IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received October 5, 2018, accepted November 15, 2018, date of publication November 21, 2018, date of current version December 19, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2882595

An Optimization Heuristic Based on Non-Dominated Sorting and Tabu Search for the Fixed Spectrum Frequency Assignment Problem

UMAIR F. SIDDIQI[1], (Member, IEEE), AND SADIQ M. SAIT[1,2], (Senior Member, IEEE)
[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, King Fahd University of Petroleum and Minerals, Dhahran 31261, Saudi Arabia

Corresponding author: Sadiq M. Sait (sadiq@kfupm.edu.sa)

This work was supported by the King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia.

ABSTRACT Frequency assignment for minimum interference is a fundamental problem in telecommunications networks. Combinatorial optimization heuristics are among the successful techniques employed to solve this problem. The heuristic proposed in this paper is a hybrid of the non-dominated sorting genetic algorithm-II and tabu search (TS) heuristics. Although several hybrid heuristics exist, in addition to the parameters of the metaheuristics, they also contain problem-specific parameters. The proposed heuristic does not have any problem-specific parameter. In each iteration, we apply TS to each element of the population to replace it with the best solution in its neighborhood. We select the elements for the population of the next generation using the principles of the non-dominated sorting. The non-dominated sorting considers two attributes of a solution: 1) interference (i.e., a measure of the violation of constraints) and 2) entropy (i.e., a measure of the diversity of the solutions in a population). Experimental results show that the proposed heuristic can produce solutions of quality better than four existing known heuristics. The gain of the proposed heuristic over a recently proposed hybrid heuristic is verified using the Wilcoxon statistical test. The convergence time of the proposed heuristic is also comparable with the existing heuristics.

INDEX TERMS Frequency assignment problem, genetic algorithm, tabu search, heuristics, non-dominated sorting, combinatorial optimization, NP-hard.

I. INTRODUCTION

Wireless communication systems have gained a lot of importance because of their widespread use. Examples of wireless communications include: mobile telephones, satellite communications, TV broadcasting, wireless LANs, and military communications operations. With some variations, the Frequency assignment (FA) problem appears in many different realms of wireless communications, such as in general system for mobile communications (GSM), and in satellite communication [1]. The models and solution techniques are different for each type of FA problem.

The FA problem in wireless networks bears a resembles to graph coloring problem [2]. The fixed spectrum FA problem of wireless networks occurs in telecommunication networks in which the service provider is licensed to use a spectrum of channels. However, the number of frequencies is usually lesser than the number of communication devices or connections in the network. Therefore, some frequencies should be re-used within the network. The assignment of frequencies to communication devices or connections is an offline process, and one cannot modify the FA during the operation of the system.

The FA problem has three components: (i) transmitters; (ii) a fixed spectrum of channels; and (iii) a set of constraints. The transmitters need a channel from the spectrum to be able to operate, and the constraints specify the separation necessary between frequencies of different transmitters. The transmitters can interfere with each other if their frequencies are not sufficiently separated or spaced. When the frequencies of any two transmitters do not satisfy the constraints, then they cause interference. This work solves the FA problem that aims to minimize the interference [3]. The FA problem is NP complete and polynomial time algorithms cannot find a reasonable solution even when the network size is small [4]–[6].

In the past, many different optimization heuristics have addressed the FA problem, but they have some issues such as ordinary solution quality or presence of problem-specific parameters, to name a few. Some examples of the proposed optimization heuristics are: Constraint Programming [7], Neural Networks [8], [9], Tabu search (TS) [10]–[12], Simulated annealing (SA) [12]–[15], Genetic algorithm (GA) [15]–[17], Threshold accepting (TA) algorithm [10], [18] and Memetic algorithm (MA) [6].

Some optimization heuristics work with a single solution, while others work with a population of solutions. Two classes of the optimization heuristics according to the number of their objectives are single-objective and multiobjective optimization heuristics [19]. The existing single-solution and single-objective heuristics for the FA problem use excellent local search metaheuristics (e.g., TS) and have robust exploitation capability [15]. The exploitation capability of a heuristic is its ability to search for a solution within a small region of the search space [20]. However, they are still unable to deliver extraordinary results because of their inadequate exploration capability. The exploration is synonymous to diversity and is the ability of the optimization heuristic to visit new regions in the search space [20]. These deficiencies led to the development of heuristics for the FA problem that emphasize on the diversity. The optimization heuristics that gives importance to diversity maintain a population of solutions and use sophisticated methods to keep diversity in their population to keep exploring new regions of the search space.

This work proposes a population-based optimization heuristic that uses the principles of Pareto-optimality and preservation of solutions for the next generation, based on their non-domination by other solutions. The attributes of a solution represent its unique features such as its interference, and diversity from other solutions. A solution dominates another solution if it is better than the other solution in at least one attribute and not worst in any attribute. Despite its simple design, the proposed heuristic is quite handy in solving the FA-problem. The proposed heuristic also employs Tabu Search (TS) heuristic to determine the solution of minimal interference in the neighborhood of each solution in the population. The Pareto-optimality and non-domination sorting depends on its two attributes of each solution: interference and entropy. The entropy of a solution is a quantitative measure of its similarity with other solutions in the population. A large value of entropy of a solution means that it is considerably similar to other solutions in the population. The principle of non-domination sorting was first employed in the non-dominated sorting genetic algorithm (NSGA) [21]. This work is the first application of the principles of non-domination sorting and Pareto-optimality to solve the single-objective FA-problem.

The existing optimization heuristics for the FA problem contain many parameters. Some are standard parameters of the metaheuristics, whereas, others are problem-specific. The adjustment of values of problem-specific parameters requires knowledge of the problem. The proposed optimization heuristic is free from all problem-specific parameters, and a practitioner who is skilled in the application of metaheuristics can employ it without much knowledge of the problem. A recent study revealed that non-dominated sorting [22] is critical in the convergence capability of the NSGA-II. In the absence of non-dominated sorting, NSGA-II cannot converge quickly to optimal solutions. In our work, non-dominated sorting has played a critical role in the development of a parameter-free, fast and efficient heuristic for solving the FA problem.

Experimental results show that the proposed heuristic outperforms results from well-known heuristics for the FA problem. In short, the proposed heuristic which is hybrid of NSGA and TS can yield good quality solutions and is also free from problem-specific parameters [6], [10], [11], [23].

The organization of this paper is as follows. Section II briefly describes some relevant previous work. Section III describes some basic concepts and definitions related to FA-problem. Aspects of the proposed heuristic are detailed in Section IV. Section V shows the experimental results and a discussion on them. The last section contains the conclusion and future work.

II. RELEVANT WORK

In the past, many different types of hybrids of optimization metaheuristics have been devised to solve the FA and bandwidth coloring problems [24]. The bandwidth coloring and frequency assignment problem are similar, and hence one can also model the frequency assignment as a bandwidth coloring problem. We can classify the existing hybrid heuristics for the FA and bandwidth coloring problem into the following: (i) Standalone local search metaheuristics; (ii) Constructive method with a local search metaheuristic; (iii) Iterative local search metaheuristic with a perturbation method; and (iv) Iterative global search metaheuristic with a local search metaheuristic.

A. STANDALONE LOCAL SEARCH METAHEURISTICS

Zhao *et al.* [25] solved the FA problem for the D2D cellular networks that provide direct communication among devices that lie within a proximity of each other. The size of the FA problem (i.e., number of transmitters, and number of constraints) of the D2D cellular network is small as compared to the FA problem of the cellular networks. The authors solved the FA problem using a greedy heuristic and TS metaheuristic. Both greedy and the TS heuristics produced similar results; however, the greedy heuristic has much lesser computational complexity. The initial works on the FA problem of the cellular networks solved small problems, and employed the standalone local search metaheuristics such as TS. However, the current problems are too complex and need more sophisticated techniques such as hybrid heuristics.

B. CONSTRUCTIVE METHOD WITH A LOCAL SEARCH METAHEURISTIC

Jin and Hao [24] proposed a hybrid heuristic that comprises a constructive method and a TS heuristic. In the constructive method, the first step is to order the transmitters and the second step is to assign them a valid frequency. A valid frequency means that it does not violate any constraint. When the assignment of frequencies in the constructive method reaches to a condition in which no valid frequency is available to a transmitter, then it immediately terminates. The next step is to apply the TS heuristic on the current partial solution. The TS heuristic minimizes the violation of constraints. If the TS meets its goal (i.e., either complete or partial removal of the violations), then the constructive method will continue assigning frequencies to the remaining transmitters. However, if the TS does not meet its goal, then the constructive method performs an undo operation on the current solution, re-orders the transmitters and restarts the assignment of frequencies to the transmitters. Jin and Hao [24] minimize the number of frequencies required to determine a violation free solution. In this article, we considered the fixed spectrum FA problem in which the spectrum (or the number of frequencies) is fixed, and we should minimize the interference due to the violation of constraints.

C. ITERATIVE LOCAL SEARCH METAHEURISTIC WITH A PERTURBATION METHOD

These types of heuristics solve the FA-problem by combining a local search metaheuristic such as SA or TS with an additional technique to avoid getting trapped in local minima. The local search metaheuristics have excellent exploitation (or intensification) ability, but they get trapped into local optima. The introduction of exploration in the local search metaheuristics can help them to avoid this. In the literature we find techniques suggested by Duque-Anton *et al.* [14] and Montemanni *et al.* [10] that have been proposed to avoid the above situation. Duque-Anton *et al.* [14] employed a technique they referred to as 'long jump' in their SA algorithm. In the long jump, they simultaneously change the frequencies of many transmitters [14]. Montemanni *et al.* [10] employed a cell optimization technique in their dynamic Tabu list based TS algorithm. The cell optimization technique tries to find an optimal frequency for each transmitter in the network and thereby shifts the solution to a completely new location in the search space. The main features of their dynamic TS algorithm are: (i) The move operation chooses a new solution which is different from the current solution by one channel and is also a valid solution; (ii) A threshold accepting criterion is used while accepting new moves; and (iii) The values of threshold and the size of the Tabu list decreases with iterations.

Another recent work for the same problem employs a compound move with a local search heuristic that uses probabilities to select a subset of transmitters and then assign to them frequencies which are randomly selected from the spectrum [26]. Some proposals improved the runtime of the heuristics by reducing/restricting the size of the search space. Montemanni and Smith [11] proposed a Heuristic Manipulation (HM) technique based TS algorithm that uses artificial constraints to reduce the search space. The method to create artificial constraints analyzes the best solutions discovered in the search and determines the pair of transmitters that should have different frequencies to produce good quality solutions.

D. ITERATIVE GLOBAL SEARCH METAHEURISTIC WITH A LOCAL SEARCH METAHEURISTIC

In this type, we solve the FA-problem by combining both exploration and exploitation (or intensification) within a single heuristic [2]. The heuristics of this type are generally hybrids of a population-based global search metaheuristic with a single-solution local search metaheuristic. The population-based metaheuristic maintains a population of solutions and applies a local search heuristic to each solution of its population. The efficiency of the population-based metaheuristic is measured as its ability to explore the regions that contain optimal solutions. The efficiency of the local search is a measure of its ability to find the best solution within a small region around its initial solution. Some examples of these types of heuristics are: (i) (1 + 1)-EA with hill-climbing (or greedy) local search [2], Ant Colony Optimization (ACO) with a greedy local search [27], Memetic Algorithm (MA) [6] with hill-climbing local search, and Path-re-linking algorithm with TS-based local search [23]. In the following paragraphs, we briefly mention two best heuristics that emphasis on both exploration and exploitation.

Lai and Hao [23] proposed a population-based heuristic for the FA-problem. They select the solutions for the population based on their diversity because a diverse population generally covers many unique locations of the search space. The diversity of a solution is its minimum Hamming distance from any other solution in the population. The heuristic has three principal features: (i) Path Relinking operators to create offsprings; (ii) Diversity and solution quality-based criteria to selects solutions for the population of the next generation; and (iii) A TS algorithm to perform the local search. The path relinking operator chooses two solutions that have good quality and a considerable Hamming distance between them and creates two offspring in such a way that in each child the channel assignment of up-to a certain number of cells is different from any one of the parents. The new frequencies for the cells can be selected based on a greedy or a random approach. TS algorithm is applied to the offspring to obtain the best solutions in their neighborhoods. The experimental results indicated that their heuristic performs much better than the earlier TS-based heuristics. This heuristic is referred to as Path-relinking algorithm because of its unique path relinking operators [23].

Segura *et al.* [6] proposed an MA based heuristic that advances the application of population-based heuristics to solve the FA-problem. The heuristic contains a diversity

TABLE 1. Overview of the parameters in the existing optimization heuristics for the CA problem.

Algorithm	General Parameters	Problem-Specific Parameters
HMT [11], [10]	# of parameters= 2 (i) Tabu list size; and (ii) Rate of change in the Tabu list size	# of parameters= 4 (i) Time between the update of artificial constraints; (ii) Maximum number of artificial constraints that can be active at any time; (iii) Number of artificial constraints substituted periodically; and (iv) Interval between the update of the cost table
Path relinking [23]	# of parameters= 2 (i) Population size; and (ii) Depth of the Tabu list	# of parameters= 4 (i) Dynamic update of the Tabu tenure; (ii) & (iii) Distance and list size in the relinking operator; and (iv) Number of times to re-build population
MA [6]	# of parameters= 3 (i) Population size; (ii) Crossover; and (iii) Mutation probabilities	# of parameters= 2 (i) Radius of the hypersphere that is used by any solution to determine the solutions of the population that lies within its proximity; and (ii) Number of iterations in the local search

based technique for the selection of elements for the population of the next generation [6]. The selection technique considers two attributes of each solution: interference and diversity. The diversity cost of a solution takes into account the elapsed time, maximum allowed time and initial diversity. The diversity cost gets penalized if it contributes little to the diversity of the population. The non-dominance of a solution is determined using its interference and diversity cost. The selection technique is an iterative process which in each iteration selects solution that are non-dominated among all the unselected solutions. Genetic operators are used to create offsprings, and each child goes through a hill-climbing based local search heuristic soon after its creation. The experimental results showed that it could produce good results in many test problems.

The existing heuristics such as the ones mentioned above employ many parameters, and most of them are problem-specific, i.e., adjustment of their values need knowledge of the problem. A unique feature of our proposed heuristic is that it has no problem-specific parameter. Table 1 lists the parameters of the existing heuristics with their type (general or problem-specific) and a short description of their purpose. The results of the heuristics proposed prior to the year 2010 cannot solve the current problems, and therefore are not included in Table 1.

III. PROBLEM FORMULATION

The FA-problem is modeled using a weighted and undirected quadruple graph $G(V, E, D, P)$. The set V contains vertices, and E contains edges. The vertices represent the transmitters to whom frequencies should be assigned and are represented by $\{v_0, v_1, \ldots, v_{m-1}\}$, where m is the total number of transmitters. The edges connect any two vertices (or transmitters) and are used to indicate that a constraint exists between those two transmitters. An edge is represented by (v_i, v_j), where v_i and $v_j \in V$. The set D contains the weights of the edges and a mapping $E \rightarrow D$ exists such that (v_i, v_j) is mapped to d_{ij}. d_{ij} is the amount of separation which is necessary between the frequencies assigned to v_i and v_j. The set P contains the penalties for the violation of constraints. There is a mapping $E \rightarrow P$, such that (v_i, v_j) is mapped to p_{ij}. p_{ij} is a penalty to be paid if the separation between the frequencies assigned to v_i and v_j is less than d_{ij}. The spectrum of frequencies available to the transmitters is represented by F and contains up-to n channels. The spectrum is represented as follows: $F = \{f_0, f_1, \ldots, f_{n-1}\}$.

The solution of the FA-problem is an assignment vector represented by $\Delta = \{\delta_0, \delta_1, \ldots, \delta_{m-1}\}$, where δ_k is the frequency assigned to transmitter v_k (where $k \in \{0, 1, .., m-1\}$.

The interference between transmitters v_i and v_j can occur $\exists(v_i, v_j) \in E$. The interference can be strong or weak according to the value of the penalty. The interference between any two transmitter v_i and v_j can be computed using the below-mentioned formula.

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

The interference of a solution is the sum of interferences of all edges present in it and can be computed using the following equation (2). The goal of the FA-problem is to minimize the interference of the solution as mentioned in (3).

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

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

A population-based optimization heuristic contains population is solutions that are represented by $\{\Delta_0, \Delta_1, \ldots, \Delta_{N-1}\}$. Entropy is characteristic of a solution in a population. It is a metric to express the diversity of a solution within its population. Rosca [28] introduced the use of entropy in evolutionary algorithms. Liu *et al.* [29] then proposed that diversity-driven evolutionary algorithms have a right balance of exploration and exploitation. The symbol $\xi(\Delta_i)$ represents the entropy of a solution (Δ_i) and the mathematical equations

Input: $G(V, E, D, P)$, F (Spectrum), N, α_H, β_T, $\gamma_T \in Z^+$
Output: POP $= \{\Delta_0, \Delta_1, ..., \Delta_{N-1}\}$

Initialization: Initialize POP with n random solutions.
while Stopping criterion is not reached **do**
  Application of local search: Apply TS algorithm to each solution in POP.
  Creation of offsprings: Apply GA operators to create offspring (CHD).
  Application of local search on the offsprings: Apply TS algorithm to each solution in CHD.
  Population update: Select solutions for the population of the next generation, i.e., select elements from $\{$POP $\cup$ CHD $\}$ for POP.
endwhile
return POP

FIGURE 1. Outline of the Proposed heuristic.

to determine its value are mentioned in the following.

$$\xi(\Delta_x) = -\sum_{j=0}^{m}(p_j \times log_2(p_j)) \tag{4}$$

$$p_j = \sum_{i,k\in\{0,1,...,N-1\}} M_{i,k}(p_j) \tag{5}$$

$$M_{i,k}(p_j) = \begin{cases} 1 & \text{if } \delta_{j,\Delta_i} = \delta_{j,\Delta_k} \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

The entropy is defined in (4) to (6) in top-to-bottom manner for a solution Δ_x. In (4), m represents the total number of transmitters and p_j represents the diversity of the j^{th} transmitter of solution Δ_x from the j^{th} transmitters of the remaining solutions in the population. The value of p_j can be computed using (5) and (6). Equation 6 computes a term $M_{i,k}(p_j)$, where i and k represent indices of any two solutions in the population. $M_{i,k}(p_j)$ returns 1, if the frequency of the j^{th} transmitter in solutions Δ_i and Δ_k are same, otherwise, it returns zero. Equation 5 computes the p_j value that can be used in (4).

IV. PROPOSED OPTIMIZATION HEURISTIC

We propose a population-based heuristic to solve the FA-problem. The heuristic gives importance to both exploitation and exploration. It contains TS-based local search to determine the best solution in the neighborhood of each element of the population. It also includes genetic crossover and mutation functions to explore new solutions in the search space. The remaining part of this section describes the proposed heuristic in detail.

Fig. 1 shows the main steps of the proposed heuristic. The input consists of the problem related information $G(V, E, D, P)$ and the spectrum of available frequencies (F). The remaining inputs are parameters of the GA operators and TS algorithm. The details of the remaining parameters are as follows: N is the population size; α_H is the mutation probability; β_T is the termination criterion of TS heuristic in terms of number of iterations; and γ_T is the depth of the Tabu list. The remaining part of this section describes the different steps of the proposed optimization heuristic.

A. INITIALIZATION

The initialization step creates a population of random solutions of size N. The intention of randomly generating a large population is to have diversity among solutions.

B. APPLICATION OF LOCAL SEARCH

The first step in the optimization loop applies a local search heuristic to all solutions in the population. This work uses TS heuristic which is among the most popular local search heuristics. It maintains a Tabu list that stores the moves taken in the previous γ_T iterations. In each iteration, a move that is not selected in the Tabu-list is selected. It also uses an aspiration criterion that enables selection of a good move (i.e., a move that improves the interference of the solution) even when it is present in the Tabu list. The stopping criterion of the TS algorithm is the occurrence of up-to β_T successive iterations without any improvement in the interference of the solution. The solutions obtained from the TS heuristic replace their values in POP.

C. CREATION OF OFFSPRINGS

The second step in the optimization loop is to create offspring solutions by applying GA operators. It uses the tournament selection technique for the selection of parents. In the tournament selection technique, the population is divided into two equal parts. The two parts are sorted w.r.t. their interference values. Then the topmost elements from each part are selected and go through the one-point crossover operation. The crossover operation creates two offsprings. Each offspring goes through the mutation operation that alters the frequency assignment of any cell with a probability of α_H. The set CHD stores the offsprings.

D. APPLICATION OF LOCAL SEARCH ON THE OFFSPRINGS

This step applies the TS algorithm to the offsprings. The solutions obtained from the TS algorithm replace their previous values in CHD.

Input: POP, CHD, $\{\Theta_0, \Theta_1, ..., \Theta_{N-1}\}$, $\{\xi_0, \xi_1, ..., \xi_{N-1}\}$
Output: POP of size N solutions
1: TOT= POP $\cup$ CHD
2: Clear POP and CHD
3: **Cluster the solutions in TOT:** This step determines the domination counts of all solutions and then cluster them based on their domination counts.
4: **Selection of solution from clusters for POP:** Sort the clusters and select all elements of a cluster into POP.
5: **Conditional selection of solution for POP based of crowding distance:** If the number of solutions in the last cluster which is chosen for insertion into POP has the number of solutions greater than the space available in POP, then a selection of a small number of solutions from that cluster is performed using the crowding distance method.
6: **return** POP

FIGURE 2. Method to select solutions for the population of the next iteration.

E. POPULATION UPDATE

This step selects solutions for the population (POP) of the next iteration from the sets POP and CHD. Each solution has two attributes: interference and entropy. The interference and entropy of a solution Δ_i are represented by $\Theta(\Delta_i)$ and $\xi(\Delta_i)$. The solutions that have minimum values of interference and entropy is kept in the population for the next iteration. NSGA-II algorithm proposes principles for selecting elements for a population in a multiobjective optimization problem, based on non-dominated sorting of the solutions. In this work, the two attributes of a solution act are two objectives and the principles of non-domination count as mentioned below.

The solutions are clustered based on their domination count. A solution Δ_i dominates another solution Δ_j, if it meets the conditions mentioned in (7) and (8).

$$\Theta(\Delta_i) < \Theta(\Delta_j) \text{ and } \xi(\Delta_i) \leq \xi(\Delta_j) \tag{7}$$

$$\Theta(\Delta_i) \leq \Theta(\Delta_j) \text{ and } \xi(\Delta_i) < \xi(\Delta_j) \tag{8}$$

The domination count of a solution is equal to the number of others solutions in the population that dominates it. The solutions are clustered based on their domination counts. The solutions that have the smallest domination count will be placed in the first cluster. The solutions that have the next smallest domination count will be placed in the next cluster. The domination count based clustering continues until the completion of the assignment of clusters to all solutions. The clusters are represented as $C_0, C_1, \ldots C_{d-1}$, where d is the total number of clusters. This domination count based clustering and sorting of clusters use the principles of the non-dominated sorting of NSGA-II algorithm.

The solutions in TOT are sorted in the ascending order of their cluster numbers. The solutions of the first cluster are first inserted into POP. If the POP still has room for more solutions (i.e., the number of solutions in it is less than N), then the solutions of the second cluster are inserted into it. This process continues until the POP is filled with N solutions. At any time, if the number of solutions in a cluster is more than the space available in POP. Then the crowding distance of each solution in that cluster is computed, and the top solutions with-respect-to the crowding distance will be inserted into POP. The method to compute the crowding distance of a solution is mentioned in the following.

Fig. 3 shows the algorithm for the computation of the crowding distance of a solution Δ_i that belongs to the cluster C_k. The cluster has a total of v solutions. The input consists of the solutions of the cluster and their interference and entropy values. The solutions are sorted w.r.t. to their interference and entropy values. C_k^i denotes the set that contains solutions sorted by interference, and C_k^e denotes the set that includes solutions sorted by entropy. The set R_k stores the crowding distance of the solutions present in C_k. The code in lines 3-4 assigns ∞ value to the crowding distance of the first and last elements of C_k^i and C_k^e. The code in line 5-8 determine the maximum and minimum values of interference and entropy among all the solutions of the cluster. The code in lines 9-16 computes the crowding distance of the remaining elements of C_k^i and C_k^e (i.e., excluding their first and last solutions). In line 10, y indicates the index in C_k of a solution that lies at the j^{th} index in C_k^i. Note that C_k^i is a sorted version of C_k. Therefore, the same solution could lie at different indices. The variables y^+ and y^i contain the indices in C_k of the solutions $C_k^i[j+1]$ and $C_k^i[j-1]$. The code lines 13-16 repeats the same procedure as employed in lines 9-12, but uses the sorted set C_k^e instead of C_k^i.

F. COMPUTATIONAL COMPLEXITY

In this section, we present the computational complexity of a generation of the proposed heuristic. As can be seen in Fig. 1, a generation comprises four steps: (i) Application of local search, (ii) Creation of offsprings, (iii) Application of local search on offsprings, and (iv) Population update.

Input: $C_k = \{\Delta_0, \Delta_1, ..., \Delta_{v-1}\}$ $\{\Theta_0, \Theta_1, ..., \Theta_{v-1}\}$, $\{\xi_0, \xi_1, ..., \xi_{v-1}\}$
Output: R_k: Crowding distances of the solutions in C_k.

Sort the solutions in the cluster C_K w.r.t. interference and represent the sorted set as C_k^i
Sort the solutions in the cluster C_K w.r.t. entropy and represent the sorted set as C_k^e
y_0= index of $C_k^i[0]$ in C_k, y_1= index of $C_k^i[v-1]$ in C_k, y_2= index of $C_k^e[0]$ in C_k, y_3= index of $C_k^e[v-1]$ in C_k
set $R_k[y_0]= +\infty$, $R_k[y_1]= +\infty$, $R_k[y_2]= +\infty$, $R_k[y_3]= +\infty$
M_Θ^x =max$(\Theta(\Delta_0), \Theta(\Delta_1), ..., \Theta(\Delta_{v-1}))$
M_ξ^x =max$(\xi(\Delta_0), \xi(\Delta_1), ..., \xi(\Delta_{v-1}))$
M_Θ^n =min$(\Theta(\Delta_0), \Theta(\Delta_1), ..., \Theta(\Delta_{v-1}))$
M_ξ^n =min$(\xi(\Delta_0), \xi(\Delta_1), ..., \xi(\Delta_{v-1}))$
for j= 1 to $v-2$ **do**
  y=index of $C_k^i[j]$ in C_k, y^+=index of $C_k^i[j+1]$ in C_k, y^-=index of $C_k^i[j-1]$ in C_k
  $R_k[y]= \frac{\Theta(\Delta_{y^+})-\Theta(\Delta_{Y^-})}{M_\Theta^x - M_\Theta^n}$
endfor
for j= 1 to $v-2$ **do**
  y=index of $C_k^e[j]$ in C_k, y^+=index of $C_k^e[j+1]$ in C_k, y^-=index of $C_k^e[j-1]$ in C_k
  $R_k[y]= R_k[y] + \frac{\xi(\Delta_{y^+})-\xi(\Delta_{Y^-})}{M_\xi^x - M_\xi^n}$
endfor
return R_k

FIGURE 3. Method to compute the crowding distance of the solutions of a cluster (C_k).

The computational complexities of each of the four steps is discussed below.

1) APPLICATION OF THE LOCAL SEARCH

In this step, the TS heuristic is applied to each solution of the population. The computational complexity of the TS heuristic is as follows. The first iteration has a complexity of $O(m^2n)$ (where, m is the number of transmitters and n is the number of frequencies in the spectrum) [10]. All the remaining iterations have a complexity of $O(mn)$ [10]. The total number of iterations executed by the TS heuristic depends on the termination criterion. In this work, we used the criterion that the TS heuristic should terminate if β_T number of consecutive iterations brings no improvement in the objective function. Therefore, the maximum number of iterations of the TS heuristic is a multiple of β_T and represented as $K_c\beta_T$, where K_c is a positive real number. The total complexity of this step in equal to $O((m + K_c\beta_T)mnN)$ (where, N is the population size and K_c and β_T are considered equal to 1000 and 2000, respectively.

2) CREATION OF OFFSPRINGS

In this step, we first sort the solutions in the population w.r.t. their objective function values. The complexity of the sorting is $O(N\log N)$. This step applied crossover and mutation steps to create up-yo N offsprings. The complexity of the single-point crossover and mutation operation is $O(m)$. The complexity of this step is $O(N(\log N + m))$ or $O(N\log N)$.

3) APPLICATION OF LOCAL SEARCH ON THE OFFSPRINGS

The complexity of this step is same as that of the first step.

4) POPULATION UPDATE

The update population step of the NSGA-II algorithm consists of non-dominated sorting, crowding distance computation and sorting based on crowding distance operations. The complexity of the non-dominated sorting is equal to $O(N^2)$ [21], and the complexity of crowding distance computation and sorting is $O(N\log N)$ [21].

V. EXPERIMENTAL RESULTS AND DISCUSSION

The proposed optimization heuristic was implemented using C++ and R using R-Studio version 3.3.2 and Rcpp package. The implementation of the proposed algorithm comprises several C++ functions. The R functions perform the I/O related tasks. The execution platform consists of a computer that has an Intel Xeon 2.8 GHz processor and 64 GB of memory. The performance of the proposed algorithm has been evaluated on the same benchmarks as used by most of the existing algorithms for the FA-Problem such as HM-based TS algorithm (or HMT) [11], TS with dynamic length list algorithm [10], Path relinking algorithm [23], and MA for the FA-problem [6]. Table 2 lists the problems and their main characteristics. The first column mentions the problem name. Second and third columns contain the number of vertices (or transmitters) and the number of edges (or number of constraints). The fourth column ($\hat{d}$) contains the average

TABLE 2. Characteristics of the test problems [10], [11].

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	10430	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	8.95×10^6
GSM2-227	227	10088	0.18	9.10×10^6
GSM2-272	272	14525	0.16	7.95×10^6
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

separation between transmitters in the constraints, and the fifth column ($\hat{p}$) contains the average penalty values of the constraints. In the problems of Table 2 that have $\hat{p} > 1$, the penalty values of the constraints are not uniform.

The values of the parameters of the proposed optimization heuristic are set as follows: The termination criterion of the TS algorithm (β_H) = 2000, and size of the Tabu list (γ_T) = 50. The population size (N) and mutation probability (α_H) are set equal to 120 and 0.30, respectively. We experimented with different population sizes, smaller values usually causes premature convergences, and very large values slow down the optimization. The selected value provide a balance in most of the problems. In the experiments, we noticed that for the mutation probability a value which is slightly bigger than its usual value help in the quick exploration of the search space.

The effect of the nondeterministic behavior of the proposed optimization heuristic is accounted by executing up-to 25 trials on each problem. Each trial executes for up to two hours on problems GSM2-184, GSM2-227, and GSM2-272, up to 1 hour on problems GSM- 246, and Test282, and up to ten minutes on all remaining problems. The FA-problem is an offline problem, and the runtime is not critical as long as it is within practical limits. We performed two types of comparisons. The first type uses the published results of the existing algorithms and the second type uses the results obtained through executing an existing algorithm using its application program. In the following text, we first discuss the first type of comparison and then discuss the second type.

The proposed optimization heuristic is compared with four existing optimization heuristics that includes two hybrid heuristics of the third type (i.e., Iterative local search

TABLE 3. Comparison with existing single-solution based optimization heuristics.

Problem	$\|F\|$	Proposed		Dynamic-TS [10], [11]		HMT [11]	
		$mean$	(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	33·00	(33, 33)	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	79·20	(78, 80)	80·60	(79, 81)	80·20	(79, 82)
GSM-246	31	24·80	(24, 25)	26·30	(25, 27)	26·10	(25, 27)
Test95	36	8·00	(8, 8)	8·00	(8, 8)	8·00	(8, 8)
Test282	61	53·75	(52, 55)	53·40	(51, 56)	53·20	(51, 55)
Test282	71	27·39	(27, 29)	29·40	(27, 30)	29·30	(27, 30)
Test282	81	8·32	(8, 9)	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	70·00	(70, 70)	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	5265·30	(5250, 5322)	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	56789·96	(55513, 58138)	68077·70	(61586, 70105)	66510·00	(61586, 70105)
GSM2-227	39	8700·30	(8520, 8856)	11170·30	(10979, 11276)	10897·70	(10550, 11164)
GSM2-227	49	1998·00	(1998, 1998)	2649·10	(2459, 2828)	2613·10	(2459, 2828)
GSM2-272	34	52354·46	(51493, 53788)	60473·20	(57715, 67025)	58691·40	(56128, 64353)
GSM2-272	39	26685·60	(25932, 27258)	28484·30	(27416, 29323)	28488·20	(27416, 29307)
GSM2-272	49	7129·40	(7056, 7225)	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	96·00	(96, 96)	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	32·20	(32, 33)	39·60	(38, 42)	38·70	(36, 42)
1-2-50-75-30-4-50	9	665·00	(665, 665)	680·90	(671, 691)	680·60	(671, 691)
1-2-50-75-30-4-50	11	313·00	(313, 313)	326·60	(323, 337)	325·00	(317, 335)
1-3-50-75-30-0-50	7	194·00	(194, 194)	197·10	(196, 198)	196·50	(194, 199)
1-5-50-75-30-2-100	10	168·00	(168, 168)	191·60	(186, 197)	183·80	(176, 199)
1-5-50-75-30-2-100	12	56·51	(53, 57)	70·50	(65, 74)	69·30	(63, 74)
1-6-50-75-30-0-10000	10	6777·00	(6777, 6777)	7123·40	(6942, 7279)	7064·30	(6840, 7267)
1-6-50-75-30-0-10000	13	1190·00	(1190, 1190)	1389·60	(1207, 1490)	1365·20	(1318, 1440)

TABLE 4. Comparison with existing population-based optimization heuristics.

Problem	$\lvert F\rvert$	Proposed		Path-relinking [23]		MA [6]	
		$mean$	(best, worst)	$mean$	(best, worst)	$mean$	best
AC-45-17	7	32·00	(32,32)	32·00	(32, 32)	32·00	32
AC-45-17	9	15·00	(15,15)	15·00	(15, 15)	15·60	15
AC-45-25	11	33·00	(33,33)	33·00	(33, 33)	33·00	33
AC-95-9	6	31·00	(31, 31)	31·00	(31, 31)	31·00	31
AC-95-17	15	33·00	(33, 33)	33·00	(33, 33)	34·90	34
AC-95-17	21	10·00	(10, 10)	10·00	(10, 10)	10·00	10
GSM-93	9	32·00	(32, 32)	32·00	(32, 33)	34·70	32
GSM-93	13	7·00	(7, 7)	7·00	(7, 7)	8·20	7
GSM-246	21	79·20	(78, 80)	79·00	(78, 80)	84·70	81
GSM-246	31	24·80	(24, 25)	25·10	(24, 26)	28·60	27
Test95	36	8·00	(8, 8)	8·00	(8, 8)	8·00	8
Test282	61	53·75	(52, 55)	57·10	(56, 58)	62·30	50
Test282	71	27·39	(27, 29)	30·60	(29, 32)	33·50	31
Test282	81	8·32	(8, 9)	11·50	(10, 13)	12·80	11
P06-5	11	133·00	(133, 133)	133·00	(133, 133)	134·40	133
P06-3	31	115·00	(115, 115)	115·00	(115, 115)	118·30	115
P06b-5	21	52·00	(52, 52)	52·00	(52, 52)	52·00	52
P06b-5	31	25·00	(25, 25)	25·00	(25, 25)	25·00	25
P06b-3	31	112·00	(112, 112)	112·00	(112, 112)	112·40	112
P06b-3	71	26·00	(26, 26)	26·00	(26, 26)	26·00	26
1-4-50-75-30-2-1	6	70·00	(70, 70)	70·00	(70, 70)	71·70	70
1-4-50-75-30-2-1	10	19·00	(19, 19)	19·00	(19, 19)	20·00	19
GSM2-184	39	5265·30	(5250, 5322)	5276·90	(5250, 5322)	5252·10	5250
GSM2-184	49	874·00	(874, 874)	874·00	(874, 874)	874·00	874
GSM2-184	52	162·00	(162, 162)	162·00	(162, 162)	162·00	162
GSM2-227	29	56789·96	(55513, 58138)	59907·70	(58834, 61269)	58370·90	56397
GSM2-227	39	8700·30	(8520, 8856)	9329·70	(8760, 9755)	8948·20	8568
GSM2-227	49	1998·00	(1998, 1998)	2009·40	(1998, 2045)	1998·00	1998
GSM2-272	34	52354·46	(51493, 53788)	56916·30	(54085, 58524)	54390·80	52152
GSM2-272	39	26685·60	(25932, 27258)	28880·40	(28074, 29491)	27758·10	25852
GSM2-272	49	7129·40	(7056, 7225)	7252·50	(7107, 7361)	7211·50	7036
1-1-50-75-30-2-50	5	1242·00	(1242, 1242)	1242·00	(1242, 1242)	1242·40	1242
1-1-50-75-30-2-50	10	96·00	(96, 96)	96·00	(96, 96)	100·60	96
1-1-50-75-30-2-50	11	55·00	(55, 55)	55·00	(55, 55)	60·50	55
1-1-50-75-30-2-50	12	32·80	(32, 34)	32·00	(32, 32)	34·40	32
1-2-50-75-30-4-50	9	665·00	(665, 665)	665·00	(665, 665)	665·00	665
1-2-50-75-30-4-50	11	313·00	(313, 313)	313·00	(313, 313)	316·20	313
1-3-50-75-30-0-50	7	194·00	(194, 194)	194·00	(194, 194)	194·70	194
1-5-50-75-30-2-100	10	168·00	(168, 168)	168·00	(168, 168)	178·40	168
1-5-50-75-30-2-100	12	56·51	(53, 57)	57·00	(57, 57)	59·50	53
1-6-50-75-30-0-10000	10	6777·00	(6777, 6777)	6777·00	(6777, 6777)	6777·00	6777
1-6-50-75-30-0-10000	13	1190·00	(1190, 1190)	1190·00	(1190, 1190)	1190·00	1190

metaheuristic with a perturbation method) and two hybrid heuristics of the fourth type (i.e., Iterative global search metaheuristic with a local search metaheuristic). We obtained the results of the existing heuristics from their published articles. The comparison includes the following heuristics: Dynamic TS [10], HMT-based TS [10], [11], Path relinking algorithm [23] and MA algorithm whose survivor selection method is based on the Generational evolutionary algorithm (EA) with elitism [6]. The Dynamic TS, HMT based TS and Path relinking conducted ten trials, and MA algorithm conducted thirty trials on each problem.

Tables 3 and 4 show the results of the proposed optimization heuristic and compare them with the existing heuristics for the FA-problem. The results of each heuristic consist of two column. The first column mentions the mean value of the trials and the second column indicates the best and worst result obtained in any trial. For MA [6], the worst values are not publicly available. A visual inspection of Table 3 shows that the results of the proposed heuristic are better than the other heuristics in most of the problems. We have also summarized the results in Table 5 and Fig. 4.

Table 5 shows the numbers the test problems in which the average, best and worst value of the trials of the proposed heuristic is better than, equal to or worse than the existing heuristics. We have summarized the results into three groups.

TABLE 5. Summary of the comparison of the solution quality of the proposed heuristic with existing heuristics.

Type		DTS [10]	HMT [11]	Path-relinking [23]	MA [6]
Average Results	Better	26	26	12	26
	Equal	15	15	28	15
	Worst	1	1	2	1
Best Results	Better	22	19	9	9
	Equal	19	22	33	31
	Worst	1	1	0	2
Worst Results	Better	27	26	10	NA
	Equal	15	16	31	NA
	Worst	0	0	1	NA

The first group (Average Results) compares the average solution quality of all trials of the proposed heuristic with the average results of the proposed heuristics. The second group (Best Results) compares the best value of any trial of the proposed heuristic with the best values of the existing heuristic. In the same way, the group (Worst Results) compares the worst value of any trial of the proposed heuristic with the existing heuristics. In Table 5, the first row shows that the average solution quality of the proposed heuristic is better than that of DTS and HMT in twenty-six problems, better than Path relinking in twelve problems, and better than MA in twenty-six problems. The second row conveys the information that the average solution quality of the proposed heuristic is equal to DTS, HMT, Path relinking and MA in fifteen, fifteen, twenty-eight, and sixteen problems, respectively.

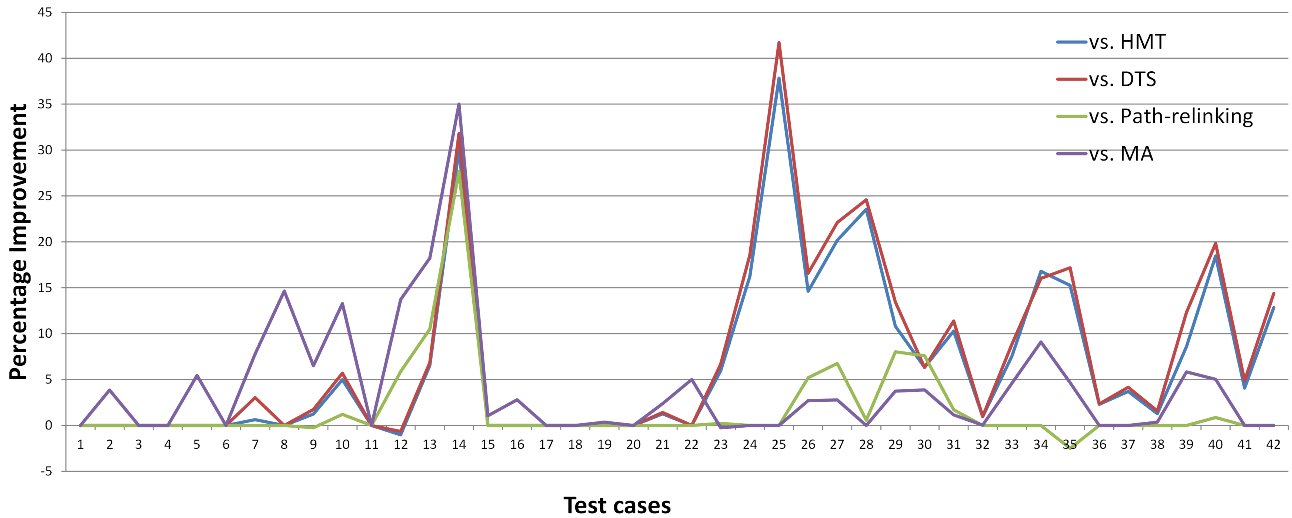


FIGURE 4. Illustration of the percentage improvement in solution quality in using the proposed heuristic as compared to the existing optimization heuristics [6], [10], [11], [23].

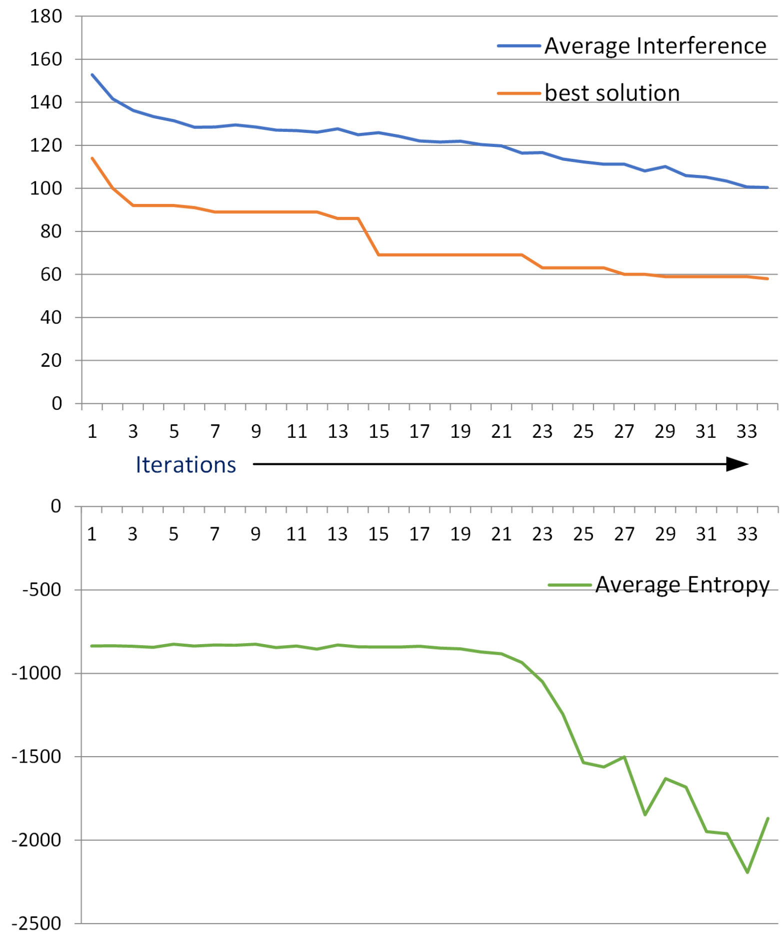


FIGURE 5. Illustration of the average interference, interference of the best solution, and average entropy of the population.

The third row presents the information when the average solution quality of the proposed heuristic is worst (i.e., inferior) than the existing heuristics.

The solution quality of the proposed heuristic is significantly better than HMT and Dynamic TS in many problems. The results in Table 4 shows that the results of the proposed

TABLE 6. Comparison between the proposed and Path relinking [23] heuristics using the Wilcoxon two-sided tests.

Problem	|F|	Solution Quality		p-value	Remarks
GSM-246	21	79.20 (±0.76)	80.00 (±0.65)	0.02	Both are equal
GSM-246	31	24.80 (±0.41)	25.56 (±0.58)	0.00001	Proposed is better
Test282	61	53.75 (±0.75)	55.4 (± 1.549)	0.002	Proposed is better
Test282	71	27.39 (±0.70)	29.10 (±1.19)	0.0001	Proposed is better
Test282	81	8.32 (±0.52)	9.7 (±1.06)	0.003	Proposed is better
GSM2-184	39	5256.30 (±4.13)	5260 (±17.01)	0.07	Both are equal
GSM2-227	29	56789.96 (±667.4)	59157 (±1343)	0.000000000002	Proposed is better
GSM2-227	39	8700.30 (±101.6)	9136 (±267.60)	0.00001	Proposed is better
GSM2-227	49	1998.00 (±0)	1998.00 (±0)	-	Both are equal
GSM2-272	34	52354.46 (±820.50)	55302.00 (±916)	0.00000000000006	Proposed is better
GSM2-272	39	26685.60 (±450.70)	28122.00 (±579.60)	0.0000002	Proposed is better
GSM2-272	49	7129.40 (±210.6)	7235 (±71.96)	0.9	Both are equal
1-1-50-75-30-2-50	12	32.20 (±0.42)	34.64 (±1.287)	0.00007	Proposed is better
1-5-50-75-30-2-100	12	56.51 (±1.271)	56.6 (±1.231)	0.6	Both are equal

TABLE 7. Solution quality and convergence rates of the proposed and path relinking heuristics.

Problem	|F|	DMO (%)		Convergence time (s)	
		Proposed	Path relinking	Proposed	Path relinking
AC-45-17	7	0	0	1.13 (± 0.1426)	3.42 (± 0.06)
AC-45-17	9	0	0	2.08 (± 0.96)	3.83 (± 0.58)
AC-45-25	11	0	0	1.20 (± 0.17)	5.31 (± 0.03)
AC-95-9	6	0	0	1.95 (± 0.55)	4.24 (± 0.07)
AC-95-17	15	0	3.27	401.90 (± 435.70)	31.72 (± 17.4)
AC-95-17	21	0	0	3.29 (± 1.20)	8.52 (± 0.065)
GSM-93	9	0	0	8.57 (± 6.63)	21.19 (± 17.54)
GSM-93	13	0	0	97.35 (± 92.92)	86.88 (± 42.99)
GSM-246	21	1.54	1.28	1314.00 (± 1589)	2007 (± 1604)
GSM-246	31	3.33	4.58	1253.00 (± 865.2)	1440 (± 1450)
Test95	36	0	0	7.19 (± 4.84)	23.15 (± 17.39)
Test282	61	5.39	11.96	3546.00 (± 1635.00)	2478 (± 441.10)
Test282	71	1.44	13.33	3448.00 (± 1760.00)	2946 (± 349.30)
Test282	81	4	21.25	3905.00 (± 1434)	4128 (± 1691)
P06-5	11	0	0	5.79 (± 10.51)	6.63 (± 0.17)
P06-3	31	0	0	109.6 (± 147.4)	62.49 (± 16.49)
P06b-5	21	0	0	0.83 (± 0.06)	9.64 (± 0.08)
P06b-5	31	0	0	0.87 (± 0.04)	12.36 (± 0.11)
P06b-3	31	0	0	26.09 (± 17.23)	83.61 (± 11.65)
P06b-3	71	0	0	10.56 (± 5.22)	75.25 (± 16.63)
1-4-50-75-30-2-1	6	0	0	2.728 (± 1.095)	5.84 (± 1.96)
1-4-50-75-30-2-1	10	0	0	4.362 (± 2.50)	4.73 (± 0.44)
GSM2-184	39	0.29	0.11	2303 (± 2159)	4770 (± 2958)
GSM2-184	49	0	0	4.027 (± 1.845)	29.57 (± 6.015)
GSM2-184	52	0	0	1.90 (± 0.04153)	17.6 (± 0.41)
GSM2-227	29	2.30	6.19	5364 (± 1478)	5964.00 (± 949.70)
GSM2-227	39	2.12	7.23	5374 (± 1179)	4235.00 (± 1944.00)
GSM2-227	49	0	0	1937 (±1247)	1577.00 (± 799.20)
GSM2-272	34	1.67	7.39	6310 (±1066)	5574.00 (± 1750.00)
GSM2-272	39	3.22	8.78	5987 (±1057)	4902.00 (± 1192.00)
GSM2-272	49	1.32	2.82	3615 (±2334)	4613.00 (± 1849.00)
1-1-50-75-30-2-50	5	0	0	3.026 (±2.33)	4.67 (± 0.23)
1-1-50-75-30-2-50	10	0	0	95.46 (±87.98)	61.94 (± 33.15)
1-1-50-75-30-2-50	11	0	0	210 (± 246.9)	76.22 (± 25.75)
1-1-50-75-30-2-50	12	2.5	8.25	277.2 (± 303.7)	22.75 (± 5.38)
1-2-50-75-30-4-50	9	0	0	16.57 (± 7.25)	31.11 (± 17.24)
1-2-50-75-30-4-50	11	0	0	102.80 (± 76.24)	174.00 (± 92.84)
1-3-50-75-30-0-50	7	0	0	37.17 (±30.18)	160.90 (± 71.05)
1-5-50-75-30-2-100	10	0	0	131.5 (± 127.4)	216.30 (± 137.90)
1-5-50-75-30-2-100	12	6.62	7.54	280.30 (±148.90)	96.66 (± 70.47)
1-6-50-75-30-0-10000	10	0	0	207.30 (±119.10)	77.69 (± 41.52)
1-6-50-75-30-0-10000	13	0	0	156.90 (± 78.8)	374.70 (± 91.59)

heuristic are also better than the Path relinking and MA heuristics in many problems. In Fig. 4, we show the percentage improvement in the solution quality of the proposed heuristic over the existing ones.

Fig. 4 indicates the percentage improvement of the mean value of the results of the proposed heuristic with the existing heuristics. The curve shows that the maximum percentage improvement is up-to 37.8% over HMT, 41.7% over DTS, 27.7% over Path re-linking, and 35% over MA. In the remaining part of this section, we present the results of the second part of the experiments in which we executed Path relinking heuristic in the same conditions as the proposed heuristic. Statistical tests are often used to compare two non-deterministic algorithms. This section presents a comparison of the proposed heuristic with Path relinking heuristic using statistical tests. We downloaded the source code of the Path relinking heuristic from its authors' website [23] and executed it on the same platform and with runtime conditions as that of the proposed heuristic.

The results in Table 4 show that the average solution quality of the proposed and path relinking heuristics are not equal to each other in up-to eighteen problems. Here, we compare the results of both heuristics on those eighteen problems using the unpaired two-samples two-sided Wilcoxon test [26], [30], [31] with the significance level of 0.05. The results of the statistical analysis indicate if the difference in the solution qualities is significant. Table 6 shows the results of the Wilcoxon tests. The column labeled as 'Solution Quality' contains the data in the following format: x ($\pm$ y), where x is the mean of all trials and y is the standard deviation of the trials. The results confirm that the solution quality of the proposed heuristic is better than that of the path relinking heuristic in up to nine problems.

The proposed optimization heuristic minimizes the entropy and interference of the solution as iteration proceeds. Fig. 5 shows a plot of the problem "1-5-50-75-30-2-100" with a spectrum of twelve channels. The plot shows the average interference of the solution that the initial population has maximum entropy and the entropy reduces with iterations. The plot finished at its best solution when the interference of the best solution is the minimum, the average entropy of the population is also among its smallest values at that point.

The convergence rate of any heuristic is an indication of how quickly it can converge to the best-known solutions [32]. Convergence Rate has been defined differently in different articles [32], [33]. We used the definition of convergence rate as employed by Bi *et al.* [32]. The method to determine the convergence rate requires that the program implementing the heuristics be run for a certain number of trials (η_c). In each trial the best cost of the population in each generation is stored. Convergence rate is expressed using a term known as DMO, which refers to the percentage deviation of the mean cost from the best-known solution. The DMO for the i^{th} generation can be computed as follows: If the best costs in the i^{th} generation of all trials are equal to $b_{(i,0)}, b_{(i,1)}, \ldots, b_{(i,\eta_c)}$; and the cost of the best-known solution is B^*, then the DMO value for the i^{th} generation is given by:

$$\mathrm{DMO}(i) = \frac{\frac{b_{(i,0)}+\ldots+b_{(i,\eta_c)}}{\eta_c} - B^*}{B^*} \times 100.$$

Table 7 presents the DMO values and the convergence time of the proposed and path relinking heurtsics. The results indicate that in up to twelve problems, the DMO of the proposed heuristic is better than that of Path relinking heuristic. Furthermore, the average convergence time of the proposed heuristic remains comparable to the path relinking heuristic.

VI. CONCLUSION AND FUTURE WORK

This article proposed an iterative optimization heuristic for the minimization of interference in the fixed spectrum FA-problem. It is population-based and uses conventional genetic operators (crossover and mutation) to create offsprings. It also employs a TS-based local search algorithm. Each iteration consists of the following steps. The first step is to replace each solution in the population from the best solution in its neighborhood using the TS-algorithm. The second step is to create offspring using the conventional crossover and mutation operations. The third step is to apply the TS-based local search to the offsprings and replace their values with the best solutions present in their neighborhood. The last step is to select solutions for the next iteration. The selection procedure uses the principles of the non-domination sorting of the NSGA-II algorithm. It considers two attributes of a solution, which are interference and entropy. A domination count of a solution indicates the number of solutions in the population that are better than it. The selection procedure prefers the solutions of the smaller domination count. In case of a tie in the domination count, it prefers the solutions whose interference and entropy values are different from other solutions that have the same domination count. Experimental results revealed that can produce solutions of good quality. We also analyzed the convergence rate of the proposed heuristic using experimental runs. The proposed heuristic can converge to solutions of quality which is equal to or close to the best-known solutions and also does not consume much time. In recent past, researchers have proposed metaheuristics that emphasize both exploration and exploitation in their search. Some examples of these types of metaheuristics are Gravitational Search Algorithm (GSA), and Neural network and fuzzy control system based GSA (NFGSA) [33], [34]. These new types of metaheuristics do not require separate steps of exploration and intensification. A direction for future research is to investigate the usefulness of GSA and NFGSA for the FA-problem.

APPENDIX

In the main text, we described the major components of our heuristics, such as maintenance of population using non-dominated sorting and crowding distance. The proposed heuristic also contains several standard genetic operators and a local search method. In this appendix, we briefly describe those components to complete the description of the proposed heuristic.

In the FA problem, we represent the frequencies in a spectrum using integers, i.e., set F is a finite number of integers. Therefore, we can encode the solution as a set of integers. In Section III, we mentioned that a solution is represented by Δ and comprises of m elements, i.e., $\Delta = \{\delta_0, \delta_1, \ldots, \delta_{m-1}\}$. Here m is the number of transmitters and $\delta_i \in F$. The population of the GA consists of N solutions (or chromosomes) that are represented as POP=$\{\Delta_0, \ldots \Delta_{N-1}\}$. Each chromosome $\Delta_i \in$ POP has two attributes: Interference ($\Gamma(\Delta_i)$), and entropy ($\xi(\Delta_i)$). The first step in the proposed heuristic is to apply TS based local search to all chromosomes. The TS heuristic tries to minimize the interference. The application of the TS heuristic to a chromosome Δ_i consists of the following steps.

1) Compute the cost of the solution as $c^b = \Gamma(\Delta_i)$.
2) Repeat {
3) Mutate Δ_i to create a new solution Δ_i'. The mutation consists of the following two steps: (i) In Δ_i, randomly select a transmitter δ_r, (ii) Randomly choose a new frequency f_r for δ_r from F, and (iii) Store the new solution in Δ_i'.

4) Compute the cost of the new solution as $c^c = \Gamma(\Delta_i')$.
5) If $(c^c < c^b)$ then replace Δ_i with Δ_i, and set $c^b = c^c$.
6) If $(c^c \geq c^b)$ and the transmitter δ_r has not been assigned f_r in the last γ_T iterations then replace Δ_i with Δ_i'.
7) } while(stopping criterion is not reached)

The next step is to apply the mutation and crossover operations to create N off-springs. We select parents (i.e., two chromosomes from the current population POP) using the tournament selection method [15] based on the interference values. We applied the single-point crossover to create two offsprings. We apply the mutation operation to each new offspring, and it could change the frequencies of the gene with a probability α_H.

After the creation of off-springs. We combined them with the original population. Now, we select N chromosomes for the population of the next generation from the current population which is a combination of both parents and off-springs. We employed the method of non-dominated sorting and crowding distance to select N chromosomes. The next iteration proceeds in the same way.

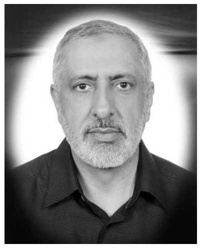
ACKNOWLEDGMENTS

The authors would like to thank Dr. Blair P. Bremberg for assistance in the technical editing of the manuscript.

REFERENCES

[1] J. Wang and Y. Cai, "Multiobjective evolutionary algorithm for frequency assignment problem in satellite communications," *Soft Comput.*, vol. 19, no. 5, pp. 1229–1253, May 2015, doi: 10.1007/s00500-014-1337-2.
[2] F. Luna *et al.*, "Optimization algorithms for large-scale real-world instances of the frequency assignment problem," *Soft Comput.*, vol. 15, no. 5, pp. 975–990, May 2011, doi: 10.1007/s00500-010-0653-4.
[3] G. K. Audhya, K. Sinha, S. C. Ghosh, and B. P. Sinha, "A survey on the channel assignment problem in wireless networks," *Wireless Commun. Mobile Comput.*, vol. 11, no. 5, pp. 583–609, 2011, doi: 10.1002/wcm.898.
[4] W. K. Hale, "Frequency assignment: Theory and applications," *Proc. IEEE*, vol. 68, no. 12, pp. 1497–1514, Dec. 1980.
[5] M. Garey and D. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness* (Series of Books in the Mathematical Sciences). San Francisco, CA, USA: Freeman, 1979. [Online]. Available: https://books.google.com.sa/books?id=fjxGAQAAIAAJ
[6] C. Segura, A. Hernández-Aguirre, F. Luna, and E. Alba, "Improving diversity in evolutionary algorithms: New best solutions for frequency assignment," *IEEE Trans. Evol. Comput.*, vol. 21, no. 4, pp. 539–553, Aug. 2017.
[7] P. Galinier and J.-K. Hao, "A general approach for constraint solving by local search," *J. Math. Model. Algorithms*, vol. 3, no. 1, pp. 73–88, 2004, doi: 10.1023/B:JMMA.0000026709.24659.da.
[8] K. Smith and M. Palaniswami, "Static and dynamic channel assignment using neural networks," *IEEE J. Sel. Areas Commun.*, vol. 15, no. 2, pp. 238–249, Feb. 1997.
[9] N. Funabikiy and Y. Takefuji, "A neural network parallel algorithm for channel assignment problems in cellular radio networks," *IEEE Trans. Veh. Technol.*, vol. 41, no. 4, pp. 430–437, Nov. 1992.
[10] R. Montemanni, J. N. J. Moon, and D. H. Smith, "An improved tabu search algorithm for the fixed-spectrum frequency-assignment problem," *IEEE Trans. Veh. Technol.*, vol. 52, no. 4, pp. 891–901, Jul. 2003.
[11] R. Montemanni and D. H. Smith, "Heuristic manipulation, tabu search and frequency assignment," *Comput. Oper. Res.*, vol. 37, no. 3, pp. 543–551, 2010. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S030505480800141X
[12] S. M. Sait and H. Youssef, *VLSI Physical Design Automation: Theory and Practice*. Singapore: World Scientific, 1999.
[13] L. Liwei and F. Rongshuang, "Simulated annealing algorithm in solving frequency assignment problem," in *Proc. 3rd Int. Conf. Adv. Comput. Theory Eng. (ICACTE)*, vol. 1, Aug. 2010, pp. V1-361–V1-364.
[14] M. Duque-Anton, D. Kunz, and B. Ruber, "Channel assignment for cellular radio using simulated annealing," *IEEE Trans. Veh. Technol.*, vol. 42, no. 1, pp. 14–21, Feb. 1993.
[15] S. M. Sait and H. Youssef, *Iterative Computer Algorithms With Applications in Engineering*. Los Alamitos, CA, USA: IEEE Computer Society Press, 1999.
[16] D. Beckmann and U. Killat, "A new strategy for the application of genetic algorithms to the channel-assignment problem," *IEEE Trans. Veh. Technol.*, vol. 48, no. 4, pp. 1261–1269, Jul. 1999.
[17] C. Y. Ngo and V. O. K. Li, "Fixed channel assignment in cellular radio networks using a modified genetic algorithm," *IEEE Trans. Veh. Technol.*, vol. 47, no. 1, pp. 163–172, Feb. 1998.
[18] G. Dueck and T. Scheuer, "Threshold accepting: A general purpose optimization algorithm appearing superior to simulated annealing," *J. Comput. Phys.*, vol. 90, no. 1, pp. 161–175, Aug. 1990, doi: 10.1016/0021-9991(90)90201-B.
[19] C. Blum and A. Roli, "Metaheuristics in combinatorial optimization: Overview and conceptual comparison," *ACM Comput. Surv.*, vol. 35, no. 3, pp. 268–308, Sep. 2003, doi: 10.1145/937503.937505.
[20] M. Črepinšek, S.-H. Liu, and M. Mernik, "Exploration and exploitation in evolutionary algorithms: A survey," *ACM Comput. Surv.*, vol. 45, no. 3, p. 35, 2013, doi: 10.1145/2480741.2480752.
[21] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002.
[22] Y. Tian, H. Wang, X. Zhang, and Y. Jin, "Effectiveness and efficiency of non-dominated sorting for evolutionary multi- and many-objective optimization," *Complex Intell. Syst.*, vol. 3, no. 4, pp. 247–263, Dec. 2017, doi: 10.1007/s40747-017-0057-5.
[23] X. Lai and J.-K. Hao, "Path relinking for the fixed spectrum frequency assignment problem," *Expert Syst. Appl.*, vol. 42, no. 10, pp. 4755–4767, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0957417415000408
[24] Y. Jin and J. K. Hao, "Effective learning-based hybrid search for bandwidth coloring," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 45, no. 4, pp. 624–635, Apr. 2015.
[25] L. Zhao, H. Wang, and X. Zhong, "Interference graph based channel assignment algorithm for D2D cellular networks," *IEEE Access*, vol. 6, pp. 3270–3279, 2018.
[26] U. F. Siddiqi and S. M. Sait, "A neighborhood search-based heuristic for the fixed spectrum frequency assignment problem," *Arabian J. Sci. Eng.*, vol. 44, no. 4, Apr. 2019, doi: 10.1007/s13369-018-3393-x.
[27] F. Luna, C. Blum, E. Alba, and A. J. Nebro, "ACO vs EAs for solving a real-world frequency assignment problem in GSM networks," in *Proc. 9th Annu. Conf. Genet. Evol. Comput. (GECCO)*, 2007, pp. 94–101. [Online]. Available: http://doi.acm.org/10.1145/1276958.1276972
[28] J. P. Rosca, "Entropy-driven adaptive representation," in *Proc. Workshop Genet. Program., Theory Real-World Appl.* San Mateo, CA, USA: Morgan Kaufmann, 1995, pp. 23–32.
[29] S.-H. Liu, M. Mernik, and B. R. Bryant, "To explore or to exploit: An entropy-driven approach for evolutionary algorithms," *Int. J. Knowl.-Based Intell. Eng. Syst.*, vol. 13, nos. 3–4, pp. 185–206, Dec. 2009. [Online]. Available: http://dl.acm.org/citation.cfm?id=1735203.1735211
[30] S. Garcia, D. Molina, M. Lozano, and F. Herrera, "A study on the use of non-parametric tests for analyzing the evolutionary algorithms 'behaviour: A case study on the CEC'2005 special session on real parameter optimization," *J. Heuristics*, vol. 15, no. 6, p. 617, May 2008, doi: 10.1007/s10732-008-9080-4.
[31] U. F. Siddiqi, Y. Shiraishi, and S. M. Sait, "Memory-efficient genetic algorithm for path optimization in embedded systems," *Inf. Media Technol.*, vol. 8, no. 2, pp. 270–278, 2013.
[32] W. Bi, H. R. Maier, and G. C. Dandy, "Impact of starting position and searching mechanism on the evolutionary algorithm convergence rate," *J. Water Resour. Planning Manage.*, vol. 142, no. 9, p. 04016026, 2016.
[33] D. Pelusi, R. Mascella, L. Tallini, J. Nayak, B. Naik, and A. Abraham, "Neural network and fuzzy system for the tuning of gravitational search algorithm parameters," *Expert Syst. Appl.*, vol. 102, pp. 234–244, Jul. 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0957417418301143
[34] E. Rashedi, H. Nezamabadi-Pour, and S. Saryazdi, "GSA: A gravitational search algorithm," *J. Inf. Sci.*, vol. 179, no. 13, pp. 2232–2248, 2009. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0020025509001200

UMAIR F. SIDDIQI (M'12) was born in Karachi, Pakistan, in 1979. He received the B.E. degree in electrical engineering from the NED University of Engineering and Technology, Karachi, in 2002, the M.Sc. degree in computer engineering from the King Fahd University of Petroleum and Minerals (KFUPM), Dhahran, Saudi Arabia, in 2007, and the Dr. Eng. degree from Gunma University, Japan, in 2013. He is currently a Research Engineer with the Center of Communications and Information Technology Research of Research Institute, KFUPM. He has authored over 25 research papers in international journals and conferences. He also holds two U.S. patents. His research interests include optimization, metaheuristics, soft computing, and machine learning.



SADIQ M. SAIT was born in Bengaluru. He received the bachelor's degree in electronics engineering from Bangalore University in 1981 and the master's and Ph.D. degrees in electrical engineering from KFUPM in 1983 and 1987, respectively. He is currently a Professor of computer engineering and the Director of the Center for Communications and IT Research, KFUPM. He has authored over 200 research papers, contributed chapters to technical books, and lectured in over 25 countries. He is also the principle author of two books. He received the Best Electronic Engineer Award from the Indian Institute of Electrical Engineers, Bengaluru, in 1981.