FAPSTER - A Genetic Algorithm for Frequency Assignment Problem

Nihat Karaoglu
Computational Modeling Lab
Vrije Universiteit Brussel
Pleinlaan 2, 1050 Brussels Belgium
+32 476 23 92 06
nkaraogl@vub.ac.be

Bernard Manderick
Computational Modeling Lab
Vrije Universiteit Brussel
Pleinlaan 2, 1050 Brussels Belgium
+ 32 2 629 35 24
bmanderi@vub.ac.be

ABSTRACT

In this paper, we describe an efficient Genetic Algorithm (GA) for solving the minimum interference frequency assignment problem (MI-FAP) using a new problem representation. The GA we present looks for the best assignment of a limited number of frequencies to a set of stations minimizing the total disturbance caused by stations operating at the same frequencies. We show that our problem representation based on permutation encoding and clustering gives better results in comparison with the existing problem representations in the literature. Our problem representation reduces the search space from f^n to $(f!)^{\lceil n/f \rceil}$ and improves the time complexity of the fitness function from $O(n^2)$ to $O(n^2/f)$ where f is the number of frequencies and n is the number of stations. We compare the performance of our algorithm with algorithms, which use other problem representations and confirm our results on a real-world problem.

Categories and Subject Descriptors

J.2 [**Engineering**]: Evolutionary Programming, Genetic Algorithms, Optimization http://www.acm.org/class/1998/

General Terms

Algorithms

Keywords

Genetic Algorithm, Frequency Assignment Problem, Channel Assignment Problem, Problem Representation, Permutation Encoding, Graph Coloring Problem, Clustering, Fitness Function

1. INTRODUCTION

The goal of the frequency assignment problem (FAP) or channel assignment problem is finding an optimal assignment of a limited number of available communication frequencies to a set of stations in a radio network, which minimizes the electro-magnetic disturbance caused by the re-use of frequencies. The solution of the FAP is classified as NP-Complete [24] as the problem can easily be reduced to graph coloring problem [15].

There has been a lot of interest to the frequency assignment problems from both the scientific community and the business community with the increasing popularity of wireless networks since optimizing the usage of the available frequencies means higher traffic capacity, more bandwidth and bigger coverage for the existing radio networks. As a result frequency assignment problems have been investigated by many researchers and a wide variety of models and solution techniques have been proposed including relaxation based [8][4][27] and heuristics based algorithms like Tabu Search[5][7][6] and Simulated Annealing [14][20][2]. Unfortunately these heuristics methods suffer from getting stuck to local minima or the convergence behavior is affected strongly from the seed used for random number generators, where as the application of stochastic methods like GAs makes it possible to explore the solution space efficiently in FAP and lowers the danger of dwelling on a local minimum [3].

GAs have been effective heuristic search methods for solving NP-Complete problems such as the traveling salesman problem or graph coloring problems [18]. Since FAP can be reduced to graph coloring problem GAs have been applied to FAP by many authors [11][12]. However, these solutions use models originating from the analog cellular design, which do not reflect the true nature of the problem [9] nor do they benefit from the problem specific information to reduce the search space.

Valenzuela et al. [26] have presented a GA which uses permutation based representation to reduce the search space and have shown that it compares favorably to simulated annealing and tabu search algorithms. Crisan and Muhlenbein [9] presented a new crossover operator, which takes the problem specific information into account in looking for solutions to reduce search space. Moreover, most of the studies have been done on theoretical problems [23] as benchmark cases have been too simple [10].

In this paper we present a genetic algorithm tailored for MI-FAP, which uses a new problem representation that reduces the search space from f^n to $(f!)^{\lceil n/f \rceil}$ and further improves the time complexity of the fitness function from $O(n^2)$ to $O(n^2/f)$, where f is the number of available frequencies and n is the number of stations.

The paper is organized as follows. In Section 2, we present the MI-FAP, existing models and the representations along with the associated problems. In Section 3 we first introduce the FAPSTER approach and give an algorithm for efficient partitioning of the interference graph followed by a discussion of the generated search space. We then introduce the new problem representation and discuss the efficiency of the fitness function with regard to other problem representations. In Section 4, we give the benchmarks for our GA using a real-world problem together with the GAs using other problem representations. We

then continue with a short discussion in Section 5 about the optimality of the partitions generated by the cluster discovery algorithm. In Section 6 we provide our conclusions.

2. MI-FAP

Although the frequency spectrum is continuous, carrier frequencies used for communication have to be separated from each other with a distance depending on the bandwidth of the communication link for efficiency. When two or more transmitters operate at the same frequency, or at frequencies that are not separated enough, they disturb each others communication. This phenomenon is known as "interference" and it makes the frequency spectrum a limited natural resource.

FAP in general is the problem of optimizing the use of this natural resource while keeping the interference minimal. There are several classes of the problem [1]; (1) MS-FAP (2) MO-FAP (3) MI-FAP. In MS-FAP the aim is to reduce the span of the lowest and highest used frequencies, where in MO-FAP the aim is to reduce the number of frequencies used. In MI-FAP class the aim is to find the best assignment of a fixed number of frequencies to the stations in the network which minimizes the global interference.

2.1 MI-FAP Models

Attempts to solve the FAP mainly originate from the graph coloring problem [1] based on simplified models of the radio networks. A standard approach is to model the networks as hexagonal cells with each cell containing a base station in the center as shown in Figure 1. In this model it is assumed that there exists a fixed distance *d* between the center of the two cells where a frequency can be reused without any interference between the two base stations.

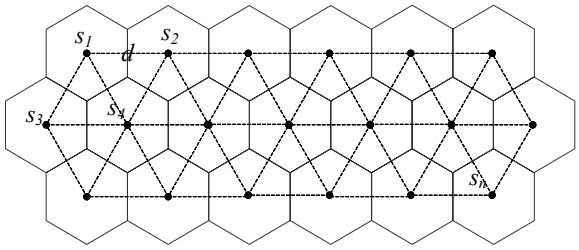

Figure 1 Hexagonal grid representation of a radio network. It is easy to convert this model to a graph by adding a vertex for each base station in the center of each hexagonal cell and an edge between the vertices in neighboring cells. After that graph coloring algorithms can be applied with an additional constraint to minimize the penalty value given in case a frequency has to be used within the re-use distance.

This hexagonal grid model is then transformed into a graph where a vertex is placed in the center of each cell representing base stations and by placing an edge between centers of the neighboring cells. The algorithms for graph coloring can then be applied to find an efficient coloring of this graph with an additional constraint to keep given penalty minimal in case the same color has to be used within the re-use distance.

A convenient model for representing interference by means of a graph $G=(V,E)$ is representing each station by a vertex $v \in V$ and connecting pairs of stations which may interfere with each other with an edge $\{u,w\} \in E$ with distance $\delta(u,w)$ denoting the penalty values or interference between the stations *u* and *w*. This graph is known as the *interference graph* or *constraint graph* and usually penalties or interferences below a certain threshold are tolerated to remove some of the edges from the graph relaxing the hardness of the problem.

2.2 Representations

There are only two problem representation techniques encountered in the literature, which can be referred as *R1* and *R2* [1] applied to MI-FAP.

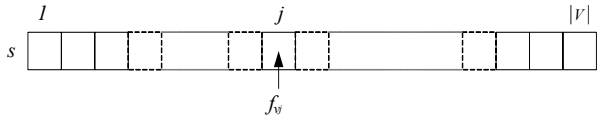

Figure 2 - Representation *R1. E*ach chromosome is a vector $s \in Z^{|V|}$ where s_j is the frequency assigned to station v_j.

The most common representation technique *R1* [21][9] is based on value encoding as shown in Figure 2, where each chromosome is a vector $s \in Z^{|V|}$ and s_j is the frequency assigned to station v_j.

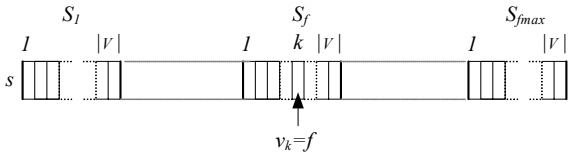

Figure 3 Representation *R2.* Binary string representation with f_{max} genes, each containing the stations assigned to the frequency *f*.

Another common representation of the problem referred as *R2*[22][19] is based on binary string representation of chromosomes with genes S_1 to S_{fmax} for each frequency from 1 to f_{max} and each containing a partition of vertices that are assigned to respective frequency as illustrated in Figure 3.

2.3 Problems

In fact, the models for MI-FAP discussed above are oversimplified. Firstly, in real life the stations are not distributed uniformly. They usually concentrate around certain areas, which are by no coincidence high population areas; city centers, shopping malls and so on where as the distribution in the rural areas and outskirts of the cities are less concentrated. The buildings and geographic structure also deform the regularity of the graph.

Secondly, when considering the electromagnetic energy it is not possible to draw such exact boundaries to the cells as shown above. As a natural phenomenon, electromagnetic energy travels in the space until it is totally absorbed or reflected. Therefore, a change of the frequency in one station affects the other stations in the network. In that sense a vertex is a neighbor to every other vertex in the graph, which usually makes the interference graph in real-life a complete graph, resulting in a big search space. Relaxation of the constraints by removing edges in a given interference graph to reduce the search space may cause algorithms to satisfy with an approximation instead of the global optimum even with exhaustive search methods.

The GAs using *R1* and *R2* look for the solution in the whole search space or reduce the search space by relaxation of the

constraints. Furthermore, the time complexity of the fitness function contributes badly to the overall performance of these algorithms. In *R2* preserving consistency of the chromosomes during crossover and mutation comprises a problem since only the fixed size subsets contain feasible solutions [22]. Otherwise, the search space gets bigger and consequently the computation time will increase.

3. FAPSTER APPROACH

Our intuition comes from the observation that the most problematic areas are the places where the station density is high. In our model, we see the physical map in clusters, where each cluster is associated with a property called *problem-level*.

The main factors that affect the problem level are the density of the stations in the cluster, the output power of the stations and the geography of the area.

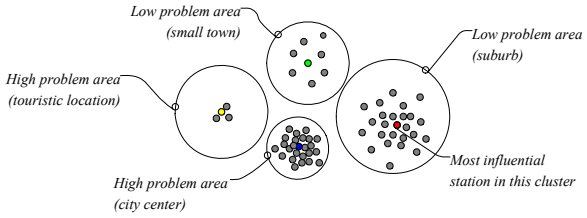

Figure 4 A real-life radio network and clusters. The station density or output power of stations in some areas is high which makes these areas more problematic than the others do.

Problem level of a cluster can be taken as the sum of the measured interference between every pair of stations within a cluster. Considering the most influential parameters that affect the interference, we observe the following case as illustrated in Figure 4. When the density of the stations in a cluster is low, then the problem level is low meaning that problem of assignment is much easier to solve.

Intuitively we think that it is beneficial to start the frequency assignment from the clusters where the problem level is high, as we want the interference level to be minimal in these areas. In fact, we want the interference level to be *0* in these clusters in the best case. The primary way to achieve this is to assign different frequencies to all stations in the same cluster. Therefore, we limit the maximum number of stations in a cluster to the maximum available frequencies. The cluster size might be adjusted for other variants of the FAP problems.

Once we have achieved the minimum interference in the problematic clusters, we think about the effect of the assignments with respect to each other. We expect the problem-level between two neighboring clusters to be less than the problem-level within a cluster, otherwise we would already include the stations causing the problem-level to be high within another cluster. Thus if there are already assignments made in two clusters, the assignments inside the clusters can be re-organized to reduce the unwanted disturbance between them.

3.1 Cluster Discovery

Given a matrix of interferences between the stations in an MI-FAP, the cluster discovery algorithm returns a list of clusters found in the interference graph.

Let $G(E,V)$ be an interference graph with V stations and $\delta(u, v)$ be the interference between the two stations u and v. A cluster $C \subseteq V$ is defined as

$$V' = V - C, \quad \forall u, v \in C \wedge \forall s, t \in V', \ \partial(u,v) > \partial(s,t).$$

The cluster discovery algorithm given in Figure 5 starts with calculating the cumulative interference around each station in the map as if they all operate at the same frequency. This gives an indication of areas (clusters) where the station density is high. If all of the clusters are isolated (ideally), the cumulative interference for each station in the same cluster has to be the same. In practice, there might be differences because of the physical barriers, which weaken the links between particular stations (isolated incomplete graph). This leads us to create smaller clusters, where the graph is closer to a complete graph. The idea of selecting the most complete graphs as clusters is to increase the efficiency in the frequency distribution, since we give all of them a different frequency. As the graph is more complete, we will have more clever distributions.

```
DISCOVER_CLUSTERS( I, M )
I: n×n interference matrix
M: maximum cluster size
List L, S
for each column Xi in the interference matrix I
    SORT Xi in descending order
    r←SUM( Xi )
    INSERT PAIR(i, r) into L
SORT L in descending order
i←0
CREATE cluster Ci
for each pair p in L
    if SIZE(Ci) = MAX-CLUSTER-SIZE then
        INSERT Ci in S
        i←i+1
        CREATE Ci
    j←INDEX(p)
    if not PROCESSED(j) = TRUE then
        INSERT j into Ci
        PROCESSED(j)←TRUE
return S
```

Figure 5 Cluster discovery algorithm

By sorting the cumulative interferences from biggest to smallest, we determine the candidate clusters. Thus beginning from the station that has the biggest cumulative interference, we look for the neighborhoods around each station with a population equal or less than the maximum available frequencies. We start with the area around the most influential station and pick the first m (where m is the cluster size with $m = \lceil n/f \rceil$) stations which are most affected by the current station. As we build the clusters, we remove the stations that we have included from the map. A station cannot be a member of two clusters. Preferably, a station will be included in a cluster, which has more relations (degrees).

Once a cluster reached to its population limit, it is added to the list of found clusters and a new cluster is created with the next most influential station, which remained on the map. After all of the

clusters are discovered, the isolated stations can be removed or assigned to any frequency, as it does not have any effect to the solution of the problem. Finally, the algorithm returns the list of the clusters discovered on the interference map.

3.2 Search Space Generated by Clustering

The search space for the problem is all possible frequency assignments for n stations and f frequencies, which contains f^n possible frequency assignments. With clustering, we partition the station into groups that contain f stations. Thus by clustering the stations, the search space reduces to $(f!)^{\lceil n/f \rceil}$, as there are only (f!) possible frequency assignments for each cluster in a total of $\lceil n/f \rceil$ clusters.

3.3 *R** Representation

We have reduced the number of possible solutions to look for drastically with our intuition for the clusters however, it is still a big solution space to search. Therefore, we apply the genetic algorithm to carry on further with the search after we encode the problem using the new representation which we introduce here as *R**. This encoding will enable us to search the state-space of the permutations of the assignments within clusters.

In *R** representation each gene in a chromosome represents a cluster, which is discovered by the cluster discovery algorithm discussed earlier. Each locus in a gene corresponds to a frequency and each nucleotide contains the station to which that frequency is assigned as illustrated in Figure 6. The order of the frequencies is identical across all genes in all chromosomes.

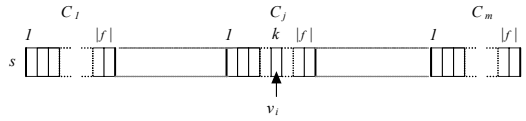

Figure 6 Representation *R. A chromosome contains $m=\lceil n/f \rceil$ genes C_i and each gene represents a cluster with maximum f stations as discovered by the cluster discovery algorithm. Each locus k in a gene contains the station, which is assigned to frequency k.**

This way each nucleotide, which holds the same location across the chromosome, contains the stations that share the same frequency. Once the clusters are assigned to the genes, the locations and the contents of the genes are always the same across all of the chromosomes. Evidently, the order of the nucleotides of the same gene in different chromosomes may vary. For the sake of efficiency, all the genes are chosen to be of the same size.

3.4 Fitness Function

The fitness or evaluation function for the MI-FAP calculates the overall interference for the given individual, as the objective is to find the minimum for this value. Since this function has to be evaluated for every individual in the population during the execution of the genetic algorithm, it is important to minimize the time complexity of this function to reduce the solution time. In that sense any small improvement to the calculation of the fitness, will have a great impact on the overall performance of the algorithm. Although there are schemes like caching the fitness values to avoid unnecessary calculations, it is the problem representation, which determines how the evaluation function will look like. The representation *R** used in our algorithm enables us to improve the time complexity of the evaluation function with regard to *R1* and *R2* representations. Our representation is compact and efficient since the stations are grouped by the frequency that they use in chromosomes.

A typical evaluation function for *R1* as in Lai and Coghill [21] or Crisan and Muhlenbein [9] is given in Figure 7. It is easy to see that this function has $O(n^2)$ time complexity, if we assume that the given interference graph is complete.

```
for each station v_i in chromosome, i from 1 to n
  for each station v_j in chromosome, j from 0 to n
      INTERFERENCE += I(i,j)
```

Figure 7 A typical evaluation function for *R1* with $O(n^2)$ complexity computes the interference between every pair of stations encoded in the chromosome.

In representation *R2* the stations are binary encoded in each gene. A naïve implementation of the evaluation function for this representation requires going through every pair of bits in the gene and lookup the interference value as illustrated in Figure 8. The evaluation time of this function is minimum f times higher than the function for *R1*. A more efficient version of this function needs an additional decoding step to collect the stations per each frequency before computing the interference between each pair.

```
for each gene g_i in chromosome, i from 1 to f
  for each station v_j in gene g_i, j from 1 to n-1
    for each station v_k in gene g_i, k from j+1 to n
      if v_j and v_k use frequency i then
        INTERFERENCE += I(j,k)
```

Figure 8 A typical evaluation function for *R2* needs f times more computation time than *R1* due to decoding of bits.

In *R** the evaluation function given in Figure 9 only checks for the interference between the stations that belong to different clusters as the total interference in a cluster is arranged to be *0*.

```
for each cluster C_i in chromosome, i from 0 to ⌈n/f⌉-1
  for each cluster C_j in chromosome, j=i+1 to ⌈n/f⌉
    for each frequency k, k=0 to f
      INTERFERENCE += I( C_i[k] C_j[k])
```

Figure 9 The evaluation function for *R with $O(n^2/f)$ complexity removes intra-gene interference calculations as the interference in a cluster is arranged to be *0*.**

This improves the time complexity to $O(n^2/f)$ compared to evaluation functions for *R1* and *R2* because we only calculate the interference for the stations encoded in pairs of genes and remove unnecessary intra-gene interference calculations.

3.5 Genetic Algorithm

The genetic algorithm is based on the simple genetic algorithm given by Holland [17] with steady state defined by Syswerda [25]. The algorithm given in Figure 10 starts with discovering the clusters in the network, then the discovered clusters are encoded using *R** where each gene represents a cluster and each locus in a gene represents a frequency. Initial population is created by assigning frequencies randomly to each station in the cluster for every cluster.

```
FAPSTER(I, M)
    t←0
    S←DISCOVER-CLUSTERS( I, M)
    ENCODE(S)
    INIT-POPULATION (P_t )
    while not STOP()
        C ← SELECT-PARENTS (P_t)
      Z←RECOMBINE (C)
        MUTATE (Z)
        t = t + 1 ;
        UPDATE-POPULATION (Pt, Z)
    Z←BEST( P_t)
    return DECODE(Z);
```

Figure 10 FAPSTER (FAP*) Algorithm. Starts with discovering the clusters in the interference graph then encodes this to a primordial chromosome using random assignments.

Selection operator uses the *linear ranking* proposed by Goldberg [18]. Whitley [28] has shown that linear ranking gives better results than proportional selection. Our tests have confirmed that truncated linear ranking produces good results for the problem instance we have used.

The algorithm crosses over using the best individual of the population with another individual selected in *roulette wheel* fashion. Although Dorne and Hao [12] states that *One-point Crossover, Two-point Crossover* or *Uniform-crossover* don't perform well in FAP, our empirical studies have shown that *One-point Crossover* gives better results than *Uniform-crossover* or *Two-point Crossover* in this encoding. In order to preserve the consistency of the chromosomes without adding too much computational burden, we let the crossover operate only at gene level in our GA.

Mutation is simply achieved by swapping contents of two loci in two different genes in the chromosome. Since the crossover only recombines the genes not alleles, mutation operator has a crucial role to provide variety in the population. Therefore, additional parameters for controlling the probability and the impact of the mutation are included in the algorithm. The parameter mutation impact determines the number of mutations to happen on a gene and the mutation factor determines how many genes are affected from the mutation. A mutation probability of *%52* has given good results with the test cases in our algorithm. After the mutation a new population is created by replacing the weakest individuals in the previous population.

The algorithm stops when the predefined condition (e.g. achieving a target fitness value or reaching to a number of generations) is satisfied and the contents of the best individual in the current population is decoded and returned as the solution of the encoded MI-FAP instance.

4. BENCHMARKS

We have used a real-life problem for benchmarking obtained from the measurements of a GSM network operator in Belgium with *500* base stations and *60* available frequencies. We have applied genetic algorithms *GA1* using representation *R1, GA2* using *R2* and *FAP** using *R** to the problem and observed the fitness evaluation over generations. The results of the three algorithms are plotted on a graph as shown in Figure 11 and compared.

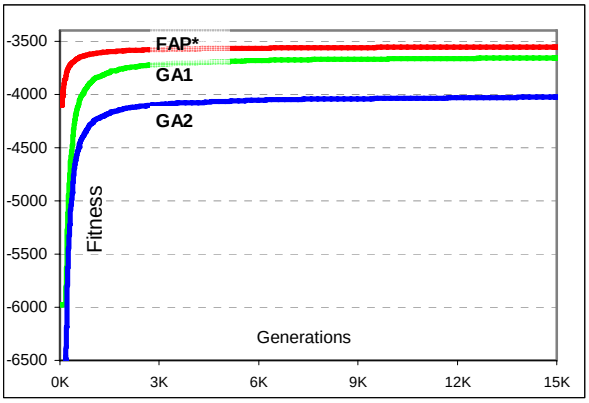

Figure 11 Comparison of genetic algorithms using *R1, R2* and *R. Algorithm *FAP** starts already with a good assignment due to the clustering and converges to the solution (*-3552*) with relatively less generations.**

Given enough iterations all of the three algorithms converge to the same predetermined fitness value (-3552) obtained by applying another local search algorithm to the problem. However we have observed that *FAP** algorithm starts with a very good assignment due to the clustering and furthermore takes relatively less generations to reach the solution in comparison to the other algorithms.

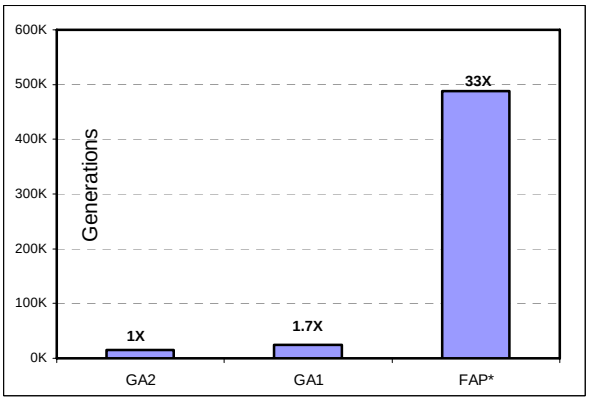

Figure 12 Comparison of evaluation speeds. FAP* algorithm computes *33* times more generations than *GA2* and *19* times more generations than *GA1* during the same amount of time thanks to the improvement in time complexity of the fitness function.

We have also compared the evaluation speeds of each algorithm in terms of generations as shown in Figure 12. The results have confirmed that the fitness calculation function for *R** speeds up the evaluation up to *33* times thanks to the improvement in time complexity of the fitness function by removing intra-gene interference calculations.

5. DISCUSSION

We would like to show that the partitioning generated by the *Cluster Discovery Algorithm* presented in Section 3.1 is optimal. Our intuition for the clustering have been empirically given good results and also reduced the time for computing the fitness of the individuals. It is our aim to motivate the idea of clustering formally using the graph theoretical notions. There has been many

studies in solving graph coloring problems by partitioning so that the ideas presented here may be quite obvious to some readers or may have already been proven by other authors.

Let C_i and C_j be two clusters obtained by the cluster discovery algorithm in V with $\sum_{u,v \in C_i} \partial(u,v) > \sum_{s,t \in C_j} \partial(s,t)$ then

$$\forall u, v \in C_i, \forall s \in C_j, \partial(u,v) > \partial(u,s).$$

This simply states that if the sum of the internal interferences in a cluster is bigger than the other one, then the interference between the stations of the two can never be bigger than the interference between the stations of the bigger cluster. Proof follows from the definition of cluster and the algorithm. The algorithm starts with discovering the stations, which has the biggest interference between them, and then it discards the found stations from the list of the stations. Assuming that there exist two stations, which have the biggest interference between them and not included in the first cluster, contradicts with the way the clusters are built by the algorithm.

We would like to confirm that given *f* frequencies there's no other partitioning other than clustering that minimizes the total interference. If we start with two available frequencies, we see that assigning the stations, which have the higher interference between them, minimizes the total interference. Considering this with the clustering algorithm, it is the case where the cluster size is two and the first cluster contains the two stations that have the highest interference. Therefore, the members of the first cluster have to be assigned different frequencies. We can extend this idea inductively to *f* frequencies and observe that assigning the members of the first cluster minimizes the total interference. Similarly, if the assignment in the first cluster is optimal we can consider the second cluster with the highest interference level and so on. This way we see that assigning different frequencies to the members of the cluster minimizes the total interference. In our algorithm, we achieve this by making the maximum cluster size *f*. Since for every cluster C_i and $|C_i| \leq f$ the interference for each cluster is minimal.

6. CONCLUSIONS

We have introduced a genetic algorithm using a new problem representation for the MI-FAP and shown empirically that it obtains better results than the algorithms based on previous problem representations referenced as *R1* and *R2* in the literature. We have also presented an algorithm for efficient partitioning of a given interference graph for a smaller search space in comparison to the conventional graph coloring based GAs for FAP. Our representation of the problem enables us to reduce the search space from f^n to $(f!)^{\lceil n/f \rceil}$. Furthermore, it improves the time complexity of the fitness calculation from $O(n^2)$ to $O(n^2/f)$ with regard to the other problem representations which results up to consumption of 33 times less time for the same amount of population generations.

In this paper we have used a real-life MI-FAP instance obtained from the measurements of a network operator in Belgium for benchmarking, however, the performance and the efficiency of the algorithm needs further to be explored with the standard benchmarking instances like CALMA or Philadelphia for comparison with the other algorithms proposed.

The algorithm presented here uses one-point crossover and may further be improved with other crossover strategies for permutation encodings such as *Order Crossover* or *Partially Matched Crossover* and it has been proposed by Valenzuela et al. [26] that the Cycle Crossover gives good results in their permutation encoding based algorithm for MS-FAP.

7. ACKNOWLEDGMENTS

We would like to thank Mr. Tsuyoshi Okita for his valuable support during the preparation of this paper.

8. REFERENCES

[1] Aardal, K.I., Hoesel, C.P.M., Koster, A.M.C.A., Mannino, C., & Sassano, A.. Models and solution techniques for the frequency assignment problem. Berlin, Germany: Konrad-Zuse-Zentrum für Informationstechnik (2001).

[2] D. Beckmann and U. Killat, Frequency planning with respect to interference minimization in cellular radio networks, Tech. Report TD(99) 032, COST259, Vienna, Austria, Jan. 1999.

[3] D. Beckmann and U. Killat, A new strategy for the application of genetic algorithms to the channel assignment problem, IEEE Transactions on Vehicular Technology 48 (1999), no. 4, pp. 1261-1269.

[4] R. Borndorfer, A. Eisenblatter, M. Grotschel, and A. Martin, The orientation model for frequency assignment problems, Tech. Report TR 98-01, Konrad-Zuse-Zentrum fur Informationstechnik Berlin (1998).

[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, Applied Decision Technologies (ADT'95) (London), 1995.

[6] A. Capone and M. Trubian, Channel assignment problem in cellular systems: a new model and a tabu search algorithm, IEEE Transactions on Vehicular Technology 48 (1999), no. 4, pp. 1252-1260.

[7] D. J. Castelino, S. Hurley, and N. M. Stephens, A tabu search algorithm for frequency assignment, Annals of Operations Research 63 (1996), pp. 301-319.

[8] K.-N. Chang and S. Kim, Channel allocation in cellular radio networks, Computers and Operations Research 24 (1997), 849-860.

[9] C. Crisan and H. Muhlenbein, The breeder genetic algorithm for frequency assignment, Lecture Notes in Computer Science 1498 (1998), 897-906.

[10] Crisan, Muhlenbein, The frequency assignment problem: A look at the performance of evolutionary search, Lecture Notes in Computer Science 1363 (1998), 263-274.

[11] W. Crompton, S. Hurley, and N.M. Stephens, A parallel genetic algorithm for frequency assignment problems, Proceedings IMACS/IEEEInt. Symp. on Signal Processing, Robotics and Neural Networks (Lille, France), April 1994, pp. 81-84.

[12] R. Dorne and J.-K. Hao, An evolutionary approach for frequency assignment in cellular radio networks, IEEE Int.

Conference on Evolutionary Computing (Perth, Australia), 1995.

[13] R. Dorne and J.-K. Hao, Constraint handling in evolutionary search: A case study of the frequency assignment, Lecture Notes in Computer Science 1141 (1996), 801-810.

[14] M. Duque-Anton, D. Kunz, and B. Ruber, Channel assignment for cellular radio using simulated annealing, IEEE Transactions on Vehicular Technology 42 (1993), 14-21.

[15] A. Gamst, Some lower bounds for a class of frequency assignment problems, IEEE Transactions on Vehicular Technology 35 (1986), 8-14.

[16] A. Gamst and W. Rave, On frequency assignment in mobile automatic telephone systems, Proceedings of GLOBECOM'82, IEEE (1982).

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

[18] D. E. Goldberg. Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley, 1989, ISBN 0-201-15767-5.

[19] F. J. Jaimes-Romero, D. Munoz-Rodriguez, and S. Tekinay, Channel assignment in cellular systems using genetic algorithms, Proceedings of the 46th IEEE Vehicular Technology Conference (Atlanta, USA), 1996, pp. 741-745.

[20] A. Knalmann and A. Quellmalz, Solving the frequency assignment problem with simulated annealing, IEEE conference publication 396 (1994), 233-240.

[21] W. K. Lai and G. G. Coghill, Channel assignment through evolutionary optimization, IEEE Transactions on Vehicular Technology 45 (1996), 91-95.

[22] C. Y. Ngo and V. O. K. Li, Fixed channel assignment in cellular radio networks using a modified genetic algorithm, IEEE Transactions on Vehicular Technology 47 (1998), 163-171.

[23] Renaud, D., Caminada, A. (1997) Evolutionary methods and operators for the frequency assignment problem. SpeedUp, 11, 27-32

[24] K.N. Sivarajan, R.J. McEliece, and J.W. Ketchum, Channel assignment in cellular radio, Proceedings of the 39th IEEE Vehicular Technology Conference, 1989, pp. 846-850.

[25] G. Syswerda. Uniform crossover in genetic algorithms. In J.D. Schaffer, edi tor. Proceedings 3rd International Conference on Genetic Algorithms pp. 2-9. Lawrence Erlhaurn Associates 1989.

[26] C. Valenzuela, S. Hurley, and D. H. Smith, A permutation based genetic algorithm for minimum span frequency assignment, Lecture Notes in Computer Science 1498 (1998), 907-916.

[27] J. P. Warners, T. Terlaky, C. Roos, and B. Jansen, A potential reduction approach to the frequency assignment problem, Discrete Applied Mathematics 78 (1997), 251-282.

[28] Whitley D. ed J Schaffer Proc. 3rd Int. Conf. on Genetic Algorithms (Fairfax, VA, June 1989) The GENITOR Algorithm and Selective Pressure: Why Rank-Based Allocation of Reproductive Trials is Best , San Mateo, CA: Morgan Kaufmann, 1989