A Permutation Based Genetic Algorithm for Minimum Span Frequency Assignment

Christine Valenzuela[1], Steve Hurley[2] and Derek Smith[3]

[1] School of Computing and Mathematics, University of Teesside, TS1 3BA.[†]
[2] Department of Computer Science, Cardiff University, CF2 3XF, UK.
[3] Division of Mathematics and Computing, University of Glamorgan, CF37 1DL, UK.

Abstract. We describe a Genetic Algorithm (GA) for solving the minimum span frequency assignment problem (MSFAP).The MSFAP involves assigning frequencies to each transmitter in a region, subject to a number of constraints being satisfied, such that the span, i.e. the range of frequencies used, is minimized. The technique involves finding an ordering of the transmitters for use in a sequential (greedy) assignment process. Results are given which show that our GA produces optimal solutions to several practical problem instances, and compares favourably to simulated annealing and tabu search algorithms.

1 Introduction

The frequency assignment problem is a difficult, NP-hard, problem of considerable importance. The radio spectrum is a limited natural resource that is used in a variety of civil and military services. The most well known example would be in cellular mobile phone networks. Third generation mobile systems will achieve a world-wide mass market giving enhancements in the areas of quality and security, incorporating broad-band and multi-media services, and offering higher capacity based on bandwidth on demand. These features will be supported by means of integrated terrestrial and satellite modes of delivery to provide comprehensive coverage - ranging from the office environment, through street and urban areas, to complete coverage of rural and remote regions throughout the world.

To facilitate this expansion the radio spectrum allocated to a particular service provider needs to be assigned as efficiently and effectively as possible. The minimum span frequency assignment problem (MSFAP) is one in which it is required to assign frequencies to a set of transmitters such that certain compatibility constraints, which model potential interference between pairs of transmitters, are satisfied. In addition to satisfying the constraints the objective is to minimise the *span* of the assignment i.e. the difference between the largest frequency used and the smallest frequency used.

The purpose of this paper is to explore the possibility of combining a Genetic Algorithm (GA) with sequential (greedy) assignment methods. Traditionally,

[†] *Current address:* Department of Computer Science, George Mason University, Virginia, USA.

GAs and other meta-heuristics have been applied to an initial solution consisting of an assignment of frequencies to transmitters, the techniques operating on the problem directly using local search to perturb the allocation of frequencies in an attempt to minimise the number of constraint violations [3, 11, 17]. In [11] a sequential method is used to find an initial assignment, which is then improved by simulated annealing or tabu search. With the GA described here the iterative transformations are applied to permutations of transmitters. A simple sequential assignment algorithm is then applied to each of these permutations to produce an allocation of frequencies that does not violate any constraints. Thus the permutations of transmitters output by the GA are interpreted by the sequential algorithm to produce candidate solutions to the MSFAP.

1.1 Interference and Constraints

Interference can occur between a pair of transmitters if the interfering signal strength is sufficiently high. Whether a transmitter pair has the potential to interfere depends on many factors, e.g. distance, terrain, power, antenna design. The higher the potential for interference between a transmitter pair the larger the *frequency separation* that is required. For example, if two transmitters are sufficently geographically separated then a frequency can be re-used i.e. the same frequency assigned. At the other extreme if two transmitters are located at the same site then they may require, say, five frequencies separation (this is called the *co-site* constraint).

To model this interference a *constraint graph* is constructed which gives the separations needed between each transmitter pair. This graph is usually represented by a $N \times N$ matrix, A, (N is the number of transmitters in the network) where each element a_{ij} defines the frequency separation between transmitters i and j i.e. if f_i and f_j are the frequencies assigned to transmitter i and j respectively then

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

The MSFAP is to find a frequency assignment that satisfies all the constraints and such that the span of the assignment is minimised.

1.2 Sequential Assignment Algorithms

Sequential assignment methods mimic the way the problem might be solved manually. They are fast enough for large problems but tend to give results which are well short of the best possible. The transmitters are simply considered one at a time, successively assigning allowable frequencies as we proceed, until either we have assigned all transmitters or run out of frequencies. An important factor affecting the quality of solutions generated by this method is how the next transmitter is chosen. We may therefore generate a series of assignment methods based on three components:

- **initial ordering,**
- choice of next transmitter,
- assignment of frequency.

The simplest way to choose the next transmitter is sequentially, simply picking the next one on the list produced by the initial ordering. A more complicated method, which has proved more effective than sequential selection with the various initial ordering methods, is called general saturation degree. In this method the choice of the next transmitter is influenced by the constraints imposed by all those transmitters that have already been chosen. One could view the more complicated process as a method for correcting those mistakes that have already been made by the initial ordering technique.

The simplest assignment technique is to assign the smallest acceptable channel i.e. the lowest numbered channel to which it can be assigned without violating any constraints. Variations upon this technique attempt to assign transmitters to channels that are already used in favour of those that are not. A detailed description of sequential assignment methods can be found in [11].

In this paper we use our GA to search a state-space of initial orderings. The choice of the next transmitter is made sequentially using the ordering obtained, with the smallest acceptable frequency assigned to each transmitter.

2 The Search Space Generated by Permutations of Transmitters

It is important to establish that the state-space of initial orderings contains permutations capable of producing good (or even optimal) solutions following the application to the orderings of the chosen method for allocating frequencies to the transmitters. Some experiments on small problems are documented below. We allocate frequency channels to the initial orderings in the following way: the choice of the next transmitter is made sequentially and the smallest acceptable channel is assigned to each transmitter. All possible permutations of a simple 12 transmitter problem are generated (479,001,600 in total) allowing the examination of the entire state-space, and 1000 permutations are produced at random for a problem containing 95 transmitters. Graphs indicating the frequency of occurrence of the spans evaluated for the permutations, using the simple frequency assignment algorithm outlined above, are plotted in Figure 1 and Figure 2.

The graphs show a range of spans for both of the problems. The exhaustive search of the permutation space for the 12 transmitter problem locates the optimum span of 22, whilst the random search carried out in the case of the 95 transmitter problem locates a best span of 52, which is 4 channels above the optimum solution of 48 [17]. In both cases, however, these values represent improvements over the best produced by combining together the various sequential assignment methods (initial ordering, selecting the next transmitter and selecting a frequency), where best solutions of 24 and 54 are obtained respectively for the 12 and 95 transmitter problem. The above experiments demonstrate clearly that "good" permutations of transmitters can be converted into excellent solutions

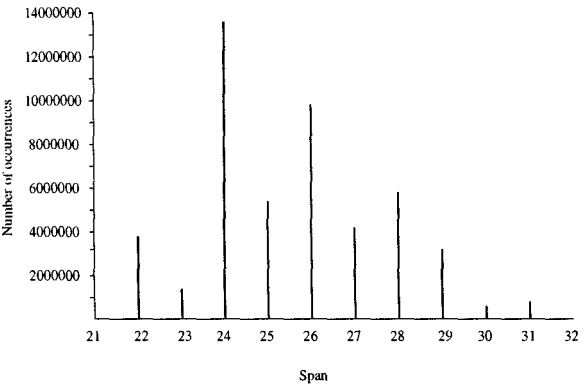


Fig. 1. Spans evaluated from all permutations of a 12 transmitter problem

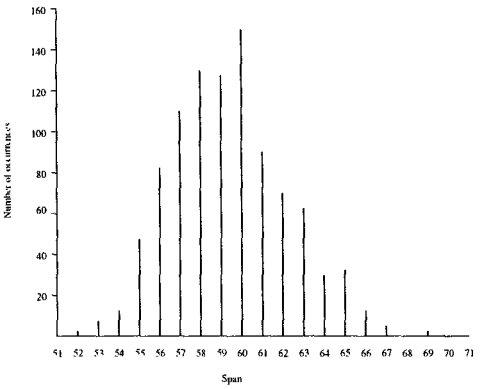


Fig. 2. Spans evaluated from 1000 permutations of a 95 transmitter problem

to the MSFAP, at least in the case of small problem instances, thus providing the motivation to try a genetic search on larger problem instances.

3 The Genetic Algorithm

The simple genetic algorithm (GA) used here is based on that which has appeared in [20]. It is derived from the model of Holland in [9] and is an example of a 'steady state' GA (based on the classification of Syswerda in [18]). It uses the 'weaker parent replacement strategy' first described by Cavicchio in [2]. The GA, outlined in Figure 3, applies the genetic operators to permutations of transmitters. The fitness values are based on the spans produced when the simple sequential assignment algorithm is applied to each permutation list produced by the GA. In Algorithm 1 the first parent is selected deterministically in sequence, but the second parent is selected in a roulette wheel fashion, the

```
Procedure GA1
BEGIN
   Generate N_pop random permutations (N_pop is the population size).
   Apply the Generalised Saturation Degree (GSD) algorithm to each
   individual to produce N_pop frequency assignments and store each one.
   Store best-so-far.
   REPEAT
     FOR each member of the population
       this individual becomes the first parent;
       select a second parent using roulette wheel selection on ranks;
       apply cycle crossover to produce one offspring;
       apply mutation to offspring;
       evaluate span produced by offspring;
       if offspring better than weaker parent then it replaces it in
        population;
       if offspring better than best-so-far then it replaces best-so-far;
     ENDFOR
   UNTIL stopping condition satisfied.
Print best-so-far.
END
```

Fig. 3. Algorithm 1 - The Genetic Algorithm

selection probabilities for each genotype being calculated using the following formula:

$$genotype\ selection\ probability = \frac{(population\ size + 1 - Rank\ of\ geneotype)}{\sum Ranks}$$

where the genotypes are ranked according to the values of the spans that they have produced, with the best ranked 1, the second best 2 etc.

The GA breeds permutations of transmitters and frequencies are assigned to the resulting lists of transmitters by a simple sequential assignment technique (see section 1.2 above).

Experiments have demonstrated that better results are obtained if the Generalized Saturation Degree (GSD) algorithm is applied to the initial permutation lists produced by a pseudo-random number generator, prior to invoking the Genetic Algorithm. (These experiments will be documented elsewhere.) The GSD algorithm used in the second step of Algorithm 1 is described next.

3.1 Generalized 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

u and v 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 generalized saturation degree of v. The technique for selecting the next transmitter is as follows: Select a transmitter with maximal generalized saturation degree (break ties by selecting the transmitter occurring first in the initial ordering).

3.2 Mutation

The mutation chosen was to select two transmitters at random from a permutation list, and swap them.

3.3 Cycle Crossover

Permutation crossovers were originally developed primarily for the travelling salesman problem (TSP), where the genotypes consist of lists of cities which are converted to TSP tours. Because TSP tours are circuits, it is irrelevant which city is represented first on the list. The permutation lists represent cycles and an edge in a TSP tour always joins the last city on the list to the first. Thus for the TSP it is the relative sequence of cities that is important, rather than the absolute sequence. In the frequency assignment problem, however, the permutation lists making up the genotypes represent lists of transmitters, and intuitively it would seem likely that absolute sequences are important in this case.

The best known permutation operators from an historical standpoint (which are also amongst the simplest to implement) are *Partially Matched Crossover* (PMX) [8], *Order Crossover* (OX) [4, 5] and *Cycle Crossover* (CX) [13]. In test experiments if was found that CX produced better results on the MSFAP than either PMX or OX, thus it is the chosen operator here. (For a description of the three crossovers, PMX, OX and CX see [8]).

In the next subsection we examine the ability of the cycle crossover to pass characteristics of parents onto their offspring. In order to do this Pearson's Correlation Coefficient has been calculated between the spans of 1000 offspring versus and the corresponding mid-parent values.

3.4 Offspring versus Mid-parent Correlation

From an initial population of 1000 individuals, 1000 pairs of parents are selected using the selection mechanism defined in Figure 3, and from them 1000 offspring are generated using CX and one mutation. It is important to establish that offspring resemble their parents and produce solutions with similar values for

the span. If this turns out not to be the case, then the GA is at best equivalent to a random search.

Table 1 shows the values of Pearson's Correlation Coefficient, r_{xy}, for a range of problems using CX for offspring versus mid-parent values of the span for 1000 samples. Results are shown for six representative problem instances.

Table 1. Examples of offspring versus mid-parent correlation

Problem	r_{xy}
P1	0.2703
P2	0.2405
P3	0.2419
P4	0.2767
P5	0.2061
P6	0.2098

The values for the correlation coefficient in the table are all highly significant at the 0.0001% level, showing that parental features that contribute to their span values are indeed passed on to their offspring.

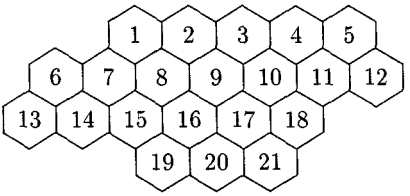
4 Results

For the GA the population size is 500. One mutation per offspring is used, and cycle crossover is the recombination operator. The GA terminates after 200 generations have elapsed with no improvement to the best-so-far. The population is initially seeded with 500 random permutations. These are then subject to the Generalized Saturation Degree algorithm, and it is the new orderings produced by the application of this algorithm which form the starting population for the GA.

The examples are based on the so-called Philadelphia problem which originally appeared in [1], and was subsequently used by several authors [6, 7, 12, 14, 21]. The problem is based on a cellular phone network consisting of 21 cells. The *demands* in each cell define the number of frequenices that need to be assigned to each of the cells. The distance between cell centres is taken to be 1. The hexagonal geometry is given in Figure 4 Constraints between transmitters are generated by considering the distance between the transmitters.

Additional variations on the Philadelphia problem can be also be defined to further test the performance of the GA. Table 2 defines the Philadelphia variations used for the test problems (d_k denotes the smallest distance between transmitters which can use a separation of k channels, N denotes the total number of frequencies that need to be assigned i.e. the number of transmitters and C denotes the total number of compatability constraints that need to be satisfied. Problem P8, described in [19], contains the same number of transmitters and

Fig. 4. The Cellular Geometry of the Philadelphia Problems



constraints as P1, the difference being some of the constraints require a higher separation (since the d_1 distance is smaller). The following cell demand vectors are used:

$\mathbf{m}$ = (8,25,8,8,8,15,18,52,77,28,13,15,31,15,36,57,28,8,10,13,8)
$\mathbf{m_2}$ = (5,5,5,8,12,25,30,25,30,40,40,45,20,30,25,15,15,30,20,20,25)
$\mathbf{m_3}$ = (20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20)
$\mathbf{m_4}$ = (16,50,16,16,16,30,36,104,154,56,26,30,62,30,72,114,56,16,20,26,16)

Table 2. Philadelphia problem variations

Problem	d_0	d_1	d_2	d_3	d_4	d_5	Cell demands	N	C
P1	$\sqrt{12}$	$\sqrt{3}$	1	1	1	0	$\mathbf{m}$	481	97,835
P2	$\sqrt{7}$	$\sqrt{3}$	1	1	1	0	$\mathbf{m}$	481	76,979
P3	$\sqrt{12}$	$\sqrt{3}$	1	1	1	0	$\mathbf{m_2}$	470	78,635
P4	$\sqrt{7}$	$\sqrt{3}$	1	1	1	0	$\mathbf{m_2}$	470	56,940
P5	$\sqrt{12}$	$\sqrt{3}$	1	1	1	0	$\mathbf{m_3}$	420	65,590
P6	$\sqrt{7}$	$\sqrt{3}$	1	1	1	0	$\mathbf{m_3}$	420	44,790
P7	$\sqrt{12}$	$\sqrt{3}$	1	1	1	0	$\mathbf{m_4}$	962	391,821
P8	$\sqrt{12}$	2	1	1	1	0	$\mathbf{m}$	481	97,835

The GA results are given in Table 3. Comparisons are given with Tabu Search (TS) and Simulated Annealing (SA) algorithms which are detailed in [11]. The TS and SA algorithms do not operate by generating a good permutaion of transmitters (as in the GA) but instead attempt to iteratively improve a complete assignment. The value in parentheses indicates the number of assignments processed by the SA, TS, and GA algorithms. The"Best seq" column gives the best span obtained from using a standard sequential (greedy) assignment method, details can be found in [11]. Lower bounds on the minimum span have been calculated in [16, 17, 15]. It can be seen that for the problems considered here the GA always equals or outperforms the SA algorithm and only fails to improve on the results of TS algorithm in one case (P6). The number of assignments

Table 3. Genetic algorithm results (* denotes one run)

Problem	Lower bound	Best seq.	TS	SA	GA	GA (mean 4 runs)
P1	426	447	428 (76,902,785)	428 (8,973,775)	426 (147,500)	426.25
P2	426	475	429 (76,843,178)	438 (81,066,945)	426 (186,000)	426*
P3	257	284	269 (74,920,399)	260 (103,073,177)	258 (225,500)	259.25
P4	252	268	257 (81,215,422)	259 (9,057,107)	253 (186,000)	253.75
P5	239	250	240 (66,941,918)	239 (9,494,007)	239 (275.500)	239.5
P6	178	230	188 (70,277,837)	200 (9,052,191)	198 (119,000)	198
P7	855	894	858 (9,617,414)	858 (162,899,774)	856 (212,000)	856*
P8	524	592	535 (190,046,283)	546 (410,198,288)	527 (650,000)	527*

tested is considerably lower for the GA, although the time to generate a new assignment is much higher than TS or SA.

It should be mentioned that the TS algorithm can outperform the GA in examples P3, P4, P6 and P8 (finding optimal solutions for P1,P2,P3,P4,P5,P7 and P8) if a *critical subgraph* of the original constraint graph is identified and a minimum span assignment for this subgraph initially found and this assignment is used as the starting (partial) assignment for the complete problem. However, generating assignments using this subgraph approach remains exploratory and no firm foundation exists for its general applicability. Full details of the subgraph approach can be found in [10, 17, 15].

5 Conclusions

A genetic algorithm which computes minimum span frequency assignments has been presented. Results of using the GA on a set of standard benchmark problems show that the GA performs well against SA and TS algorithms that have appeared previously in the literature. It has been noted that the TS algorithm when used in conjuction with the assignment of critical subgraphs improves on the assignments generated by the GA. However, it is important to note that the the critical subgraph approach is exploratory. Consequently, algorithms are needed which compute frequency assignments from the constraint graph of the complete problem. However, future work will involve testing the performance of the GA on the critical subgraph approach.

References

1. 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.
2. D.J. Cavicchio. *Adaptive search using simulated evolution.* PhD thesis, University of Michigan, 1970.

3. W. Crompton, S. Hurley, and N.M. Stephens. A parallel genetic algorithm for frequency assignment problems. In *Proc. IMACS/IEEE Conference on Signal Processing, Robotics and Neural Networks*, pages 81–84, Lille, France, 1994.
4. L. Davis. Applying adaptive algorithms to epistatic domains. In *Proceedings 9th International Joint Conference on Artificial Intelligence*, pages 162–164, 1985.
5. L. Davis. Job shop scheduling with genetic algorithms. In J. Grefenstette, editor, *Proceedings International Conference on Genetic Algorithms and their Applications*, pages 136–140. Lawrence Erlbaum Associates, 1985.
6. N. Funabiki and Y. Takefuji. A neural network parallel algorithm for channel assignment problems in cellular radio networks. *IEEE Transactions on Vehicular Technology*, 41(4):430–437, 1992.
7. A. Gamst. Some lower bounds for a class of frequency assignment problems. *IEEE Transactions on Vehicular Technology*, 35:8–14, 1986.
8. D.E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning.* Addison Wesley, 1989.
9. J.H. Holland. *Adaption in natural and artificial systems.* University of Michigan Press, Ann Arbor, 1975.
10. S. Hurley and D.H. Smith. Meta-heuristics and channel assignment. In R.A. Leese, editor, *Methods and algorithms for channel assignment.* Oxford University Press, to appear 1998.
11. S. Hurley, D.H. Smith, and S.U. Thiel. FASoft: A system for discrete channel frequency assignment. *Radio Science*, 32:1921–1939, 1997.
12. R. Leese. Tiling methods for channel assignment in radio communication networks. In *3rd ICIAM Congress*, 1996.
13. I.M. Oliver, D.J. Smith, and J.R.C. Holland. A study of permutation operators on the travelling salesman problem. In *Proceedings 2nd International Conference on Genetic Algorithms*, pages 224–230, 1987.
14. 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.
15. D.H. Smith, S.M. Allen, and S. Hurley. Lower bounds for channel assignment. In R.A. Leese, editor, *Methods and algorithms for channel assignment.* Oxford University Press, to appear 1998.
16. D.H. Smith and S. Hurley. Bounds for the frequency assignment problem. *Discrete Mathematics*, 167/168:571–582, 1997.
17. D.H. Smith, S. Hurley, and S.U. Thiel. Improving heuristics for the frequency assignment problem. *European Journal of Operational Research*, to appear 1998.
18. G. Syswerda. Uniform crossover in genetic algorithms. In J.D. Schaffer, editor, *Proceedings 3rd International Conference on Genetic Algorithms*, pages 2–9. Lawrence Erlbaum Associates, 1989.
19. D. Tcha, Y. Chung, and T. Choi. A new lower bound for the frequency assignment problem. *IEEE/ACM Transactions on Networking*, 5(1):34–39, 1997.
20. C.L. Valenzuela. *Evolutionary divide and conquer: A novel genetic approach to the TSP.* PhD thesis, Imperial College, University of London, 1995.
21. 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.