International Journal of Computer
Science and Engineering (IJCSE)
ISSN (P): 2278–9960; ISSN (E): 2278–9979
Vol. 12, Issue 2, Jul–Dec 2023; 1–10

International Academy of Science, Engineering and Technology
Connecting Researchers; Nurturing Innovations

A DYNAMIC LOCAL SEARCH FOR SOLVING THE STATIC FREQUENCY ASSIGNMENT PROBLEM

Khaled Alrajhi

Research Scholar, King Khaled Military Academy, Riyadh, Saudi Arabia

ABSTRACT

This study proposes a novel approach to solve a variant of the frequency assignment problem known as the static minimum order frequency assignment problem. This problem involves assigning frequencies to a set of requests while minimizing the number of frequencies used. This approach solves the static problem by modeling it as a dynamic problem through dividing this static problem into smaller sub-problems, which are then solved in turn in a dynamic process using an improved local search algorithm. Several novel and existing techniques are used to improve the efficiency of this approach. This includes a technique that aims to determine a lower bound on the number of frequencies required from each domain for a feasible solution to exist in each sub-problem, based on the underlying graph coloring model. These lower bounds ensure that the search focuses on parts of the solution space that are likely to contain feasible solutions. Another technique, called the gap technique, aims to identify a good frequency to be assigned to a given request. Our approach was tested on real and randomly generated benchmark datasets of the static problem and achieved competitive results.

KEYWORDS: *Frequency Assignment*

Article History

Received:* *02 Oct 2023* | *Revised:* *19 Oct 2023* | *Accepted:* *25 Oct 2023

INTRODUCTION

The frequency assignment problem (FAP) is related to wireless communication networks, which are used in many applications such as mobile phones, TV broadcasting and Wi-Fi. The aim of the FAP is to assign frequencies to wireless communication connections (also known as requests) while satisfying a set of constraints, which are usually related to prevention of a loss of signal quality. Note that the FAP is not a single problem. Rather, there are variants of the FAP that are encountered in practice. The static minimum order FAP (MO-FAP) is the first variant of the FAP that was discussed in the literature, and was brought to the attention of researchers by [1]. In the MO-FAP, the aim is to assign frequencies to requests in such a way that no interference occurs, and the number of used frequencies is minimized. As the static MO-FAP is NP-complete [2], it is usually solved by meta-heuristics.

Many meta-heuristics have been proposed to solve the MO-FAP including genetic algorithm (GA)[3], evolutionary search (ES)[4], ant colony optimization (ACO) [5], local search [6] and tabu search (TS)[7-9]. It can be seen from the literature that TS is a popular meta-heuristic for solving difficult combinatorial optimization problems. This generally applicable algorithm has proved to be an efficient way of finding a high quality solution for a variety of optimization problems e.g. [10]. However, existing algorithms in the literature are unable to find optimal solutions in some instances for the static MO-FAP.

In this paper, we present a novel approach associated with an improved local search (LS) algorithm using multiple neighborhood structures as introduced in[11]to solve the static MO-FAP. The proposed approach is called the dynamic local search (DLS). This approach models the static MO-FAP as a dynamic problem through dividing this problem into smaller sub-problems, which are then solved in turn in a dynamic process. In contrast, existing LS algorithms in the literature for the static MO-FAP solve it once as a whole static problem. Another technique used in the DLS approach is applying a lower bound on the number of frequencies that are required from each domain in each sub-problem for a feasible solution to exist, based on the underlying graph coloring model. These lower bounds ensure that the search focuses on parts of the solution space that are likely to contain feasible solutions. This technique was also used in [11].Experiments were carried out on the CELAR and GRAPH datasets, and the results show that our DLS approach competitive comparing with existing algorithms in the literature.

This paper is organized as follows: the next section gives an overview of the static MO-FAP. Section 3 explains how to model the static MO-FAP to a dynamic problem.

Section 4 present how the underlying graph coloring model for the static MO-FAP can be used to provide a lower bound on the number of frequencies and how this information can then be used to assist the search. In Sections5, the description of the DLS approach for the static MO-FAP is given. In Section 6, the results of this approach are given and compared with those of existing algorithms in the literature before this paper finishes with conclusions and future work.

OVERVIEW OF THE STATIC MO-FAP

The main concept of the static MO-FAP is assigning a frequency to each request while satisfying a set of constraints and minimizing the number of used frequencies. The static MO-FAP can be defined formally as follows: given

A set of requests$\mathrm{R} = \{r_1, r_2, \ldots, r_{\mathrm{NR}}\}$, where NR is the number of requests,

A set of frequencies$\mathrm{F} = \{f_1, f_2, \ldots, f_{\mathrm{NF}}\} \subset \mathbb{Z}^+$, where NF is the number of frequencies,

A set of constraints related to the requests and frequencies (described below),

The goal is to assign one frequency to each request so that the given set of constraints are satisfied and the number of used frequencies is minimized. The frequency that is assigned to requests $r_i$is denoted as f_{r_i}throughout of this study. The static MO-FAP has four variants of constraints as follows:

Bidirectional Constraints

This type of constraint forms a link between each pair of requests$\{r_{2i-1}, r_{2i}\}$, where $i = 1, \ldots, NR/2$. In these constraints, the frequencies $f_{r_{2i-1}}$ and $f_{r_{2i}}$should be distanced$d_{r_{2i-1}r_{2i}}$apart. These constraints can be written as follows:

$$\left|f_{r_{2i-1}} - f_{r_{2i}}\right| = d_{r_{2i-1}r_{2i}} \quad \text{for } i = 1, \ldots, \mathrm{NR}/2 \qquad (1)$$

Interference Constraints

This type of constraint forms a link between a pair of requests$\{r_i, r_j\}$, where the pair of frequencies $f_{r_i}$and f_{r_j}should be more than distanced$d_{r_ir_j}$apart.These constraints can be written as follows:

$$\left| f_{r_i} - f_{r_j} \right| > d_{r_ir_j} \quad \text{for } 1 \leq i < j \leq \text{NR} \qquad (2)$$

Domain Constraints

The available frequencies for each request r_i are denoted by the domain $D_{r_i} \subset F$, where $\cup_{r_i \in R} D_{r_i} = F$. Hence, the frequency which is assigned to $r_i$must belong toD_{r_i}.

Pre-Assignment Constraints

For certain requests, the frequencies have already been pre-assigned to given values i.e. $f_{r_i} = p_{r_i}$, where $p_{r_i}$is a given value.

MODELING THE STATIC MO-FAP AS A DYNAMIC PROBLEM

In the DLS approach, the static MO-FAP is broken down into smaller sub-problems, each of which is considered at a specific time period. To achieve this, each request is given an integer number between 0 and n (where n is a positive integer) indicating the time period in which it becomes known. In effect, the problem is divided into $n + 1$smaller sub-problems $P_0, P_1, \ldots, P_n$, where n is the number of sub-problems after the initial sub-problem P_0. Each sub-problem P_i contains a subset of requests which become know at time period i. The initial sub-problem $P_0$is solved first at time period0. After that, the next sub-problem $P_1$is considered at time period1 and the process continues until all the sub-problems are considered. In this study, we found that the number of sub-problems does not impact on the performance of the DTS approach for solving the static MO-FAP, so the number of sub-problems is fixed at 21 (i.e. $n = 20$).

Based on the number of the requests known at time period 0 (belonging to the initial sub-problem P_0), 10 different versions of a dynamic problem are generated. These versions are named using percentages which indicate the number of requests known at time period 0. These 10 versions are named 0%, 10%, 20%, 30%, 40%, 50%, 60%, 70%, 80%, 90% (note that 100% means all the requests are known at time period 0 and so corresponds to the static MO-FAP).

An example of how a static MO-FAP is modeled as a dynamic problem is illustrated in Figure 1, where each node represents a request, each edge a bidirectional or interference constraint and each color a time period in which a request becomes known for the first time.

After breaking the static MO-FAP into smaller sub-problems, these sub-problems will be solved in turn.

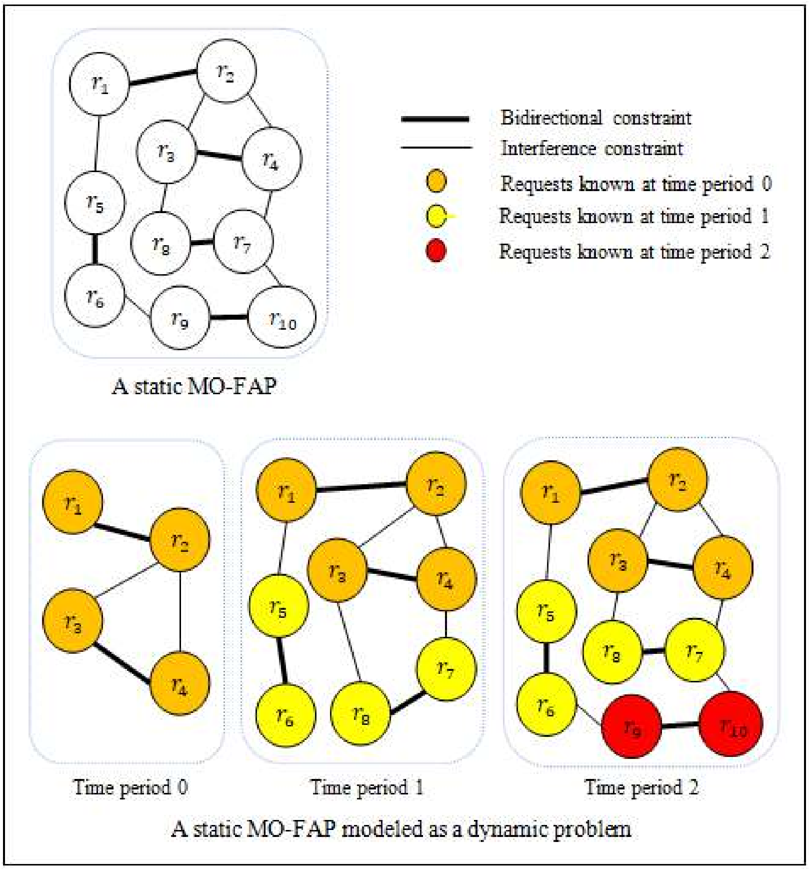

Figure 1: An Example of Modeling a Static MO-FAP as a Dynamic Problem Over 3 Time Periods.

GRAPH COLORING MODEL FOR THE STATIC MO-FAP

The graph coloring problem (GCP) can be viewed as an underlying model of the static MO-FAP [12]. The GCP involves allocating a color to each vertex such that no adjacent vertices are in the same color class and the number of colors is minimized. The static MO-FAP can be represented as a GCP by representing each request as a vertex and a bidirectional or an interference constraint as an edge joining the corresponding vertices.

One useful concept of graph theory is the idea of cliques. A clique in a graph can be defined as a set of vertices in which each vertex is linked to all other vertices. A maximum clique is the largest among all cliques in a graph. Vertices in a clique have to be allocated to a different color in a feasible coloring. Therefore, the size of the maximum clique acts as a lower bound on the minimum number of colors.

As the requests belong to different domains, the graph coloring model for each domain can be considered separately and then a lower bound on the number of frequencies that is required from each domain can be calculated. An overall lower bound on the total number of frequencies for a whole instance can also be calculated in a similar way. A branch and bound algorithm is used to obtain the set of all maximum cliques for each domain within each sub-problem.

OVERVIEW OF THE DYNAMIC LOCAL SEARCH ALGORITHM

A key decision when designing DLS is the definition of the solution space and the corresponding cost function.

Solution Space and Cost Function

In most cases, it has been found to be relatively straightforward to find solutions that satisfy the bidirectional, the domain and the pre-assignment constraints, as well as to define a neighborhood operator that moves between such solutions[13]. Hence, the solution space is defined here as the set of all possible assignments satisfying all of the bidirectional, the domain and the pre-assignment constraints. Note that the interference constraints are relaxed. Only the interference constraints are relaxed because these are the most difficult constraint to be satisfied. The cost function is defined as the number of broken interference constraints, also known as the number of violations. This configuration has been used previously in the literature e.g. [6, 9]. One of the advantages of using this configuration is that, in effect, the number of requests is halved because each request is linked with another request based on the bidirectional constraints. As a result, here requests and frequencies are considered as pairs (instead of individuals).

Based on the definition of the above solution space which relaxes the interference constraints, another problem is introduced: minimizing the number of violations with a fixed number of used frequencies. This problem is solved in the optimization phase (see Section 5.2) by DLS hybridized with three neighborhood structures, which is described in [11].

Structure of the Dynamic Local Search Algorithm

The DLS algorithm consists of two phases, namely the online assignment phase and the optimization phase.

First, the sub-problem P_0 is solved in the online assignment phase (see Section 5.3). Then, the optimization phase aims to find an optimal solution in P_0 by going through three neighborhood structures(as described in [11]). Then, the sub-problem P_1 is solved in a similar way. DLS continues in the same way until all the sub-problems are considered.

The overall structure of the DLS approach for the static MO-FAP can be described as follows.

Algorithm 1: The DLS approach

Set i to -1

While$i$ < a given number of sub-problems

Increase i by 1

Implement the online assignment phase forP_i

If (the number of violations is not equal to 0) **or** (the number of violations is equal to 0 **and** the number of used frequencies is not equal to the lower bound of the sub-problem)

Implement the optimization phase forP_i

End If

End While

Return the current solution

The Online Assignment Phase

This phase aims to solve the sub-problemsP_0, …, P_{20}in turn. Several decisions need to be made when solvingeach sub-problem:

In what order should the new requests be considered?

For the chosen pair of requests, which used feasible frequencies (if available) should be selected?

For the chosen pair of requests, which unused feasible frequencies should be selected, if necessary?

Here, the candidate pair of requests to be selected in the current sub-problem is the one which has the least number of feasible frequencies. If there is more than one, then the one which is involved in the highest number of constraints is the candidate pair of requests. If there is still more than one pair, then one of them is selected at random.

After that, the selected pair of requests is feasibly assigned a pair of used frequencies, if possible. In case there are more than one pair of used frequencies, then the most occupied one (i.e. the pair of frequencies assigned to the most pairs of requests) is selected. In case of a tie, one of them is selected randomly.

In case no used frequencies can be feasibly assigned, thena pair of unused frequencies is feasibly assigned, if possible. In case there is more than one pair of feasible frequencies, then the pair of frequencies with the largest minimum gap between it and an already used frequency is selected, this technique called gap technique. In case of a tie, one of them is selected randomly. The largest minimum gap leads to a greater probability that the interference constraints are satisfied.

In case no pair of feasible frequencies can be found for the selected pair of requests, then the pair of frequencies which results in the minimum number of violations is assigned.

EXPERIMENTS AND RESULTS

This section provides the results of the DLS algorithm for the static MO-FAP using CELAR and GRAPH datasets (available on the FAP website[1]). The performance of the DLS algorithm is compared with other algorithms in the literature.

Table 1presentsdetails of static MO-FAP instances in the CELAR and GRAPH datasets, including the numbers of requests and constraints for each instance.

In this study, the DLS results was coded using FORTRAN 95 and all experiments were conducted on a 3.0 GHz Intel Core I3-2120 Processor (2nd Generation) with 8GB RAM and a 1TB Hard Drive.

Table 1: Details of the CELAR and the GRAPH Datasets

Instance	No. Of Requests	No. Of Bidirectional Constraints	No. Of Interference Constraints	No. of Domain Constraints	No. Of Pre-assignment Constraints	Total No. Of Constraints
CELAR 01	916	458	5,090	916	0	6,464
CELAR 02	200	100	1,135	200	0	1,435
CELAR 03	400	200	2,560	400	0	3,160
CELAR 04	680	340	3,627	400	280	4,647
CELAR 11	680	340	3,763	680	0	4,783
GRAPH 01	200	100	1,034	200	0	1,334
GRAPH 02	400	200	2,045	400	0	2,645
GRAPH 08	680	340	3,417	680	0	4,437
GRAPH 09	916	458	4,788	916	0	6,162
GRAPH 14	916	458	4,180	916	0	5,554

[1]http://fap.zib.de/problems/CALMA/ (last accessed 25 December 2015).

Results of the DLS Algorithm

This section gives the results of the DLS algorithm to solve the static MO-FAP including the run time. Each instance has 10 different versions of dynamic instances based on the number of requests known at time period 0. These results are given in Table 2, where the bold results indicate the optimal solution (available on the FAP website[1]).

The DLS algorithm achieves feasible solutions for all of the instances. Moreover, this approach achieves the optimal solutions for the majority. In terms of the run time, it can be seen that the run time gradually increases with the number of requests known at time period 0 as it can be seen in Figure 2 which presents the run time on all dynamic instances based on CELAR 01, GRAPH 01 and GRAPH 02 (with different numbers of requests known at time period 0). The selected instances represent different numbers of requests and constraints.

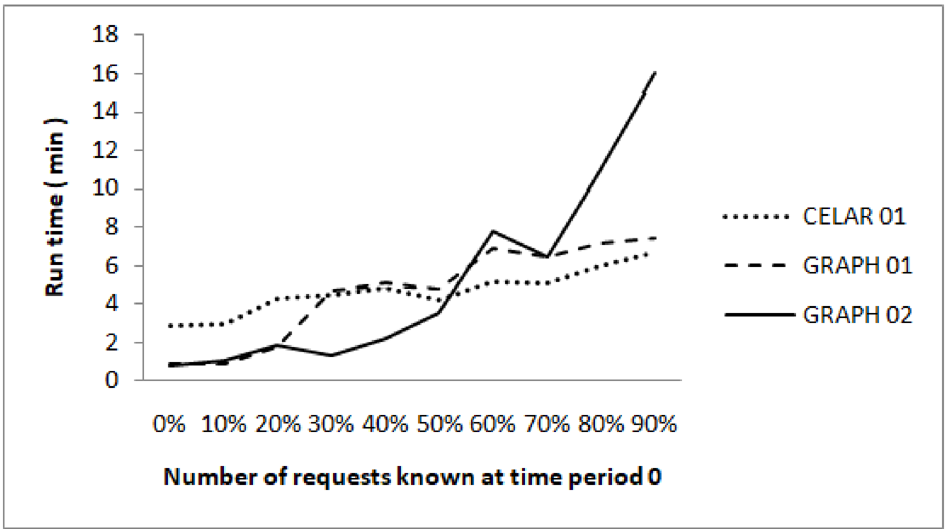

Figure 2: Run Time on all Dynamic Instances Based on CELAR 01, GRAPH 01 and GRAPH 02.

Figure 3 suggest that 0% is the best number of requests known at time period 0. Therefore, we decide to implement the DLS approach with no requests known at time period 0. This is possible because the percentage of requests that are known at time period 0 is not fixed but can be chosen as part of the approach.

Results Comparison with Other Algorithms

Here, the results of DLS and other algorithms in the literature are compared. Table 2 shows the best results,where a bold number is an optimal solution and a dash "-" means that the result is not available.

Table 2: Results Comparison of DLS with other Algorithms in the Literature

Instance	GA[3]	ES [4]	SA [6]	TS [8]	DLS	Optimal Solution
CELAR 01	20	-	**16**	18	18	16
CELAR 02	**14**	**14**	**14**	**14**	**14**	14
CELAR 03	16	**14**	**14**	**14**	**14**	14
CELAR 04	**46**	-	**46**	**46**	**46**	46
CELAR 11	32	-	24	24	28	22
GRAPH 01	20	**18**	-	**18**	**18**	18
GRAPH 02	16	**14**	-	16	18	14
GRAPH 08	-	-	-	24	20	18
GRAPH 09	28	-	-	22	24	18
GRAPH 14	14	-	-	12	**8**	8

It can be seen from Table 2that the DLS algorithm achieved competitive results compared with those of other algorithms in the literature. In fact, it achieved the optimal solution for five instances. Moreover, it is the only approach in Table 2 that achieved the optimal solution for GRAPH 14. This reflects the improvement obtained by transforming the static problem to a dynamic one. Overall, the DLS approach showed competitive performance compared with other algorithms in the literature. This suggests that solving the static problem in dynamic process by modeling it as dynamic problem leads to better results.

CONCLUSIONS AND FUTURE WORK

In this paper, we presented a novel approach for solving the static MO-FAP by local search algorithm. This approach solves this problem by modeling it as a dynamic problem through dividing this problem into smaller sub-problems, which are then solved in turn in a dynamic process using DLS algorithm. Several novel and existing techniques are used to improve the efficiency of this approach. This includes a technique that aims to determine a lower bound on the number of frequencies required from each domain for a feasible solution to exist in each sub-problem, based on the underlying graph coloring model. These lower bounds ensure that the search focuses on parts of the solution space that are likely to contain feasible solutions. Based on the results comparison, the DLS approach show competitive results comparing with other algorithms in the literature. Furthermore, this study suggests that solving the static problem in dynamic process by modeling it as a dynamic problem leads to better results by using local search algorithm.

Clearly, there are many other variants of this approach that could have been assessed such as adifferent way of modeling a static problem as a dynamic problem and using different heuristic algorithms. Further investigations of these are left as future work.

REFERENCES

1. *Metzger, B., Spectrum Management Technique presented at 38th National ORSA meeting. Detroit, MI (Fall 1970), 1970.*
2. *Garey, M.R. and D.S. Johnson, A Guide to the Theory of NP-Completeness. WH Freemann, New York, 1979.*
3. *Kapsalis, A., et al., The radio link frequency assignment problem: A case study using genetic algorithms, in Evolutionary Computing. 1995, Springer. p. 117-131.*
4. *Crisan, C. and H. Mühlenbein. The frequency assignment problem: A look at the performance of evolutionary search. in Artificial Evolution. 1997. Springer.*
5. *Parsapoor, M. and U. Bilstrup, Ant colony optimization for channel assignment problem in a clustered mobile ad hoc network, in Advances in Swarm Intelligence. 2013, Springer. p. 314-322.*
6. *Tiourine, S.R., C.A.J. Hurkens, and J.K. Lenstra, Local search algorithms for the radio link frequency assignment problem. Telecommunication systems, 2000. 13(2-4): p. 293-314.*
7. *Bouju, A., et al., Tabu search for the radio links frequency assignment problem. Applied Decision Technologies (ADT'95)(London), 1995.*
8. *Bouju, A., et al. Intelligent search for the radio link frequency assignment problem. in Proceedings of the International Conference on Digital Signal Processing, Cyprus. 1995.*

9. *Hao, J.-K., R. Dorne, and P. Galinier, Tabu search for frequency assignment in mobile radio networks. Journal of heuristics, 1998. 4(1): p. 47-62.*

10. *Glover, F. and M. Laguna, Tabu Search Applications, in Tabu Search. 1997, Springer. p. 267-303.*

11. *Alrajhi, K., J.M. Thompson, and W. Padungwech. Tabu Search Hybridized with Multiple Neighborhood Structures for the Frequency Assignment Problem. in 10th International Workshop on Hybrid Metaheuristics (HM2016) Lecture Notes in Computer Science (LNCS 9668). 2016. Plymouth, UK: Springer. p. 157-170.*

12. *Hale, W.K., Frequency assignment: Theory and applications. Proceedings of the IEEE, 1980. 68(12): p. 1497-1514.*

13. *Dorne, R. and J.-K. Hao, Constraint handling in evolutionary search: A case study of the frequency assignment, in Parallel Problem Solving from Nature—PPSN IV. 1996, Springer. p. 801-810.*