Arabian Journal for Science and Engineering
https://doi.org/10.1007/s13369-018-3393-x

RESEARCH ARTICLE - COMPUTER ENGINEERING AND COMPUTER SCIENCE

# A Neighborhood Search-Based Heuristic for the Fixed Spectrum Frequency Assignment Problem

**Umair F. Siddiqi[1] · Sadiq M. Sait[2]**

Received: 29 April 2018 / Accepted: 7 June 2018


**Abstract**

This article proposes a heuristic for the fixed spectrum frequency assignment (FS-FA) problem of telecommunications networks. A network composes of many connections, and each connection needs a frequency from the spectrum. The assignment of frequencies to the transmitters should satisfy a set of constraints. The constraints specify the separation which is necessary between frequencies of different transmitters. Violation of constraints creates interference. The goal of the FS-FA problem is to find an assignment of frequencies for the transmitters, which has minimum interference. The proposed heuristic has two main components: a local search heuristic and a compound move. The local search heuristic employs one-change moves (i.e., a move that changes the frequency of one transmitter at a time). It also employs a lookup table that classifies all possible one-change moves as positive or negative. The local search heuristic chooses positive/negative moves until it traps in a locally minimal solution. The compound-move operation shifts the local search to a new location in the search space. We can repeatedly apply the local search and compound move for many iterations. The proposed heuristic has been evaluated on the same benchmarks as used by others in the recently published literature. We have compared our algorithm with two existing tabu-search-based algorithms: dynamic-list-based tabu search (DTS) (Montemanni et al. in IEEE Trans Veh Technol 52(4):891–901, 2003. https://doi.org/10.1109/TVT.2003.810976) and heuristic manipulation technique-based TS (Montemanni and Smith in Comput Oper Res 37(3):543–551, 2010. https://doi.org/10.1016/j.cor.2008.08.006) (HMT). The solution quality of the proposed algorithm is found to be better than or equal to the HMT and DTS in 88% and 79% of test problems, respectively.

**Keywords** Frequency assignment problem · Graph coloring problem · Heuristics

## 1 Introduction

Frequency assignment is an important problem in wireless communication networks. It occurs in many different types of wireless communication systems such as satellites and cellular networks. This article proposes a heuristic for the fixed spectrum frequency assignment (FS-FA) problem. The input to an FS-FA problem is a fixed spectrum of frequencies, a set of transmitters, and a set of constraints. The constraints specify the minimum separation necessary between frequencies assigned to different transmitters. When frequencies of any two transmitters have separation lesser than the amount specified in a constraint, then they create interference [1]. The FS-FA problem can be converted to the classic graph coloring problem and is therefore an NP-hard problem [2–4]. The goal of any heuristic designed to solve the FS-FA problem must be to determine an assignment of frequencies for the transmitters that can minimize the total interference that could occur due to the violation of constraints.

Optimization heuristics are frequently used to solve the FS-FA problem. We can classify the optimization heuristics as neighborhood search and global search. The neighborhood search heuristics (also known as local search heuristics) aim to find an optimal solution within a smaller region around the initial solution. The global search heuristics, on the other

✉ Sadiq M. Sait
sadiq@kfupm.edu.sa

Umair F. Siddiqi
ufarooq@kfupm.edu.sa

1 Center for Communications and IT Research, Research Institute, King Fahd University of Petroleum and Minerals, Dhahran 31261, Saudi Arabia

2 Department of Computer Engineering, and Center for Communications and IT Research, Research Institute, King Fahd University of Petroleum and Minerals, Dhahran 31261, Saudi Arabia

Published online: 14 June 2018

hand explore a larger region of the solution space in search of an optimal solution. The FS-FA problem has been mostly solved using neighborhood search heuristics. The global search heuristics also repeatedly apply local search heuristics at different places in the search space to find the optimal solution. Therefore, the local search heuristics are critical in solving the FS-FA problem.

The classification of heuristics is not only based on their region of search (local search or global search), but they can also be classified based on other properties. Some popular classifications of heuristics are as follows: (i) population-based and single-solution-based heuristics, and (ii) memory-less and memory-based heuristics. The single-solution-based heuristics use less memory and computational resources as compared to the population-based heuristics. Examples of population-based heuristics include the genetic algorithm (GA) [5–7] and particle swarm optimization. The memory-less heuristics do not keep track of their previous moves or move attributes, whereas memory-based heuristics remember their last moves and choose new moves using their memory. Tabu search (TS) heuristic [5] belongs to the category of memory-based, single-solution local search heuristic.

Several heuristics have been proposed to solve the FS-FA problem [4,6–13]. These include population type, those that work on single solution, and those that use memory to store history. The population-based heuristics simultaneously search different locations in the search space and depend on a local search heuristic to find the locally optimal solutions at each location [4,13]. Single-solution-based heuristics, on the other hand, search one location at a time; however, they can shift to a new location using a proper move operation. The survey of literature on the application of single-solution-based heuristics to the FS-FA problem revealed that memory-based heuristics (i.e., TS-algorithm) [8,9] outperform memory-less heuristics such as simulated annealing (SA) [11] and threshold accepting (TA) [14,15].

Bandwidth coloring problem (BCP) is closely related to the FA problem. The BCP problem is also solved using TS-based [16] and neighborhood search [17] heuristics. Recently, Matic et al. [17] have proposed a heuristic that applies local search is the neighborhoods which have good solutions. It also uses a shaking operation to discover new neighborhoods. The local search heuristics for the FA problems are also useful in solving the BCPs.

The proposed heuristic employs a lookup table (LUT) that contains precomputed values of the possible increase/decrease in the interference when any transmitter changes its frequency to another one from the spectrum. However, the contents of the LUT depend on the current solution that needs to be updated after any change in the current solution. We usually need to make an incremental update to a small portion of the LUT after a change in the frequency of a transmitter. The proposed heuristic uses the LUT to choose a positive move. However, when the LUT does not have any positive moves, then it chooses a negative move. The successive application of positive and/or negative moves results in an efficient local search. The proposed heuristic also employs a compound move to shift the search to a new location in the search space.

The main contribution of the proposed heuristic is that it is a single-solution, memory-less type of heuristic that can solve the FS-FA problem and produce results competitive with the existing TS-based heuristics. The proposed heuristic contributes toward improvement in the design of heuristics in solving important industrial problems. The experiments used the same benchmarks as used by many existing heuristics [4, 8,9,13]. Although the benchmarks are academic, they provide excellent insight into the performance of the heuristics.

The organization of this paper is as follows. Section 2 describes some basic concepts and definitions related to FS-FA problem. Section 3 presents the proposed heuristic in detail. Section 4 shows the experimental results and a discussion on them. The last section contains the conclusion and future work.

## 2 Problem Formulation

In this section, we present a model of the FS-FA problem from the graph theory perspective [1]. The model consists of a weighted and undirected quadruple graph $G(V, E, D, P)$. The set $V$ contains the vertices of the graph, and each vertex represents a transmitter. The set $V$ contains a total of $m$ transmitters, and its representation is as follows: $V = \{v_0, v_1, ..., v_{m-1}\}$. The set $(E)$ contains the edges of the graph, and the edges represent the separation constraints between the frequencies of the transmitters. An edge $(v_i, v_j) \in E$ indicates that a separation constraint exists between vertices $v_i$ and $v_j$. The sets $D$ and $P$ contain weights of the edges, and $E$ has a one-to-one correspondence with $D$ and $P$. Any element $d_{i,j} \in D$ corresponds to the edge $(v_i, v_j) \in E$. Similarly, any element $p_{i,j} \in P$ corresponds to the edge $(v_i, v_j) \in E$. The $D$ contains the values of the separation constraints and the $P$ contains the penalties (i.e., an increase in interference of the solution) for the violation of constraints. The model also contains a set $F$ which contains up to $N$ distinct frequencies, i.e., $F = \{f_0, f_1, ..., f_{n-1}\}$. The frequencies for the transmitters should be selected from $F$.

The solution of the FS-FA problem is an ordered set that contains elements equal to the number of transmitters (i.e., $m$). The solution is represented by $\Delta = \{\delta_0, \delta_1, ..., \delta_{m-1}\}$, where $\delta_k$ is the frequency of the transmitter $v_k$.

An interference could exists between transmitters $v_i$ and $v_j$, if $\exists(v_i, v_j) \in E$ and the frequencies of $v_i$ and $v_j$ violates the separation constraint $d_{i,j} \in D$. The mathematical

**Input:** $G(V, E, D, P)$, $F$, $\beta$, $\rho_1$, $\rho_2$, $\gamma$
1: **Initialization:** Initialize the solution ($\Delta$) to a random value, $\Delta_{\text{best}} = \Delta$, and $\alpha = \emptyset$
2: **while** Stopping criterion is not reached **do**
3: **Local Search:** Apply the local search heuristic with the initial solution equal to $\Delta$.
4: **Update Archive:** Conditionally copy the solution ($\Delta$) into the archive ($\alpha$) and update $\Delta_{\text{best}}$
5: **Shifting:** Apply a compound move to shift the search to a new location in the search space
6: **endwhile**
7: **return** $\Delta_{\text{best}}$

**Fig. 1** Overview of the proposed heuristic

expression to compute the interference is as follows.

$$I(v_i, v_j) = \begin{cases} p_{i,j} & \text{if } |\delta_i - \delta_j| < d_{i,j} \\ 0 & \text{otherwise} \end{cases} \quad \forall (v_i, v_j) \in E \tag{1}$$

The cost of the solution (or total interference) is the sum of the penalties of all violating constraints. We can determine the cost of a solution ($\Delta$) using (2).

$$\text{cost}(\Delta) = \sum_{(v_i, v_j) \in E} I(v_i, v_j) \tag{2}$$

$$\text{Minimize}(\text{cost}(\Delta)) \tag{3}$$

## 3 Proposed Heuristic for the FS-FA Problem

In this section, we present details of our proposed heuristic for solving the FS-FA problem. Figure 1 depicts an overview of the heuristic. The input consists of the quadruple graph ($G(V, E, D, P)$), spectrum ($F$) and four parameters ($\beta$, $\rho_1$, $\rho_2$, and $\gamma$). The value of $\beta$ controls the termination of the local search, while $\rho_1$ and $\rho_2$ are the probabilities used in the compound move. The parameter $\gamma$ sets the size of the archive which has an important role in the compound move. The remaining part of this section describes each step in detail.

### 3.1 Initialization

In this step, we initialize the solution ($\Delta$) by assigning random frequencies to the transmitters. The proposed heuristic also employs an archive ($\alpha$) to store up to $\gamma$ solutions. The archive ($\alpha$) initializes to an empty set. The proposed heuristic also uses elitism and keeps a copy of the best seen solution thus far in $\Delta_{\text{best}}$.

### 3.2 Local Search

This subsection contains the description of the proposed single-solution, memory-less local search heuristic for the FS-FA problem. The proposed local search heuristic employs a lookup table (LUT) to reduce computations. The move operations in our local heuristic change the frequency of only one transmitter. The total number of moves is equal to $mN$, where $m$ is the number of transmitters, and $N$ is the number of frequencies in the spectrum. The evaluation of all $mN$ moves in every iteration is time consuming. The change in cost of only a few moves need to be evaluated in every iteration. When we change the frequency of transmitter $v_i$, then only the moves that belong to the transmitters that have a constraint with $v_i$ need re-evaluation. In this way, maintaining an LUT helps to reduce the computation and improves the runtime of the heuristic. The LUT has the number of columns equal to the number of frequencies to be assigned (that is, the size of the spectrum) and number of rows equal to the number of transmitters. An element $(i, j)$ stores a positive penalty value if the assignment of the frequency $f_j$ to the transmitter $v_i$ violates any separation constraint. The contents of the LUT depend on the value of the current solution ($\Delta$). Therefore, the LUT requires an update in case of any change in the current solution. The value of each entry in the LUT can be determined using (4) and (5).

$$J(i, j, k) = \begin{cases} p(i, k) & \text{if } |f_j - \delta_k| < d_{i,k} \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

$$\text{LUT}(i, j) = \sum_{k=0}^{n-1} J(i, j, k) \tag{5}$$

In (4), $J(i, j, k)$ returns a zero value if $\nexists\ (v_i, v_k \in E)$ and returns a nonzero value if the assignment of frequency $f_j$ to $v_i$ violates the separation constraint specified by the edge $(v_i, v_k)$, where the frequency of the $v_k$ is $\delta_k$. In (5), we compute the value to store at the location $(i, j)$ in the LUT. We can use a nested for loop in which $i$ varies from 0 to $m$-1 and $j$ varies from 0 to $n$-1 to determine the contents of the complete LUT.

The proposed heuristic uses the LUT to choose a move. A move here is changing the frequency assigned to a transmitter in the current solution ($\Delta$). A move is represented by a 2-tuple $(i, j)$, where $i$ refers to the transmitter $v_i$ and $j$

**Fig. 2** Local search heuristic

**Input:** $G(V, E, D, P)$, $F$, $\beta$, $\rho_1$, $\rho_2$, $\Delta$

1: Initialize the LUT
2: $\Delta_T = \Delta$, $i = 0$
3: **while** $i < \beta$ **do**
4:     **if** Any positive-move exists in the LUTs **then**
5:         Randomly selects a positive-move and apply it to $\Delta$
6:         **if** cost($\Delta$) $<$ cost($\Delta_T$) **then**
7:             $\Delta_T = \Delta$, $i = 0$
8:         **else**
9:             $i = i + 1$
10:         **endif**
11:     **else**
12:         Selects a negative-move with the help of the LUT and apply it to $\Delta$
13:         $i = i + 1$
14:     **endif**
15:     Update the LUT
16: **endwhile**
17: $\Delta = \Delta_T$
18: **return** $\Delta$

refers to the frequency $f_j \in F$. We can divide the moves in the LUT into two types: (i) positive moves; and (ii) negative moves. A positive move $(i, j) \in$ LUT meets the condition that $\text{LUT}(i, j) < \text{LUT}(i, \delta_i)$, where $\delta_i$ is the current frequency of $v_i$. When we apply the move $(i, j)$ to the current solution, then the frequency of the transmitter $v_i$ (i.e., $\delta_i \in \Delta$) changes to $f_j$. The cost of the solution decreases after the application of a positive move. The negative moves include all the moves in the LUT that do not belong to the type of positive moves. A negative move also changes the frequency of a transmitter, but the cost of the solution can either remains unchanged or increases. During the execution of the heuristic, the positive moves might not always be available. However, the negative moves are always available. For a transmitter $v_i$, $\xi^+(i)$ represents a set of available positive moves and $\xi^-(i)$ represents a set of available negative moves in the LUT. We can determine the elements for the sets $\xi^+(i)$ and $\xi^-(i)$ using (6) and (7).

$$\xi(i)^+ = \{(i, j), \quad \text{s.t. } \text{LUT}(i, j) < \text{LUT}(i, \delta_i)\} \tag{6}$$

$$\xi(i)^- = \{(i, j), \quad \text{s.t. } \text{LUT}(i, j) \geq \text{LUT}(i, \delta_i)\} \tag{7}$$

A complete description of the proposed local search heuristic is depicted in Fig. 2. The input to the heuristic consists of the following items: (i) current solution ($\Delta$); (ii) A set of frequencies ($F$); (iii) graph that models the problem ($G$); and (iv) three parameters ($\beta$, $\rho_1$, and $\rho_2$). The value of $\beta$ should be a positive integer, and the values of remaining two parameters should be real numbers between 0 and 1.

In the pseudo-code of Fig. 2, the execution flow is as follows. In line 1, we initialize the LUT by using Eqs. (4) and (5). In line 2, we introduce two variables $\Delta_T$ and $i$ that store the best solution and the number of successive iterations with no improvement in the cost of the solution, respectively. The while loop implements the termination condition of the local search heuristic. In line 4, we use (8) to determine the availability of any positive move in the LUT. In (8), the set $\xi^+$ contains all positive moves present in the LUT. When $\xi^+$ is empty, then there does not exist any positive move. The set $\xi^-$ in (9) contains all negative moves present in the LUT.

$$\xi^+ = \cup_{i=0}^{n-1} |\xi^+(i)| \tag{8}$$

$$\xi^- = \cup_{i=0}^{n-1} |\xi^-(i)| \tag{9}$$

In line 5, we randomly select a move from $\xi^+$ and update the frequency of a transmitter in the current solution ($\Delta$). The lines 6–10 consist of the following tasks: if the cost of the current solution $\Delta$ is better than the cost of the best solution $\Delta_T$, then $\Delta_T$ should be updated, and the variable $i$ should also be re-initialized to zero. Otherwise, we increment the variable $i$. If the condition on line 4 returns false, then we execute the steps of lines 11–14. In line 12, the selection of a negative move from $\xi^-$ consists of the following steps.

1. Compute $\alpha_v = \frac{\sum_{(i,j)\in\xi^-} |\text{LUT}(i,j) - \text{LUT}(i,\delta_i)|}{|\xi^-|}$
2. Create a subset of $\xi^-$ as $\xi_s^- = \{(i, j), s.t. |\text{LUT}(i, j) - \text{LUT}(i, \delta_i)| \leq \alpha_v\}$
3. Randomly select a move from $\xi_s^-$

The selection of a negative move comprises three steps. In first step, the expression $\text{LUT}(i, j) - \text{LUT}(i, \delta_i)$ computes the difference between the cost of the current solution (i.e., the solution in which the frequency of the $i$th transmitter is $\delta_i$) and a modified version of the current solution in which the frequency of the $i$th transmitter is $f_j$. The $a_v$ variable stores the average value of the term $|\text{LUT}(i, j) - \text{LUT}(i, \delta_i)|$ of all moves $(i, j) \in \xi^-$. In the second step, we create a set $\xi_s^-$ that contains all moves of $\xi^-$ except the ones that increase the cost of the solution more than a certain value $\alpha_v$. In the third step, we randomly select a move from $\xi_s^-$ and apply it to the current solution.

After a positive or negative move is selected and applied, the next step is to update the LUT. LUT should be updated after any change in the current solution (as mentioned in line 15 of Fig. 2). When we change the frequency of a transmitter, then only a small portion of the LUT needs to be updated, not the entire table. Suppose we have changed the frequency of a transmitter $v_i$, then a partial update of the LUT proceeds as follows: (i) We build a set $R_i$ that contains all transmitters that have a separation constraint with $v_i$, i.e., $R_i = \{v_j, s.t., \exists(v_i, v_j) \in E\}$; (ii) We select a subset of rows of the LUT that corresponds to the transmitters present in $R_i$ (where the $j$th row of the LUT corresponds to the transmitter $v_j$); (3) We update the selected rows of the LUT using Eq. (5).

### 3.3 Update the Archive

The proposed heuristic employs an archive ($\alpha$) of $\gamma$ solutions. Until the archive is completely full, we keep adding new solutions in it. However, when the archive becomes full, then we need to replace an existing solution with the new solution. If the cost of the new solution is not better than any of the solution already stored in the archive, then the new solution is ignored. Otherwise, we first find a solution in the archive whose cost in maximum in the archive and more than the new solution, and then replace it with the new solution.

### 3.4 Shifting

The local search heuristic terminates upon encountering $\beta$ number of successive non-improving iterations. However, we can again apply the local heuristic by making some changes in the current solution ($\Delta$). The changes aim to shift the search to a new location in the search space. We can keep repeating the steps of applying the local search heuristic and shifting the search to a new locations until the runtime of the heuristic reaches its maximum limit.

We apply a compound move to shift the search to a new location. The compound move alters several elements of the solution. The following text describes the compound move.

1. Divide the transmitters of $V$ into two distinct sets, $V_i$ and $V_n$, such that $V_i \cup V_n = V$ and $V_i \cap V_n = \cap$.
2. The set $V_i$ contains transmitters that violate any constraint and $V_n$ contains all the remaining transmitters. Mathematically, it can be expressed as, $V_i = \{v_x | I(v_x, v_y) > 0 \text{ or } I(v_y, v_x) > 0\}$, where $v_x, v_y \in V$ and $V_n = V - V_i$.
3. Each transmitter in $V_i$ could change its assignment with a fixed probability $\rho_1$. The new frequency for the transmitter is determined randomly from the spectrum ($F$).
4. Each transmitter in $V_n$ could change its assignment with a fixed probability $\rho_2$. The new frequency for the transmitter is determined randomly from the spectrum ($F$). The value $\rho_2$ is usually much smaller than $\rho_1$.

## 4 Experimental Results

In this section, we present the experimental results and their analysis. We implemented the proposed heuristic using C++ and R (using Rcpp package). We used the C++ language to implement the local search heuristic and used the R language to implement the operations to maintain the archive and shift the search to new locations. The program executed on a computer having Intel Xeon 2.8 GHz processor and 64 GB of memory.

The benchmarks used in the evaluation are among the most popular ones and used by many authors in recent past [4,8,9,13]. Table 1 lists the main characteristics of the bench-

**Table 1** Characteristics of the test problems [8,9]

| Problem | $\|V\|$ | $\|E\|$ | $\hat{d}$ | $\hat{p}$ |
|---|---|---|---|---|
| AC-45-17 | 45 | 482 | 0.29 | 1.00 |
| AC-45-25 | 45 | 801 | 0.34 | 1.00 |
| AC-95-9 | 95 | 781 | 0.00 | 1.00 |
| AC-95-17 | 95 | 2298 | 0.15 | 1.00 |
| GSM-93 | 93 | 1073 | 0.28 | 1.00 |
| GSM-246 | 246 | 7611 | 0.32 | 1.00 |
| Test95 | 95 | 1214 | 1.37 | 1.00 |
| Test282 | 282 | 10,430 | 1.38 | 1.00 |
| P06-5 | 88 | 3021 | 0.39 | 1.00 |
| P06-3 | 153 | 9193 | 0.59 | 1.00 |
| P06b-5 | 88 | 3021 | 0.39 | 1.00 |
| P06b-3 | 153 | 9193 | 0.40 | 1.00 |
| GSM2-184 | 184 | 6809 | 0.20 | $8.95 \times 10^6$ |
| GSM2-227 | 227 | 10,088 | 0.18 | $9.10 \times 10^6$ |
| GSM2-272 | 272 | 14,525 | 0.16 | $7.95 \times 10^6$ |
| 1-1-50-75-30-2-50 | 75 | 835 | 0.26 | 10.81 |
| 1-2-50-75-30-4-50 | 75 | 835 | 0.62 | 11.01 |
| 1-3-50-75-30-0-50 | 75 | 835 | 0.00 | 10.97 |
| 1-4-50-75-30-2-1 | 75 | 835 | 0.25 | 1.00 |
| 1-5-50-75-30-2-100 | 75 | 835 | 0.26 | 21.35 |
| 1-6-50-75-30-0-10000 | 75 | 835 | 0.00 | 2068.48 |

marks. The first column mentions the name of the problem; the second gives the number of transmitters; the third column mentions the number of constraints; the fourth column mentions the average of the separation requirements in the constraints, and the last column mentions the average of the penalty values of the constraints. We executed a total of forty-two test cases in which each problem is solved several times with a different spectrum size.

A description on the selection of the parameter values is mentioned below. In the three problems that have $p > 1 \times 10^6$ (i.e., GSM2-184, GSM2-227, GSM2-272), we set the value of $\beta$ equal to 2000, and the value of $\gamma$ equal to 100. In the remaining problems, we set the value of $\beta$ equal to $1 \times 10^5$ and the value of $\gamma$ equal to 20. The values of the remaining two parameters ($\rho_1$ and $\rho_2$) are same in all problems. We set the value of $\rho_1$ equal to 0.05 and the value of $\rho_2$ equal to 0.02. The parameter $beta$ controls the termination of the local search heuristic. In the experiments, we found that in large problems, if the local search heuristic could not improve the solution in up to 50,000 iterations, then it is never able to improve the solution. In smaller problems, the local search traps into a local minimum within few thousand iterations. The parameter $gamma$ controls the archive size, and an increase in its size increases the runtime of the heuristic. Therefore, it should be set to the smallest value that can yield good results. A detailed investigation of the parameters values is outside the scope of this work. The parameters $\rho_1$ and $\rho_2$ are mutation probabilities; therefore, their values should be kept small.

The three problems (GSM2-184, GSM2-227, GSM2-272) have a relatively smaller value of $\beta$ because in those problems the penalties are small for some constraints and very large for others. In the local search heuristic, we filter out the negative moves that cause a large increase in the cost of the solution and hence eliminate the moves that have large penalty values. This elimination of some possible moves may lead the heuristic to quickly trap into a local minimum. Therefore, we can detect the trapping of the search into a local minimum using lesser number of successive non-improving iterations. The empirical results verified our observation and we noticed an improvement in runtime by setting a smaller value to the parameter $\beta$ in problems GSM2-184, GSM2-227, and GSM2-272. The compound move employed the parameters $\rho_1$ and $\rho_2$ and treat them like mutation probabilities. Therefore, their values should be small (0.05 and 0.02). The effect of the non-deterministic behavior of the proposed heuristic is determined by conducting twenty trials on each problem. The stopping criterion of the proposed heuristic used in the experiments is as follows: If the last 1000 iterations could not produce any improvement in the cost of the solution, then the heuristic should terminate. The FS-FA problem is an offline problem. Therefore, the runtime is not critical as long as it remains within practical limits. We set the maximum runtime to solve any problem equal to 2 h. However, most of the problems converge to their best solutions within the first 30 min.

We compared the proposed heuristic with two existing single-solution-based heuristics, which are dynamic list tabu search (DTS) algorithm and heuristic manipulation technique-based tabu search (HMT) algorithm. The DTS and HMT belong to the class of single-solution memory-based local search heuristics. Some authors have also applied single-solution memory-less heuristics such as threshold accepting (TA) to the FS-FA problem, but the results are not as good as that of DTS and HMT heuristics [8,9]. The authors of HMT and DTS heuristics conducted ten trials, and they set the maximum runtime to 45 min. However, they also mentioned that any increase in the runtime is not likely to improve the solution quality of their heuristics [8].

Table 2 shows the results of the proposed heuristic and that of the DTS [8] and HMT [9]. The results of each heuristic consist of two columns. The first column mentions the average values of the trials, and the second column mentions the best and the worst results of the trials. The results show that the proposed heuristic has obtained results better than HMT and DTS in many problems. In Fig. 3, we summarize the results of Table 2 to show the advantage of the proposed heuristic. In Fig. 3, the $y$-axis shows the percentage with which the solution of the proposed heuristic is better than that of the HMT and DTS heuristics and the numbers on the $x$-axis corresponds to different problems in the same order as in Table 2. The results reveal the following about the comparison of the solution quality of the proposed heuristic with HMT and DTS on the 42 test problems. The solution quality of the proposed heuristic is better than that of HMT and DTS in 21 and 17 problems, respectively. The proposed heuristic has a solution quality equal to that given by HMT and DTS heuristics in 16 problems. The solution quality of the proposed heuristic is also worse than that of HMT and DTS in up to 5 and 9 problems, respectively. In the problems in which the proposed heuristic performed better than the existing heuristics, the solution quality is up to 41% better than that of HMT and 38% better than that of DTS. In the problems in which the proposed heuristic performed worse than the existing heuristics, the solution quality is up to -9% lesser than that of HMT and DTS.

In Table 3, we list the problems in which the worst result of all trials of the proposed heuristic is better than the best result of the other heuristics. Under this condition, the proposed heuristic has a solution quality which is better than that of HMT or DTS heuristic in all trials. Therefore, the proposed heuristic has a clear advantage over the other heuristic (DTS or HMT). As Table 3 shows, the results of all trials of the proposed heuristic are better than the results of all trials of HMT in up to eight problems and better than the results of all trials of DTS in up to three problems.

**Table 2** Comparison with existing single-solution and memory-based optimization heuristics

| Problem | \|F\| | Proposed | | HMT [9] | | DTS [8,9] | |
|---|---|---|---|---|---|---|---|
| | | Mean | (best, worst) | Mean | (best, worst) | Mean | (best, worst) |
| AC-45-17 | 7 | 32.00 | (32, 32) | 32.00 | (32, 32) | 32.00 | (32, 32) |
| AC-45-17 | 9 | 15.00 | (15, 15) | 15.00 | (15, 15) | 15.00 | (15, 15) |
| AC-45-25 | 11 | 33.00 | (33, 33) | 33.00 | (33, 33) | 33.00 | (33, 33) |
| AC-95-9 | 6 | 31.00 | (31, 31) | 31.00 | (31, 31) | 31.00 | (31, 31) |
| AC-95-17 | 15 | 33.26 | (33, 34) | 33.00* | (33, 33) | 33.00* | (33, 33) |
| AC-95-17 | 21 | 10.00 | (10, 10) | 10.00 | (10, 10) | 10.00 | (10, 10) |
| GSM-93 | 9 | 32.00* | (32, 32) | 33.00 | (32, 34) | 32.20 | (32, 33) |
| GSM-93 | 13 | 7.00 | (7, 7) | 7.00 | (7, 7) | 7.00 | (7, 7) |
| GSM-246 | 21 | 82.10 | (80, 83) | 80.20* | (79, 81) | 80.60 | (79, 82) |
| GSM-246 | 31 | 26.58 | (25, 27) | 26.30 | (25, 27) | 26.10* | (25, 27) |
| Test95 | 36 | 8.00 | (8, 8) | 8.00 | (8, 8) | 8.00 | (8, 8) |
| Test282 | 61 | 58.05 | (55, 59) | 53.40 | (51, 56) | 53.20* | (51, 55) |
| Test282 | 71 | 30.00 | (29, 31) | 29.40 | (27, 30) | 29.30* | (27, 30) |
| Test282 | 81 | 12.06 | (10, 14) | 12.20 | (11, 13) | 11.90* | (10, 13) |
| P06-5 | 11 | 133.00 | (133, 133) | 133.00 | (133, 133) | 133.00 | (133, 133) |
| P06-3 | 31 | 115.00 | (115, 115) | 115.00 | (115, 115) | 115.00 | (115, 115) |
| P06b-5 | 21 | 52.00 | (52, 52) | 52.00 | (52, 52) | 52.00 | (52, 52) |
| P06b-5 | 31 | 25.00 | (25, 25) | 25.00 | (25, 25) | 25.00 | (25, 25) |
| P06b-3 | 31 | 112.00 | (112, 112) | 112.00 | (112, 112) | 112.00 | (112, 112) |
| P06b-3 | 71 | 26.00 | (26, 26) | 26.00 | (26, 26) | 26.00 | (26, 26) |
| 1-4-50-75-30-2-1 | 6 | 71.00 | (71, 71) | 71.00 | (71, 71) | 70.90* | (70, 71) |
| 1-4-50-75-30-2-1 | 10 | 19.00 | (19, 19) | 19.00 | (19, 19) | 19.00 | (19, 19) |
| GSM2-184 | 39 | 5414.57* | (5322, 5571) | 5642.80 | (5481, 5758) | 5598.80 | (5447, 5689) |
| GSM2-184 | 49 | 874.00* | (874, 874) | 1073.40 | (999, 1143) | 1043.60 | (874, 1120) |
| GSM2-184 | 52 | 162.00* | (162, 162) | 277.90 | (186, 311) | 260.60 | (162, 287) |
| GSM2-227 | 29 | 60,186.63* | (58, 542, 62,734) | 68,077.70 | (61,586, 70,105) | 66,510.00 | (61,586, 70,105) |
| GSM2-227 | 39 | 9890.86* | (9529, 10,493) | 11,170.30 | (10,979, 11,276) | 10,897.70 | (10,550, 11,164) |
| GSM2-227 | 49 | 2315.06* | (2170, 2559) | 2649.10 | (2459, 2828) | 2613.10 | (2459, 2828) |
| GSM2-272 | 34 | 56,336.68* | (53,962, 59,069) | 60,473.20 | (57,715, 67,025) | 58,691.40 | (56,128, 64,353) |
| GSM2-272 | 39 | 28,878.50 | (28,046, 30,518) | 28,484.30* | (27,416, 29,323) | 28,488.20 | (27,416, 29,307) |
| GSM2-272 | 49 | 7967.95 | (7909, 8796) | 8043.80 | (7785, 8411) | 7946.70* | (7785, 8459) |
| 1-1-50-75-30-2-50 | 5 | 1242.00* | (1242, 1242) | 1254.10 | (1242, 1260) | 1253.90 | (1242, 1260) |
| 1-1-50-75-30-2-50 | 10 | 96.80* | (96, 97) | 105.30 | (101, 109) | 103.80 | (97, 109) |
| 1-1-50-75-30-2-50 | 11 | 55.00* | (55, 55) | 65.50 | (59, 69) | 66.10 | (59, 70) |
| 1-1-50-75-30-2-50 | 12 | 33.31* | (32, 36) | 39.60 | (38, 42) | 38.70 | (36, 42) |
| 1-2-50-75-30-4-50 | 9 | 669.00* | (665, 680) | 680.90 | (671, 691) | 680.60 | (671, 691) |
| 1-2-50-75-30-4-50 | 11 | 315.00* | (313, 318) | 315.00* | (313, 318) | 325.00 | (317, 335) |
| 1-3-50-75-30-0-50 | 7 | 197.00 | (194, 198) | 197.10 | (196, 198) | 196.50* | (194, 199) |
| 1-5-50-75-30-2-100 | 10 | 172.70* | (168, 179) | 191.60 | (186, 197) | 183.80 | (176, 199) |
| 1-5-50-75-30-2-100 | 12 | 59.90* | (58, 63) | 70.50 | (65, 74) | 69.30 | (63, 74) |
| 1-6-50-75-30-0-10000 | 10 | 6842.00* | (6777, 7187) | 7123.40 | (6942, 7279) | 7064.30 | (6840, 7267) |
| 1-6-50-75-30-0-10000 | 13 | 1128.15* | (1190, 1309) | 1389.60 | (1207, 1490) | 1365.20 | (1318, 1440) |

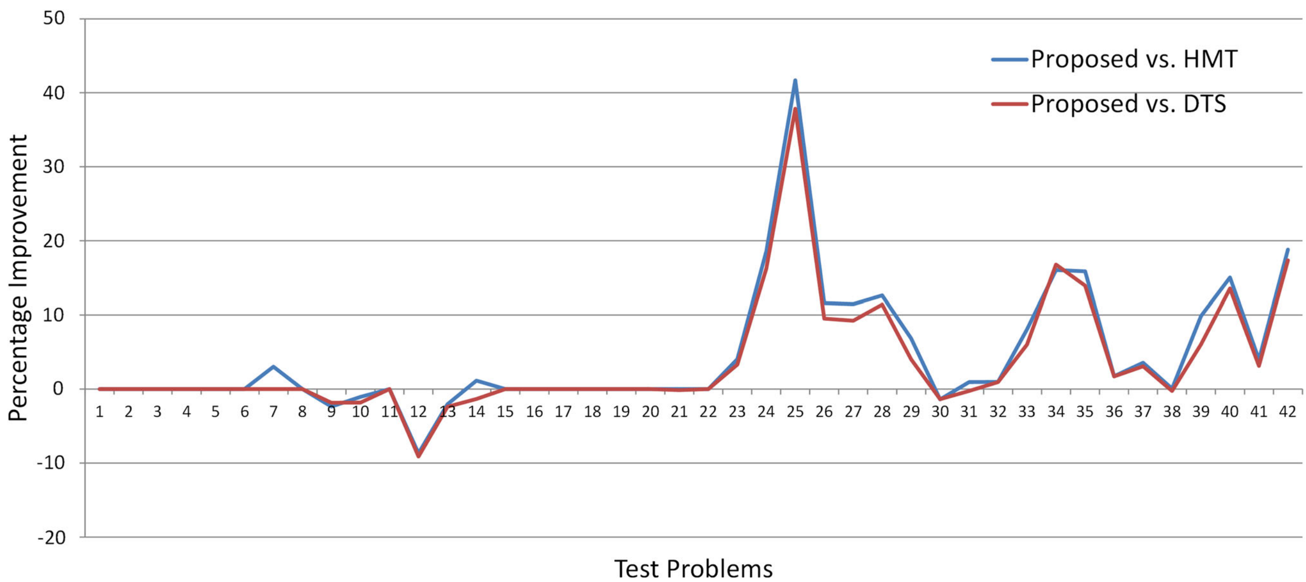


**Fig. 3** Illustration of the percentage improvement in solution quality in using the proposed heuristic as compared to the existing heuristics

**Table 3** Test problems in which the results of all trials of the proposed heuristic are better than the HMT and DTS heuristics

| Problem | $\|F\|$ | Proposed (worst) | HMT (best) [9] | DTS (best) [8] |
|---|---|---|---|---|
| GSM2-184 | 49 | 874* | 999 | – |
| GSM2-184 | 52 | 162* | 186 | – |
| GSM2-227 | 39 | 10,493* | 10,979 | 10,550 |
| 1-1-50-75-30-2-50 | 10 | 97* | 101 | – |
| 1-1-50-75-30-2-50 | 11 | 55* | 59 | 59 |
| 1-1-50-75-30-2-50 | 12 | 36* | 38 | |
| 1-5-50-75-30-2-100 | 10 | 179* | 186 | |
| 1-5-50-75-30-2-100 | 12 | 63* | 65 | |
| 1-6-50-75-30-0-10000 | 13 | 1309* | – | 1318 |

We also compared the proposed heuristic with a TS heuristic which employs a dynamic length tabu list (DL-TS heuristic) [13]. We downloaded the source code of the DL-TS heuristic from its author's website [13]. We executed DL-TS heuristic with default parameters on the same platform as the proposed heuristic. We conducted up to twenty trials on each problem. We applied the Wilcoxon signed tests to compare the results of the proposed and DL-TS heuristics. The DL-TS heuristic is a new TS heuristic for the FA problem, and the comparison results show the superiority of the proposed heuristic over an exclusive application of TS heuristic. The Wilcoxon test is among the methods recommended by experts to compare two heuristics [18]. Table 4 shows the results of the Wilcoxon signed tests. In Table 4, the first two columns specify the problem. Third and fourth columns specify the mean values of the objective function. The fifth column specifies the $p$ values of the Wilcoxon tests. The last column specifies conclusion about the comparison using the $p$ value. When the $p$ value is greater than or equal to $\alpha = 0.01$, then we consider the two heuristics are equal in performance. When the $p$ value is smaller than $\alpha = 0.01$, then the heuristic that has a smaller mean dominates the other heuristic. The results indicate that in eleven problems the results of the proposed and DL-TS heuristics are equal and in thirty problems the proposed heuristic performs better than the DL-TS heuristic. We cannot apply the Wilcoxon test when all trials produced the same result. In those problems, we can compare the results without any statistical tests.

## 5 Conclusion and Future Work

This article proposed a heuristic for the minimization of interference in the FS-FA problem. The heuristic has two main components: a local search heuristic, and a compound-move operation. The local search heuristic employs a LUT. Each entry in the LUT is a possible move and store the value by which it can increase or decrease the interference of the solu-

**Table 4** Results of Wilcoxon signed-rank tests

| Problem | $\|F\|$ | Mean values | | $p$ value | Remarks |
|---|---|---|---|---|---|
| | | Proposed | TS [13] | | |
| AC-45-17 | 7 | 32.00 | 32.00 | – | Equal |
| AC-45-17 | 9 | 15.00 | 15.15 | 0.22 | Equal |
| AC-45-25 | 11 | 33.00 | 33.00 | – | Equal |
| AC-95-9 | 6 | 31.00 | 31.00 | – | Equal |
| AC-95-17 | 15 | 33.26 | 34.00 | 0.0472 | Equal |
| AC-95-17 | 21 | 10.00 | 10.00 | – | Equal |
| GSM-93 | 9 | 32.00 | 34.15 | 0.0103 | Equal |
| GSM-93 | 13 | 7.00 | 8.25 | 7.57e−06 | Better |
| GSM-246 | 21 | 82.10 | 95.45 | 1.08e−05 | Better |
| GSM-246 | 31 | 26.58 | 35.35 | 1.03e−05 | Better |
| Test95 | 36 | 8.00 | 9.70 | 4.85e−06 | Better |
| Test282 | 61 | 58.05 | 73.50 | 1.09e−05 | Better |
| Test282 | 71 | 30.00 | 45.70 | 1.02e−05 | Better |
| Test282 | 81 | 12.06 | 23.50 | 9.9e−06 | Better |
| P06-5 | 11 | 133.00 | 150.75 | 7.98e−06 | Better |
| P06-3 | 31 | 115.00 | 137.75 | 7.83e−06 | Better |
| P06b-5 | 21 | 52.00 | 55.75 | 4.47e−05 | Better |
| P06b-5 | 31 | 25.00 | 27.80 | 4.15e−05 | Better |
| P06b-3 | 31 | 112.00 | 125.15 | 7.96e−06 | Better |
| P06b-3 | 71 | 26.00 | 32.00 | 7.49e−06 | Better |
| 1-4-50-75-30-2-1 | 6 | 71.00 | 70.65 | 0.0324 | Equal |
| 1-4-50-75-30-2-1 | 10 | 19.00 | 19.10 | 0.334 | Equal |
| GSM2-184 | 39 | 5414.57 | 6475.70 | 1.2e−05 | Better |
| GSM2-184 | 49 | 874.00 | 1052.95 | 4.2e−05 | Better |
| GSM2-184 | 52 | 162.00 | 262.25 | 0.000826 | Better |
| GSM2-227 | 29 | 60,186.63 | 218,430.05 | 6.66e−08 | Better |
| GSM2-227 | 39 | 9890.86 | 15,990.05 | 6.66e−08 | Better |
| GSM2-227 | 49 | 2315.06 | 3537.75 | 6.66e−08 | Better |
| GSM2-272 | 34 | 56,336.68 | 111,740.95 | 6.66e−08 | Better |
| GSM2-272 | 39 | 28,878.50 | 43,536.45 | 6.66e−08 | Better |
| GSM2-272 | 49 | 7967.95 | 13, 177.10 | 6.66e−08 | Better |
| 1-1-50-75-30-2-50 | 5 | 1242.00 | 1243.20 | 0.0255 | Equal |
| 1-1-50-75-30-2-50 | 10 | 96.80 | 104.80 | 4.67e−05 | Better |
| 1-1-50-75-30-2-50 | 11 | 55.00 | 67.05 | 7.79e−06 | Better |
| 1-1-50-75-30-2-50 | 12 | 33.31 | 40.75 | 1.07e−05 | Better |
| 1-2-50-75-30-4-50 | 9 | 669.00 | 685.95 | 0.000224 | Better |
| 1-2-50-75-30-4-50 | 11 | 315.00 | 336.45 | 1.03e−05 | Better |
| 1-3-50-75-30-0-50 | 7 | 197.00 | 198.05 | 0.0274 | Equal |
| 1-5-50-75-30-2-100 | 10 | 172.70 | 189.55 | 5.59e−05 | Better |
| 1-5-50-75-30-2-100 | 12 | 59.90 | 71.00 | 1.09e−05 | Better |
| 1-6-50-75-30-0-10000 | 10 | 6842.00 | 7159.65 | 1.12e−05 | Better |
| 1-6-50-75-30-0-10000 | 13 | 1128.15 | 1454.05 | 1.4e−05 | Better |

tion. For example, the value at location $(i, j)$ in the LUT stores the increase/decrease in the interference, if the transmitter $v_i \in V$ chooses the frequency $f_j \in F$. We can classify the moves in the LUT as positive and negative. In the heuristic, we first try to find a positive move. However, if we could not find a positive move, then we choose a negative move that can increase the interference by an amount which is less than the average increase in all negative moves. The local search is considered to be trapped in a local minimum if many successive iterations show no improvement in the cost. The compound-move shifts the search to a new location in the search space and prevents the search from getting trapped into any local minimum. We demonstrated using experiments that the proposed heuristic is competitive to existing single-solution-based heuristics such as HMT and DTS heuristics. The experiments used the same benchmarks as used by other recent heuristics. The solution quality of the proposed heuristic is better than or equal to that of HMT in 88% test problems, and better than or equal to that of DTS in 79% test problems. In some problems, the worst result of any trial of the proposed heuristic is better than the best reported result of the HMT and DTS heuristics. In the future, it can be further enhanced to solve the dynamic frequency assignment problem.

**Acknowledgements** Acknowledgments are due to King Fahd University of Petroleum & Minerals, Dhahran, Saudi Arabia for all support.

## References

1. Audhya, G.K.; Sinha, K.; Ghosh, S.C.; Sinha, B.P.: A survey on the channel assignment problem in wireless networks. Wirel. Commun. Mob. Comput. **11**(5), 583–609 (2011). https://doi.org/10.1002/wcm.898
2. Hale, W.K.: Frequency assignment: theory and applications. Proc. IEEE **68**(12), 1497–1514 (1980). https://doi.org/10.1109/PROC.1980.11899
3. Garey, M.; Johnson, D.: Computers and Intractability: A Guide to the Theory of NP-completeness. In: Freeman, W.H. (ed.) Books in Mathematical Series (1979). https://books.google.com.sa/books?id=fjxGAQAAIAAJ
4. Segura, C.; Hernández-Aguirre, A.; Luna, F.; Alba, E.: Improving diversity in evolutionary algorithms: New best solutions for frequency assignment. IEEE Trans. Evol. Comput. **21**(4), 539–553 (2017). https://doi.org/10.1109/TEVC.2016.2641477
5. Sait, S.M.; Youssef, H.: Iterative Computer Algorithms with Applications in Engineering. IEEE Computer Society Press, California (1999)
6. Beckmann, D.; Killat, U.: A new strategy for the application of genetic algorithms to the channel assignment problem. IEEE Trans. Veh. Technol. **48**(4), 1261–1269 (1999). https://doi.org/10.1109/25.775374
7. Ngo, C.Y.; Li, V.O.K.: Fixed channel assignment in cellular radio networks using a modified genetic algorithm. IEEE Trans. Veh. Technol. **47**(1), 163–172 (1998). https://doi.org/10.1109/25.661043
8. Montemanni, R.; Moon, J.N.J.; Smith, D.H.: An improved tabu search algorithm for the fixed-spectrum frequency-assignment problem. IEEE Trans. Veh. Technol. **52**(4), 891–901 (2003). https://doi.org/10.1109/TVT.2003.810976
9. Montemanni, R.; Smith, D.H.: Heuristic manipulation, tabu search and frequency assignment. Comput. Oper. Res. **37**(3), 543–551 (2010). https://doi.org/10.1016/j.cor.2008.08.006
10. Galinier, P.; Hao, J.K.: A general approach for constraint solving by local search. J. Math. Modell. Algorithms **3**(1), 73–88 (2004). https://doi.org/10.1023/B:JMMA.0000026709.24659.da
11. Liwei, L.; Rongshuang, F.: Simulated annealing algorithm in solving frequency assignment problem. In: 2010 3rd International Conference on Advanced Computer Theory and Engineering(ICACTE), vol 1, pp V1-361–V1-364 (2010). https://doi.org/10.1109/ICACTE.2010.5579000
12. Duque-Anton, M.; Kunz, D.; Ruber, B.: Channel assignment for cellular radio using simulated annealing. IEEE Trans. Veh. Technol. **42**(1), 14–21 (1993). https://doi.org/10.1109/25.192382
13. Lai, X.; Hao, J.K.: Path relinking for the fixed spectrum frequency assignment problem. Expert Syst. Appl. **42**(10), 4755–4767 (2015). https://doi.org/10.1016/j.eswa.2015.01.025
14. Dueck, G.; Scheuer, T.: Threshold accepting: a general purpose optimization algorithm appearing superior to simulated annealing. J. Comput. Phys. **90**(1), 161–175 (1990). https://doi.org/10.1016/0021-9991(90)90201-B
15. Correia, L.M.: Wireless flexible personalised communications : COST 259, European co-operation in mobile radio research. Wiley, Chichester (2001). http://lib.ugent.be/catalog/rug01:001649471
16. Marti, R.; Gortazar, F.; Duarte, A.: Heuristics for the bandwidth colouring problem. Int. J. Metaheuristics **1**(1), 11–29 (2010). https://doi.org/10.1504/IJMHEUR.2010.033121
17. Matic, D.; Kratica, J.; Filipovic, V.: Variable neighborhood search for solving bandwidth coloring problem. Comput. Sci. Inf. Syst. **14**, 309–327 (2017)
18. Derrac, J.; García, S.; Molina, D.; Herrera, F.: A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. Swarm Evol. Comput. **1**(1), 3–18 (2011). https://doi.org/10.1016/j.swevo.2011.02.002