A New Genetic Algorithm for Minimum Span Frequency Assignment using Permutation and Clique

Shouichi Matsui **Ken-ichi Tokoro**
Communication and Information Research Laboratory
Central Research Institute of Electric Power Industry
2-11-1 Iwado-kita, Komae-shi, Tokyo 201-8511, JAPAN

Abstract

We propose a new Genetic Algorithm (GA) for solving the minimum span frequency assignment problem (MSFAP). The MSFAP is minimizing the range of the frequencies assigned to each transmitter in a region satisfying a number of constraints. The proposed method involves finding and ordering of the transmitters for use in a greedy (sequential) assignment process, and it also utilizes graph theoretic constraint to reduce search space. Results are given which show that our GA produces optimal solutions to several practical problems, and the performance of our GA is far better than the existing GAs.

1 INTRODUCTION

The frequency assignment problem (FAP) is a very important problem today, but is a difficult, NP-hard problem. The radio spectrum is a limited natural resource used in a variety of private and public services, the most well known example would be in cellular mobile phone networks. 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) requires assigning frequencies to a set of transmitters in such a way that certain compatibility constraints, which model potential interference between pairs of transmitters, are satisfied. In addition to satisfying the constraints the objective is to minimize the span of the assignment, i.e., to minimize the difference between the largest frequency and the smallest frequency used.

Because the MSFAP is a very important problem in the real world and an NP-hard problem, a number of heuristic algorithms have been proposed. There are papers that apply genetic algorithm (GA) to MSFAP (e.g., Crompton et al., 1994; Cuppini, 1994; Kim et al., 1995; Hurley and Smith, 1995; Lai and Coghill, 1996; Ngo and Li, 1998; Valenzuela et al., 1998). The existing GA-based algorithms use naïve representation, therefore the search space is too large in scale, and results obtained using these algorithms are not good enough compared with other heuristics such as tabu-search (Valenzuela et al., 1998).

This paper proposes a new genetic algorithm that performs far better than the existing GAs. The proposed algorithm uses GA as a meta-heuristics for a greedy (sequential) assignment method. 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 transmitter output by the GA are interpreted by the sequential algorithm to produce candidate solutions to the MSFAP.

The proposed GA is tested using a set of standard benchmark problems, and the performance is superior to the existing GAs. The proposed GA can obtain the optimal solutions unable to be found using the existing GAs. The GA finds the optimal solution with less iteration than existing GAs in cases where the previously proposed GAs can find the optimal solution.

In this paper we first describe the problem, provide a brief overview of the frequency assignment problem in general, constraints, formulation, constraint graph, and summary of previous works. Then we show the proposed algorithm, chromosome representation used, sequential assignment algorithm, fitness function, GA operators, and parameters. Following this, we demonstrate the performance of the GA by applying it to standard benchmark problems, and discuss our findings and future works.

2 MINIMUM SPAN FREQUENCY ASSIGNMENT PROBLEM

In this section, we briefly describe the problem of MSFAP. The survey papers (Hale, 1980; Katzeka and Naghshineh, 1996) provide a good overview of the frequency assignment problem. This section is a very brief summary of these papers.

First we describe the frequency assignment in general, then give constraints, followed by formulation of the problem, and the constraint graph that plays an important role in our algorithm. Finally we show summary of previous works.

2.1 FREQUENCY ASSIGNMENT IN GENERAL

Any given radio bandwidth can be divided into a set of non-interfering radio channels. All such channels can be used simultaneously while satisfying an acceptable received radio signal and noise ratio, or while no interference occurs between a pair of transmitters. The frequency assignment problem (FAP) is to find an assignment that satisfies these condition. Frequencies are used as channels, therefore frequency assignment is sometimes called as channel assignment.

Channel assignment schemes can be divided into three categories, fixed channel assignment (FCA), dynamic channel assignment (DCA), and hybrid channel assignment (HCA).

In FCA schemes, the area is divided into a number of cells, and a number of channels are assigned to each cell. FCA schemes are very simple, however, they do not adapt to changing traffic conditions. DCA strategies have therefore been introduced to overcome the deficiencies.

In DCA, all channels are placed in a pool and are assigned to new calls as needed in such a way that the interference condition is satisfied. The cost of higher complexity is offset by flexibility and traffic adaptability. But DCA schemes are less efficient than FCA under high load conditions. To overcome this drawback, HCA schemes were designed by combining FCA and DCA strategies.

Because FCA schemes are simple, many mobile communication systems are still designed based on the FCA, so we consider channel assignment algorithms for FCA in this paper.

2.2 INTERFERENCE AND CONSTRAINTS

Interference can occur between a pair of transmitters if strength of the the interfering signal is sufficiently high. Whether a transmitter pair has the potential to interfere depends on many factors, e.g., distance, terrain, power, or antenna design. The higher the potential for interference between a transmitter pair, the larger the frequency separation required. For example, if two transmitters are sufficiently geographically separated then a frequency can be re-used, i.e., the same frequency can be assigned. At the other extreme if two transmitters are located at the same site then they may require, say, five frequencies separation. As a general rule, we should take the following three constraints into account.

- **Co-channel constraints**: A pair of transmitters located at different site must not be assigned the same frequency unless they are sufficiently separated geographically to reduce interfering signal strength.
- **Adjacent channel constraints**: When a transmitter and receiver are tuned to similar frequencies, there is still the potential for interference. Therefore there should be some separation between two frequencies.
- **Co-cite channel separation**: Any pair of frequencies assigned to transmitters at the same site must be separated by a certain fixed number of channels. This number is usually significantly larger than the adjacent channel constraints.

To model these constraints a compatibility matrix is constructed which provides the separations needed between each transmitter pair. This matrix is usually represented by a $n \times n$ matrix C (n is the number of transmitters in the network) where each element c_{ij} denotes 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 the following condition should be satisfied for all i and j,

$$|f_i - f_j| \geq c_{ij}.$$

2.3 FORMULATION

Let $X = \{x_1, x_2, \cdots, x_n\}$ be a set of cells in a cellular system. A demand vector on X is an n-vector $M = (m_i)$ with nonnegative integer elements. The element m_i represents the number of radio frequencies required for cell x_i. Radio frequencies are assumed to be evenly spaced, so they can be identified with the positive integers. A compatibility matrix on X is a symmetric $n \times n$ matrix $C = (c_{ij})$ with nonnegative integer elements. The value c_{ij} prescribes the minimum frequency separation required between frequencies assigned to cell x_i and cell x_j. If $c_{ij} = \nu$, cell x_i and x_j are said to be ν-compatible with each other, then, a triple $P = (X, M, C)$ characterizes FAP. For simplicity, we assume that the requirements of each cell are ordered.

An *feasible frequency assignment for P* will be a collection $F = (f_i)$ of positive integers, $i = 1, \cdots, n, k = 1, \cdots, m_i$, $j = 1, \cdots, n, l = 1, \cdots, m_j$, such that

$$|f_i^k - f_j^l| \geq c_{ij},$$

for all indices i, j, k, l (except for $i = j, k = l$), where f_i^l is the frequency assigned to the l-th requirement of cell x_i. The span $S(F)$ of a frequency assignment F is the difference between the largest frequency and the smallest frequency assigned to the system, i.e.,

$$S(F) = \max_{i,l} f_i^l - \min_{i,l} f_i^l.$$

The objective of the MSFAP is to find a feasible frequency assignment F with the minimum span $S_o(P)$, i.e., to find

$$S_o(P) = \min\{S(F') | \text{all feasible } F' \text{ for } P\}.$$

Without loosing generality, we can assume that $\min\limits_{i,l} f_i^l$ is 1, therefore the MSFAP is the problem of finding the feasible assignment of which $\max\limits_{i,l} f_i^l$ is minimum.

2.4 CONSTRAINT GRAPH

The frequency assignment problem can be stated as a generalized graph coloring problem (e.g., Hale, 1980). The transmitters are vertices of a graph and the frequencies can be regarded as a set of colors to be assigned to the vertices. The edge that connects vertices x_i and x_j is labeled $c_{ij}(> 0)$. If colors are numbered from 1, and a_i^k, a_j^l are the color assigned to vertex i and j respectively then they should satisfy the condition

$$|a_i^k - a_j^l| \geq c_{ij},$$

which is equivalent to the interference constraints.

A constraint graph G is a finite, simple, undirected graph with each edge labeled with an integer that corresponds to an element of the compatibility matrix. A clique in G is a maximal complete subgraph of G. The nodes in a clique cannot be assigned the same frequency.

A constraint graph and clique are important for calculating lower bounds of span (Smith and Hurley, 1997). They can also improve the performance of heuristic algorithms (Smith et al., 1998).

2.5 PREVIOUS WORKS

Because MSFAP is an important and very difficult problem to be solved exactly, GA based algorithms have been proposed (e.g., Crompton et al., 1994; Cuppini, 1994; Kim et al., 1995; Hurley and Smith, 1995; Lai and Coghill, 1996; Ngo and Li 1998; Valenzuela et al., 1998).

The chromosome representation used in the previous works can be divided into three categories. In the first, a possible assignment is represented by a matrix $V[M, N]$ where M is the number of frequencies and N is the number of transmitters. Having '1' in position (i, j), i.e., $V_{ij} = 1$, means that frequency i can be used in transmitter j, while '0' in position (i, j) means it is forbidden to use frequency i in transmitter j.

In the second, a possible assignment is represented by a string where the total length is the sum of all the frequencies required for each of the transmitter. A gene takes the values in the range of $[1, M]$ where M is the number of frequencies, and the gene in position i represents the frequency number assigned to the transmitter i.

These two schemes simply map possible solutions to a chromosome, therefore special operations for crossover or mutation are required to satisfy interference constraints. For example, offsprings generated from two parents that satisfy interference constraints do not always satisfy the constraints. Search space that is represented by these schemes tends to be very extensive, and large portion of the space corresponds to infeasible assignments.

The third scheme is the representation that maps a possible assignment indirectly. Assignment order of transmitters is represented by permutations and assignment is carried out using a sequential algorithm. While this scheme has overcome the weakness of the previous two schemes and the performance has improved, it is still not good enough when compared with other heuristic algorithms (Valenzuela et al., 1998).

Moreover, previous works do not utilize the graph theoretic constraints described in the previous section. For example, we can utilize the fact that the transmitters in a clique in a constraint graph cannot share a frequency, so we can expect that assigning frequencies to the transmitters in clique first, then assigning to the remaining transmitters would improve performance. This approach is mentioned as a future work in the paper (Valenzuela et al., 1998). The results shown in later section demonstrate that this approach can improve performance.

3 THE PROPOSED ALGORITHM

We propose a new algorithm for the MSFAP based on genetic algorithm. The main idea is to use the GA as a meta-heuristics, i.e., we use the GA to find assignment order for the sequential algorithm. This scheme is the same as that of the GA (Valenzuela et al., 1998), however we also utilize the graph theoretic constraints to reduce the search space.

We use ordinal representation as proposed by Grefenstette et al. for the 'Traveling Salesman Problem (TSP)' (Grefenstette et al., 1985) for chromosome to represent traveling order. And to utilize the graph theoretic constraints, we partition a chromosome into sections representing the order in which transmitters in cliques are assigned as blocks.

The ordinal representation for TSP is a direct mapping of a solution candidate, and its performance is no better than a random search. Using it for controlling sequential (greedy) algorithm for the MSFAP, however, it performs well as shown in the later section.

The simple genetic algorithm (GA) used here is an example of 'steady state', overlapping populations GA. The proposed GA is outlined in Figure 1.

3.1 CHROMOSOME REPRESENTATION

We divide transmitters into partitions, i.e., into a family of clique $Q = (Q_1, Q_2, \cdots, Q_{N_p-1})$ and others using constraint graph G. First we find the maximum clique of G and set Q_1, then remove nodes in Q_1 from the G. Then we find the maximum clique of the new G. These steps continue until original G is divided into N_p partitions. The number of partitions (N_p) depends on the structure of the constraint graph. Setting N_p = 2, i.e., partitioning into the maximum clique and others, shows good performance in many cases. This is discussed more in detail in the later section.

Let $P = (P_1, P_2, \cdots, P_{N_p})$ be the partition, and $L = (L_1, L_2, \cdots, L_{N_p})$ be the numbers of transmitters in each partition. A chromosome is a sequence of natural numbers

```
Procedure Divide
BEGIN
  Initialize: Let $G$ be the constraint graph, $N$ the number of
    nodes of $G$, $N_p$ the given number of partition, and $i \leftarrow 1$.
  LOOP
    Find clique: Find maximum clique $C_i$ of $G$.
    Grouping: Let nodes in $C_i$ belong to the partition $P_i$.
    Update $G$: Remove nodes in $C_i$ from $G$.
    Update $i$: $i \leftarrow i + 1$.
  UNTIL $i \geq N_p$.
  Let nodes in $G$ belong to the partition $P_{N_p}$.
END

Procedure GA
BEGIN
  Initialize:
    Divide transmitters into $N_p$ partitions using Divide.
    Generate $N$ random individuals ($N$ being the population
      size).
    Produce $N$ assignments and store each one.
    Store best-so-far.
  LOOP
    Generate offsprings: Generate $N_n$ offsprings.
      1. Select two parents by roulette wheel rule;
      2. Apply two-point crossover with the probability of
         $P_c$, and generate two offsprings; when no crossover
         is applied, then two parents will be offsprings;
      3. Apply mutation to offsprings with probability of
         $P_m$ ;
      4. Evaluate span produced by offsprings;
      5. If an offspring is better than best-so-far then replace
         best-so-far.
    Selection: Select best $N$ individuals from the pool of old
      $N$ and new $N_n$ individuals.
  UNTIL stopping condition satisfied.
  Print best-so-far.
END
```

Figure 1: Outline of the proposed algorithm

that represents the followings.

Assignment Order of Partitions

The first block of N_p genes represents the assignment order of partitions $P = (P_1, P_2, \cdots, P_{N_p})$ by ordinal representation. The gene at the position $i, i = 1, \cdots, N_p$ takes an integer in the range of $[1, N_p - i + 1]$. The genes in the block are interpreted as follows. Let $G_P = (g_1, g_2, \cdots, g_{N_p})$ be the sequence of genes in the block, $P = (P_1, P_2, \cdots, P_{N_p})$ be the partitions corresponding to the genes and P be represented as a stack. Then $g_i (i = 1, \cdots, N_p)$ is interpreted in such a way that the partition in the position g_i from the stack top (assuming that the position 1 means the top) is the next block of transmitters to be assigned.

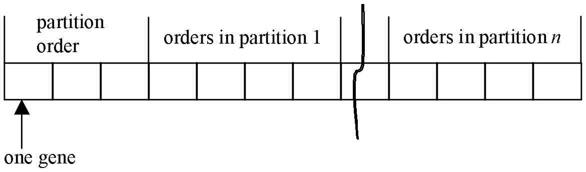

Figure 2: Chromosome in the GA

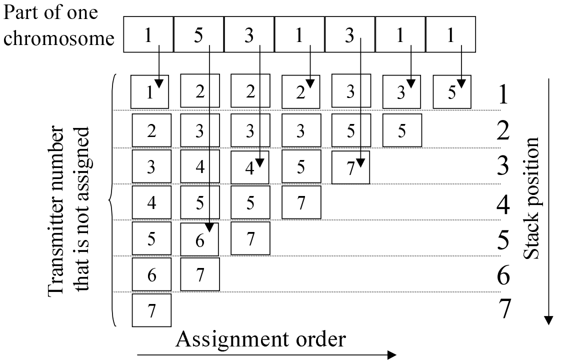

Figure 3: Chromosome representation of ordering

Assignment Order of Transmitters in a Partition

The remaining blocks of genes represents the assignment order of transmitters in partitions $P = (P_1, P_2, \cdots, P_{N_p})$, e.g., the genes at the position $i, i = 1, \cdots, L_1$ in the partition P_1 takes an integer in the range of $[1, L_1 - i + 1]$. The genes in partitions are interpreted as follows. Let $G_i = (g_1, g_2, \cdots, g_{L_i})$ be the sequence of genes in a partition, $T_i = (t_1, t_2, \cdots, t_{L_i})$ be transmitters corresponding to the genes and T_i be represented as a stack. Then $g_j (j = 1, \cdots, L_i)$ is interpreted in such a way that the transmitter in the position g_j from the stack top (assuming that the position 1 means the top) is the next transmitter to be assigned.

The chromosome representation used in the GA is depicted in Figure 2, and Figure 3 depicts a partition in a chromosome and the ordering of transmitters determined by it. It represents a case where 7 transmitters are in the block. The length of partition is 7 (the number of transmitters).

The sequence $(1, 5, 3, 1, 3, 1, 1)$ represents the assignment order indirectly, and the numbers written vertically show the sequence number of unassigned transmitters. The value '1' of gene at position 1 indicates that the first unassigned transmitter will be assigned. The assignment ordering of transmitters is determined as $(1, 6, 4, 2, 7, 3, 5)$ by this chromosome.

3.2 SEQUENTIAL ASSIGNMENT ALGORITHM

Sequential assignment methods for the MSFAP are identical to the way the frequencies might be assigned manually. They are very fast for large problems, but the results are usually far from the best possible. Transmitters are simply considered one at a time, and successively assigning usable frequencies in sequence, until all transmitters are assigned or allowable frequencies are used. The quality of solutions is generally affected by the assignment order of transmitters. Assignment methods can be classified by the following three components (Hurley et al., 1997).

- initial ordering
- choice of next transmitter
- assignment of frequency

The simplest method for choosing the next transmitter is sequentially picking up the next one on the list generated by the initial ordering. And the simplest method for the initial ordering is simply to order the transmitters according to the sequence number of transmitters.

The simplest assignment technique is to assign the smallest acceptable frequency, i.e., the smallest frequency to which it can be assigned without violating any constraints.

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

The greedy algorithm that we use in our GA is as follows. Assignment order of transmitters is determined as follows;

1. A block of transmitters is assigned sequentially according the order determined by the chromosome as described in the previous section.
2. In a block, assignment order is determined by the chromosome as described in the previous section.

Transmitters will then be assigned one at a time sequentially the minimum frequency that does not violate interference constraints.

3.3 FITNESS, OBJECTIVE FUNCTIONS

The objective function F used in the GA is defined as,

$$F = \{C_1 f_{\text{fmax}} - C_2 f^{\text{end}} - (B_1 P_1 + B_2 P_2)\}/f_{\text{fmax}},$$

where C_1, C_2, B_1, B_2 are constants, f_{fmax} is the allowable maximum frequency number, f^{end} is the span obtained. The terms P_1 and P_2 are a type of penalty, and defined as,

$$P_1 = \begin{cases} f^{\text{clq}} - f_{\text{fmax}} - A & f^{\text{clq}} > (f_{\text{fmax}} + A) \\ 0 & \text{otherwise,} \end{cases}$$

$$P_2 = \begin{cases} f^{\text{end}} - f_{\text{fmax}} - A & f^{\text{end}} > (f_{\text{fmax}} + A) \\ 0 & \text{otherwise,} \end{cases}$$

where f^{clq} is the span when the assignment to transmitters that belong to the clique is finished, and A is a constant for the threshold.

We use sigma truncation as the scaling function. The fitness function F' is defined as,

$$F' = F - (\mu_F - 2 \times \sigma_F)$$

where μ_F is the average, and σ_F is the standard deviation over the population.

The GA evolves population to maximize the F'.

3.4 SELECTION, CROSSOVER, AND MUTATION

We use roulette wheel selection, two-point crossover, and flip mutation in the proposed GA.

In the two-point crossover, two random positions are chosen, and the chromosome of two individuals P_a, P_b are cut at these points into $(P_a^1, P_a^2, P_a^3), (P_b^1, P_b^2, P_b^3)$. The offsprings are generated as (P_a^1, P_b^2, P_a^3) and (P_b^1, P_a^2, P_b^3).

As the mutation, the value of a randomly chosen gene g_i is replaced by a randomly chosen integer of the range $[1, (L_j - i + 1)]$ where L_j is the number of genes in the partition.

The crossover and the mutation do not generate invalid chromosome, i.e., offsprings are always valid representation of assignment ordering.

3.5 STOPPING CONDITION

The algorithm terminates when obtained span is less than or equal to the given lower bound. The lower bound can be calculated by a variety of methods (e.g., Gamst, 1986; Smith and Hurley, 1997).

3.6 CLIQUES

Finding maximum clique is also an NP-complete problem for general graph, but for the constraint graphs constructed from the MSFAP problem, computation time by the existing algorithm (e.g., Carraghan and Padalos, 1990) is small compared to GA iteration.

4 PERFORMANCE OF THE GA

4.1 TEST PROBLEMS

In this section, we demonstrate the performance of the proposed algorithm by considering the the same problems used in the paper by Valenzuela et al. (Valenzuela et al., 1998). The examples are based on the so-called *Philadelphia problem* and were subsequently used by several authors (Gamst, 1986; Kim et al., 1995; Lai and Coghill, 1996; Hurley and Smith, 1995; Ngo and Li, 1998). These problems are based on a cellular phone system consisting of 21 cells.

Table 1: 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	M	481	97,835
P2	$\sqrt{7}$	$\sqrt{3}$	1	1	1	0	M	481	76,979
P3	$\sqrt{12}$	$\sqrt{3}$	1	1	1	0	M_2	470	78,635
P4	$\sqrt{7}$	$\sqrt{3}$	1	1	1	0	M_2	470	56,940
P5	$\sqrt{12}$	$\sqrt{3}$	1	1	1	0	M_3	420	65,590
P6	$\sqrt{7}$	$\sqrt{3}$	1	1	1	0	M_3	420	44,790
P7	$\sqrt{12}$	$\sqrt{3}$	1	1	1	0	M_4	962	391,821
P8	$\sqrt{12}$	2	1	1	1	0	M	481	97,835

M = (8,25,8,8,8,15,18,52,77,28,13,15,31,15,36,57,28,8,10,13,8)
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 = (20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20,20)
M_4 = (16,50,16,16,16,30,36,104,154,56,26,30,62,30,72,114,56,16,20,26,16)

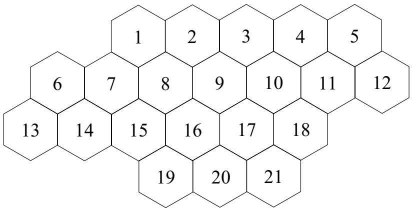

Figure 4: Cellular geometry of test problems

The demands in each cell define the number of frequencies that need to be assigned to each of the cells. The distance between the cell centers is assumed to be 1, and all transmitters in each cell are assumed to be located at the center. The hexagonal geometry is given in Figure 4. The constraints between transmitters are generated by considering the distance between the transmitters.

Table 1 defines the Philadelphia variations used as the test problem. In Table 1, 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 compatibility constraints that need to be satisfied

4.2 PARAMETERS

For the GA the population size (N) is 500, i.e., the same number as in the paper (Valenzuela et al., 1998). The mutation ratio (P_m) is 0.02 for P1 and P2, 0.025 for P3, P4, P5, and P6, and 0.0125 for P7 and P8. These values were determined by preliminary experiments.

The crossover ratio (P_c) is 0.8, and the number of new offsprings (N_n) is 500. The GA terminates if the span is less than or equal to the given lower bound, or terminates after 1000 generations have elapsed with no improvement to the best-so-far and average, or when generation number reaches the upper bound. The upper bound is set to 9000 for all problems.

The constants are set to $C_1 = 30, C_2 = 15, B_1 = 10, B_2 = 20, A = 3$ in the test, and f_{fmax} is set to the lower bound reported in the paper (Valenzuela et al., 1998).

4.3 RESULTS

We tested performance of the GA with regard to the eight problems by running 100 times. The results are shown in Table 2. Comparisons with the results by Valenzuela et al. (Valenzuela et al. 1998) are also given. In Table 2, the column labeled 'best' show the best span obtained, numbers in parentheses show the number of assignment tested, the column labeled 'mean' shows the average of spans, the column labeled 'ratio(%)' show the ratio of the obtained span is equal to the lower bound, and N_p shows the number of partitions.

The table shows that our GA performs well, only failing to find the optimal solution in P6. It outperforms the previous GA (Valenzuela et al., 1998) in all cases, the span obtained by our GA is less than or equal to that obtained by Valenzuela et al. Generations, or assignments tested in iteration, to reach optimal solutions are smaller except for the case P5. In P1, P2, the ratio of them are less than $1/10$. For the case P4 and P7, our GA can find the optimal with roughly $1/3$ tests whereas that of Valenzuela et al. could not find the optimal.

Valenzuela et al. reported that a tabu-search based algorithm can obtain the lower bound for the problem P1, P2, P3, P4, P5, P7, and P8 if *critical subgraph* of 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 (Valenzuela et al., 1998). The performance of our GA to the seven problems are same, i.e., our GA can find the optimal solutions to the seven problems.

The possible number of permutations for P1 is 481! $\sim$

Table 2: Results for Philadelphia problem variations

Problem	Lower	GA by Valenzuela et al. (4 runs)		Our GA (100 runs)			
	bound	best	mean	best	mean	ratio(%)	N_p
P1	426	426 (147,500)	426.25	426 (21,500)	426.00	100	2
P2	426	426 (186,000)	426*	426 (15,500)	426.00	100	2
P3	257	258 (225,500)	259.25	257 (778,000)	258.39	14	2
P4	252	253 (186,000)	253.75	252 (68,500)	252.33	68	2
P5	239	239 (275,500)	239.71	239 (386,000)	239.73	32	2
P6	178	198 (119,000)	198*	194 (312,500)	195.79	0	4
P7	855	856 (212,000)	856*	855 (51,500)	855.00	100	2
P8	524	527 (650,000)	527*	524 (373,500)	524.51	64	2

Notes: (1) * denotes one run, (2) numbers in parentheses indicate the minimum number of assignments tested to obtain the optimal assignment.

Table 3: Number of generations to the optimal

Problem	With Clique		Without Clique	
	μ	σ^2	μ	σ^2
P1	128	8462	207	38543
P2	56	322	146	6194

Table 4: Distribution of spans for P3, P4, P5, P6, and P8

Problem	Optimal	+1	+2	+3	+4
P3	14	44	32	9	1
P4	68	31	1	—	—
P5	32	67	3	—	—
P8	64	22	13	1	—

Distribution of spans for P6

span	194	195	196	197	198
frequency	10	30	35	21	4

10^{1084} in Valenzuela et al. and $360!(481 - 360)! \sim 10^{966}$ in our GA (481 being the total number of transmitters and 186 is the number of transmitters in the maximum clique). For P2, they are $481! \sim 10^{1084}$ and $275!(481 - 275)! \sim 10^{941}$ respectively[1]. This shows that partitioning of transmitters reduce search space considerably, leading to better performance.

To confirm the effectiveness of search space reduction, we run our GA 100 times for the problem P1 and P2 setting $N_p = 1$, i.e., without utilizing clique. Table 3 shows the mean (μ) and the variance (σ^2) of the number of generations to reach the optimal solution. It is clear that utilizing maximum clique to reduce search space surely improve the performance. Statistical tests validate the improvement, i.e., the difference of average number of generations (μ) is significant at the 0.01% level.

For the problems where our GA failed to find the optimal in runs, the distributions of spans are shown in Table 4 as the difference from the optimal for P3, P4, P5, P8 and the span itself for P6. Table 4 shows that the spans obtained by our GA are very close to the optimal for P3, P4, P5, P8, the largest being +4 from the optimal, and spans within +1 of the optimal with the probability of 50%.

For the case P6 in which our GA could not find the optimal solution, the span is less than or equal to that of the GA by Valenzuela et al. The best span has been improved by 4, decreasing from 198 to 194.

With these observations, we can conclude that the performance of the proposed GA is better than the previously reported GA.

4.4 DISCUSSION AND FUTURE WORK

The proposed GA performs well against the previously proposed GA. It can attain the lower bound, i.e, it can find the optimal solutions to seven problems, whereas the previous GA can only find in three problems. It still however is unable to find the optimal solution in one problem (P6). More sophisticated chromosome representation and GA operators would overcome the problem, so we will investigate these.

Natural permutation encoding and operators such as that are used in the GA by Valenzuela et al. seems to be a good candidate. But small experiments showed that it does not always improve the performance, i.e., it does not always decrease the number of generations to reach the optimal solutions. More extensive experiments should be carried out to find good chromosome representation and GA operators.

The parameters used in the performance test were determined by preliminary experiments, and may not be optimal.

[1] for P3 $470! \sim 10^{1053}$ and $258!(470 - 258)! \sim 10^{911}$, for P4 $470! \sim 10^{1053}$ and $180!(470 - 180)! \sim 10^{919}$, for P5 $420! \sim 10^{921}$ and $240!(420 - 240)! \sim 10^{797}$, for P6 $420! \sim 10^{921}$ and $140!100!100!(420 - 340)! \sim 10^{675}$, for P7 $962! \sim 10^{2453}$ and $720!(962 - 720)! \sim 10^{2219}$, for P8 $481! \sim 10^{1082}$ and $360!(481 - 360)! \sim 10^{966}$.

In small size experiments, we have found that the mutation probability (P_m) affects performance greatly, i.e, sensitivity of P_m to performance (number of generations to reach the optimal solution) is larger than that of the other parameters. More extensive experiments are necessary to determine the optimal parameters.

The number of partition (N_p) was set to 2 except for P6 where N_p = 4. This setting was based on preliminary experiments. Setting N_p = 2 for P6 gave a bad result, the average span increased by 3, and the frequency of obtaining a span of 194 decreased to 2. The difference between P6 and other problems is that the cell demands in P6 are uniform and the number of interference constraints is small. Further investigation is required to find firm conditions, and rules for determining the number of partitions.

5 CONCLUSIONS

We have presented here a new genetic algorithm that computes minimum span frequency assignments. The algorithm uses the GA as a meta-heuristics to determine the assignment order for the greedy algorithm. It also utilizes graph theoretic constraints to reduce search space.

The proposed GA is tested using a set of standard benchmark problems, and the performance is superior to the existing GAs. The proposed GA can obtain the optimal solutions that were unable to be found using the existing GAs. The GA finds the optimal solution with less iteration than existing GAs in cases where the previously proposed GAs can find the optimal solution. In addition, its performance is good enough compared with tabu-search based algorithm, it can find the optimal solutions that tabu-search can find and existing GAs cannot find.

We believe that our approach, reduce search space by utilizing additional constraints that are hidden in problem structure, can be applied to other real world problems.

Acknowledgments

The authors are grateful to their colleagues and reviewers for their constructive comments and suggestions.

References

Carraghan, R. and R. M. Padalos (1990). An exact algorithm for the maximum clique problem, *Oper. Res. Lett.*, vol.9, no.6, pp.375–382.

Crompton, W., S. Hurley, and N. M. Stephens (1994). Applying genetic algorithms to frequency assignment problems, *Proc. SPIE Conf. Neural and Stochastic Methods in Image and Signal Processing*, vol.2304, pp.76–84.

Cuppini, M. (1994). A genetic algorithm for channel assignment problems, *Eur. Trans. Telecommun.*, vol.5, no.2, pp.285–294.

Gamst, A. (1986). Some lower bound for a class of frequency assignment problems, *IEEE Trans. Veh. Technol.*, vol.35, no.1, pp.8–14.

Grefenstette, J. J., R. Gopal, B. J. Rosmaita, and D. Van Gucht (1985). Genetic algorithms for the traveling salesman problem, *Proc. Int. Conf. on Genetic Algorithms and Their Applications*, pp.160–168.

Hale, W. K. (1980). Frequency assignment: theory and applications, *Proc. of IEEE*, vol.68, no.12, pp.1497–1514.

Hurley, S. and D. H. Smith (1995). Fixed spectrum frequency assignment using natural algorithms, *Proc. of Genetic Algorithms in Engineering Systems: Innovations and Applications*, pp.373–378.

Hurley, S., D. H. Smith, and S. U. Thiel (1997). FASoft: a system for discrete channel frequency assignment, *Radio Science*, vol.32, no.5, pp.1921–1939.

Katzeka, I. and M. Naghshineh (1996). Channel assignment schemes for cellular mobile telecommunication systems: a comprehensive survey, *IEEE Personal Commun.*, vol.3, no.3, pp.10–31.

Kim, J.-S., S. H. Park, P. W. Dowd, and N. M. Nasrabadi (1995). Comparison of two optimization techniques for channel assignment in cellular radio network, *Proc. of IEEE Int. Conf. Commun.*, vol.3, pp.850–1854.

Lai, K. W. and G. G. Coghill (1996). Channel assignment through evolutionary optimization, *IEEE Trans. Veh. Technol.*, vol.45, no.1, pp.91–96.

Ngo, C. Y. and V. O. K. Li (1998). Fixed channel assignment in cellular radio networks using a modified genetic algorithm, *IEEE Trans. Veh. Technol.*, vol.47, no.1, pp.163–172.

Smith, D. H. and S. Hurley (1997). Bounds for the frequency assignment problem, *Discrete Mathematics*, no.167/168, pp.571–582.

Smith, D. H., S. Hurley, and S. U. Thiel (1998). Improving heuristics for the frequency assignment problem, *Eur. J. Oper. Res.*, vol.107, no.1, pp.76–86.

Valenzuela, C., S. Hurley, and D. Smith (1998). A Permutation based algorithm for minimum span frequency assignment, *Proc. 5th International Conference on Parallel Problem Solving from Nature—PPSN V*, Amsterdam, pp.907–916.