

The frequency assignment problem, known to be NP-complete, is to find an assignment of radio frequencies to a set of transmitters in a region. The transmitters satisfy a large number of constraints. This paper presents a simulated annealing algorithm for determining such an assignment and compares the results with those obtained from a genetic algorithm. We report on our computational experiments in terms of quality of the solutions obtained for realistic, computer-generated problem instances.

1 Introduction

The radio spectrum is a vital but limited natural resource and the demand for frequencies is outpacing the increase in the usable spectrum due to technological change. It is therefore vital that the spectrum be managed in the most effective way possible. The management of the spectrum would be eased if frequencies could always be assigned in an optimum or near-optimum manner. However, the assignment of radio frequencies for problems of a practical size remains a considerable challenge.

Computing methods based on exact algorithms and graph theory are successful for small problems but out of the question for large problems. Methods based on the so-called sequential heuristics, which mimic the way the problem might be solved manually, are fast enough for large problems but give results which are well short of the best possible.

The main objective is to assign radio frequencies to a number of transmitters subject to a number of constraints, such that minimum interference is suffered. Ideally zero interference is desired, but given the limited region within which the transmitters operate, a more reasonable goal is to minimise the interference for a fixed set of frequencies. The problem is classified computationally as NP-complete [1]. Hence, there is no known algorithm that can generate a guaranteed optimal solution in an execution time that may be expressed as a finite polynomial of the problem dimension.

The aim of this paper is to implement a simulated annealing algorithm for fixed spectrum frequency assignment and compare it with a parallel genetic algorithm formulation given in [3,4]. Data based on realistic sets of constraints are used as test problems.

2 Frequency Assignment

We shall consider the problem where radio communication is required between a number of sites. Each site has transmitters and receivers and it is required to assign frequencies to links between certain transmitters and receivers. The available frequencies

Genetic Algorithms in Engineering Systems: Innovations and Applications
12-14 September 1995, Conference Publication No. 414, © IEE, 1995

are normally separated by 50 kHz, equivalent to one channel, hence from a finite or *fixed spectrum* set. Not necessarily all channels are available for the system, however, since some may be reserved for other purposes. Thus there may be gaps in an otherwise consecutive set of frequencies.

2.1 Interference and Constraints

In order to reduce interference, constraints are imposed on the assignment. Interference occurs when certain pairs of links are assigned frequencies which are the same or close together. This can happen when a transmitter or a receiver of links are at the same site, 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-Site Interference

This occurs when transmitters and receivers are at the same location. Technical limitations in the construction of receivers mean that certain combinations of receiver frequency and co-sited transmitter frequency are not permitted. The constraints which arise due to co-site interference include the following:

Co-site frequency separation: Any pair of frequencies at a site must be separated by a certain fixed amount, typically, for a large problem, 500 kHz or 10 channels. If a channel is to be used by a high power transmitter then its frequency separation should be larger, say 2 MHz or 40 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 radios i and j.

Intermodulation products: The second co-site constraints are aimed at intermodulation protection. These occur when two or more signals mix in a non-linear electrical system to form a third (unwanted) frequency. If it happens to be close to the wanted frequency it may not be filtered out and so will cause interference. The constraints corresponding to the worst products are:

$2f_i - f_j \neq f_k$	(two signal, third order)
$3f_i - 2f_j \neq f_k$	(two signal, fifth order)
$f_i + f_j - f_k \neq f_l$	(three signal, third order)
$2f_i + f_j - 2f_k \neq f_l$	(three signal, fifth order)

Far-Site Interference

This occurs between equipment that is separated by some distance. There are two kinds of constraint which arise from considering this type of interference.

Co-channel constraints: This is the most important factor in the consideration of far-site interference. A pair of communication circuits located at different sites must not be assigned the same frequency unless they are sufficiently geographically separated. This gives rise to constraints of the form,

$$|f_i - f_j| > 0$$

Adjacent channel constraints: When a transmitter and a victim 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 number of channels separation.

In this paper we will not consider intermodulation products and we will assume that interference can be avoided if there is sufficient channel separation between the frequencies assigned to pairs of links. Of course, the required minimal separation may be zero if no possible interference can occur between two links whatever the assignment. For other pairs of links, the minimal separation depends on distances between transmitters and receivers and on the terrain.

For the purposes of this paper, total interference for any assignment to the N trans-

mitters is measured by counting the number of constraint violations. Other measures include:-

1. the difference between the largest and smallest frequency used; this is the *span* of the assignment;
2. the number of distinct frequencies used; this is the *order* of the assignment;
3. the weighting of constraints so that more critical constraints contribute more to the total interference;
4. the sum of the positive discrepencies of any constraint violations;
5. a combination of these four.

In mathematical terms the problem can be described using the following definitions:

Definition 1: A *constraint graph* G is a finite, simple, undirected graph with each edge labelled with an integer in the set {0,1,...,L}.

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

Definition 2: A *frequency assignment* in G is a mapping f:V→F (where F is a set of consecutive integers 0,1,...,K) 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. A particularly important case is when T_i is a set of consecutive integers {0,1,...,i} and we shall only consider constraints of this type in this paper.

Costa [2] has implemented an exact algorithm for determining the minimum span of a constraint graph of reasonable size, by enumerating implicitly all possible colourings of the graph under consideration. Results were presented for this branch and bound approach for graphs up to 36 nodes. The method is unrealistic for the large-sized problems considered here.

3 The Simulated Annealing Algorithm

The simulated annealing (SA) algorithm used for the frequency assignment is as follows:

Input:

Number of transmitters, N_{trans}.

The constraint graph G.

The frequency set F.

Minimum temperature t_{min}.

Initialisation:

Generate a random assignment of frequencies to transmitters. Denote this by $\mathbf{X_{old}}$.

Calculate number of constraint violations E_{old}.

Determine starting temperature, t_0.

Main Loop:

$k = 0$

WHILE $t_k > t_{min}$ DO

FOR $u = 1$ to N DO

generate new assignment, $\mathbf{X_{new}}$ by randomly selecting a transmitter and assigning a random frequency from F;

calculate number of constraint violations, E_{new};

calculate $\Delta E = E_{new} - E_{old}$;

IF $\Delta E < 0$ or $random < prob = e^{-\Delta E/t_k}$ THEN

$\mathbf{X_{old}} \Leftarrow \mathbf{X_{new}}$

$E_{old} \Leftarrow E_{new}$

END IF

END FOR

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

$k = k + 1$

END WHILE

N_{trans}	N^o far-site constraints	N^o co-site constraints	Size of search space	Edge density
282	10162	268	10^{479}	0.258
410	22346	411	10^{696}	0.271
450	25416	305	10^{764}	0.255
490	35535	489	10^{832}	0.301
726	74595	711	10^{1233}	0.286

Table 1: Test problems

Output:

Assignment $\mathbf{X}$ of transmitter frequencies.

Number of constraint violations for the assignment $\mathbf{X}$.

The variable *random* is a randomly generated number in the range [0,1]. 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 (i.e $N = 100$). If the acceptance ratio, χ, defined as the number of accepted trial assignments divided by 100 (N), is less than 0.8, double the current value of t_0. Continue this procedure until the observed acceptance ratio exceeds 0.8 (with χ reinitialised to zero prior to starting the FOR loop).

N is chosen such that equilibrium is reached for each temperature t_k. In the test problems considered in this paper N is set equal to the number of transmitters i.e N_{trans}. The parameter δ is set to 0.1 and

$$\sigma(t_k) = \frac{1}{N_{trans}} \cdot \sqrt{H}$$

where

$$H = N_{trans} \cdot \sum_{i=1}^{N_{trans}} \left(E_i{}^k\right)^2 - \left(\sum_{i=1}^{N_{trans}} E_i{}^k\right)^2$$

and where $E_i{}^k$ is the number of constraint violations, for each assignment i, in the k^{th} Markov chain (i.e. at temperature t_k).

4 Results

The test problems considered are detailed in the Table 1. The size of the search space for each problem is based on a fixed spectrum set of 50 frequencies. The edge density of each constraint graph is given as a real number between 0 (no edges present) and 1 (full connectivity i.e. $N_{trans}*(N_{trans}-1)/2$ edges present). All the test problems consisted of a mixture of co-channel constraints, adjacent channel constraints (up to three were considered) and co-site constraints (with 10 channels separation required) i.e.

$$|f_i - f_j| > 0$$
$$|f_i - f_j| > 1$$
$$|f_i - f_j| > 2$$
$$|f_i - f_j| > 3$$
$$|f_i - f_j| \geq 10$$

where f_i, f_j correspond to the frequencies assigned to transmitters i and j.

Since the problem is *NP*-complete, it is impossible to obtain an exact value for the minimum interference for the large problems we have considered. Comparisons, therefore can only be made with other heuristic algorithms. To allow such a comparison, we use the genetic algorithm formulation used in [3,4] as a benchmark (for completeness, Appendix A contains details of the chromosome representation used in the genetic algorithm). In particular, we tested 200,000 assignments for each test problem and noted the wall clock time for the length of each run. The times allowed were 100 minutes for the 282 problem and 153 minutes, 176 minutes, and 222 minutes for the 410, 450 and 490 transmitter problems. The results

N	Best Interference		Average Interference		Standard Deviation		Average N^o of Assignments Tested	
	SA	GA	SA	GA	SA	GA	SA	GA
282	121	501	128.5	506.9	3.532	8.02	2.769×10^6	2×10^5
410	367	1160	380.35	1173.0	7.862	15.50	2.490×10^8	2×10^5
450	151	845	161.4	849.0	4.849	4.58	2.876×10^7	2×10^5
490	625	1828	636.1	1834.4	7.823	8.08	1.204×10^8	2×10^5
726	1488	3947	1509.8	3960.8	17.523	24.43	1.437×10^7	2×10^5

Table 2: SA and GA comparison for test problems (10 runs each).

are given in Table 2. We note that the SA algorithm gives improved results over the GA implementation, but also note that the time per assignment is much lower, enabling substantially more assignments to be generated. This is because the SA implementation can use more efficient function evaluation procedures than is used in the GA implementation.

5 Conclusions

We have described a simulated annealing algorithm for the frequency assignment problem and presented a synopsis of the test results. While it is difficult to assess the quality of the solutions for practically sized computationally hard problems, there is, nonetheless, some evidence to believe that the method competes favourably with other heuristic methods such as genetic algorithms.

6 References

1. D.J. Castelino, S. Hurley and N.M. Stephens, A Tabu Search Algorithm for Frequency Assignment, to appear *Annals of Operations Research*, 1995.

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

3. W. Crompton, S. Hurley and N.M. Stephens, Frequency assignment using a parallel genetic algorithm, *Proceedings of the IEE/IEEE Natural Algorithms in Signal Processing Workshop*, 2 (1993) pp 26/1 - 26/8.

4. W. Crompton, S. Hurley and N.M. Stephens, A parallel genetic algorithm for frequency assignment problems, *Proceedings of the IMACS/IEEE Conference on Signal Processing, Robotics and Neural Networks*, (Lille, France, 1994) pp 81-84.

Appendix A

The chromosome representation used in the genetic algorithm [3] has been designed to take better advantage of the Schemata theorem by grouping together those sites assigned the same frequency and designing crossover so that children retain many of their parental features contributing to fit solutions. Let $f_1, ..., f_k$ be the available frequencies and $s_1, ..., s_n$ the transmitting sites. An assignment is a function $a : \{1, ..., n\} \to \{1, ..., k\}$ where $a(i) = j$ means that site i is assigned frequency f_j. Such an assignment partitions the set $\{1, ..., n\}$ into k subsets $P_1, ..., P_k$ (some possibly empty) so that $a(i) = j$ means that $i \in P_j$. The chromosone is the array P of subsets.

For this representation, *single point crossover* produces two children C^1, C^2 from two parents P^1, P^2 by first selecting a cross-frequency index x where $1 \leq x < k$. For $m = 1, 2$, the subsets C^m_j are the same as the subsets for the corresponding parent, i.e.

j	P_j^1	P_j^2	C_j^1	C_j^2
1	1,5,9	6,10,11	1,5,9	6,10,11
2	2,10,16	8,12	2,10,16	8,12
3	4,8,13	4,9,13,14	4,8,13	4,9,13,14
4	3,11,14,15	2,5,7,15	7,11,14,15	2,3,5,15
5	6,7,12	1,3,16	3,6,12	1,7,16

Table 3: Example of 1-point crossover

for each $j, 1 \leq j \leq x$, $C_j^m = P_j^m$. For $x < j \leq k$, however, it is not possible to define $C_j^1 = P_j^2$, for example, because the resulting array will not correspond to a partition. The following strategy is therefore adopted. Suppose $s \in P_j^1$ where $x < j \leq k$, so that s has yet to be placed in C^1. The rule is that if $s \in P_l^2$ with $x < l \leq k$ then s is placed in C_l^1 but otherwise in C_j^1. Thus for placing s in the second part of C^1, the second parent's assignment has priority as long as it is compatible. Similarly for C^2, the first parent's assignment has priority.

In the example given in Table 3 with $k = 5, n = 16$, the parents P^1, P^2 are given with their resulting children C^1, C^2 with cross-frequency index $x = 3$.

An obvious disadvantage of this representation and crossover is that subsets for lower frequencies have a smaller probability of changing than subsets for higher frequencies. *Two point crossover* can overcome this problem by choosing two frequency indices x and y $(x < y)$. For $m = 1, 2$, we set $C_j^m = P_j^m$ for $x \leq j < y$. The other subsets are treated according to similar rules for one point crossover. The ideas of two point crossover may be extended to *uniform crossover*. A k-bit random number $b_1 ... b_k$ is used to determine those subsets inherited directly by their children. If $b_j = 1$ then $C_j^m = P_j^m$ for $m = 1, 2$. If $s \in P_j^1$ with $b_j = 0$ and $s \in P_l^2$, then s is placed in C_l^1 if $b_l = 0$ but otherwise in C_j^1.

The parameters used by the genetic algorithm to generate the results given in Table 2 are the following:

maximum # of generations.	1000
# of parallel populations	4
population size	50
crossover probability	0.75
mutation probability	0.015
migration rate	2 % of population
migration frequency	every 25 generations.
scaling	yes
elitism	yes
duplication avoidance	no
niche formation	no

A two-point crossover operator is used. Mutation is at random, according to a prescribed rate; an element of a chromosome is replaced by a random, allowable frequency. Finally, selection is proportional to fitness and uses a biased roulette wheel implementation.