A Novel Genetic Trajectory Planning Algorithm With Variable Population Size for Multi-UAV-Assisted Mobile Edge Computing System

MUHAMMAD ASIM1, WALI KHAN MASHWANI2, SAMIR BRAHIM BELHAOUARI3, (Senior Member, IEEE), AND SAIMA HASSAN4

1School of Computer Science and Engineering, Central South University, Changsha 410083, China
2Institute of Numerical Sciences, Kohat University of Science & Technology, Kohat 26000, Pakistan
3College of Science and Engineering, Hamad Bin Khalifa University, Doha, Qatar
4Institute of Computing, Kohat University of Science & Technology, Kohat 26000, Pakistan

Corresponding authors: Wali Khan Mashwani (mashwanigr8@gmail.com) and Samir Brahim Belhaouari (sbelhaouari@hbku.edu.qa)

This work was supported by Qatar National Library (QNL) to conduct study of this article.

ABSTRACT This paper presents a multi-unmanned aerial vehicle (UAV)-assisted mobile edge computing (MEC) system, where multiple UAVs (variable number of UAVs) are deployed to serve Internet of Things devices (IoTDs). We aim to minimize the sum of hovering and flying energies of UAVs by optimizing the trajectories of UAVs. The problem is very complicated as we have to consider the deployment of stop points (SPs), the association between UAVs and SPs, and the order of SPs for UAVs. To solve the problem, this paper proposed a novel genetic trajectory planning algorithm with variable population size (GTPA-VP), which consists of two phases. In the first phase, a genetic algorithm (GA) with a variable population size is used to update the deployment of SPs. Accordingly, a multi-chrome GA is adopted to find the association between UAVs and SPs, an optimal number of UAVs, and the optimal order of SPs for UAVs. The effectiveness of the proposed GTPA-VP is demonstrated through several experiments on a set of ten instances with up to 200 IoTDs. It is evident from the experimental results that the proposed GTPA-VP outperforms the benchmark algorithms in terms of the energy consumption of the system.

INDEX TERMS Mobile edge computing, unmanned aerial vehicle, evolutionary algorithm, multi-chrome genetic algorithm.

I. INTRODUCTION

With the development of mobile communication systems, a huge number of new resource-intensive and latency-sensitive applications are emerging, such as virtual reality, and online gaming [1]. Such applications are usually sensitive to latency and require huge computational resources. Due to limitations of the Internet of Things devices (IoTDs), it is very difficult to execute these tasks on them.

Mobile edge computing (MEC) is a promising technology to address the above-mentioned issue. It can provide service with low latency and high reliability for IoTDs. It can execute tasks of IoTDs at the nearby edge cloud and sends back the results to IoTDs [1]. Due to the shorter physical distance between MEC’s server/edge cloud and IoTDs, it consumes less energy as compared to mobile cloud computing. However, it is still lacking in fulfilling the requirements of IoTDs, as the locations of edge clouds are usually fixed and cannot be adjusted flexibly according to the requirements of IoTDs. Therefore, it cannot provide timely services during natural disasters as the terrestrial communication link may be broken/lost.

To cope with these ever-increasing demands, unmanned aerial vehicle (UAV) is considered one of the most promising technologies. Compared to the traditional MEC systems that utilize the terrestrial fixed base stations, UAV-aided MEC systems are more cost-effective and likely to achieve a better quality of service due to their appealing properties of...
flexible deployment, fully controllable mobility, and low cost. In fact, with the assistance of UAVs, the performance of MEC systems such as data rate and latency can be significantly enhanced by establishing the line-of-sight communication links between UAVs and IoTDs. In addition, through dynamically adjusting the flying state, UAVs are capable of improving communication performance in MEC systems. Moreover, UAVs can provide emergency services to the target devices in case of emergency. Hence, the utilization of UAVs is expected to enhance data collection from IoTDs.

Recently, UAVs have received much attention and importance due to their autonomy and flexibility and have been extensively utilized in various fields, such as wireless communication [2]–[4], military [5], [6], surveillance and monitoring [7], [8], smart agriculture [9], delivery of medical supplies [10], and rescue operations [11], [12]. In the most recent research work, UAVs have been used to enhance the capabilities of the MEC systems. For example, Wang et al. [13] studied a multi-UAV-enabled MEC system, where a number of UAVs were deployed as flying edge clouds for large-scale IoT systems. They jointly optimized the deployment of UAVs and task scheduling. Zhang et al. [14] proposed a UAV-assisted MEC for efficient multitasking scheduling to minimize completion time. Garg et al. [15] studied the application of a UAV-empowered MEC system in cyber-threat detection of smart vehicles. Zhang et al. [16] studied the energy-aware dynamic resource allocation problem for a UAV-assisted MEC system over the Internet of Vehicles. They explored the energy-aware dynamic resource allocation problem by taking into account partial computation offloading, social content caching, and radio resource scheduling. Du et al. [17] optimized joint resource and workflow scheduling in a UAV-enabled wirelessly powered MEC system.

Moreover, in order to fully exploit the potential of UAV-assisted MEC systems, some researchers have explored appropriate path planning and trajectory designing of UAVs. For example, Zhang et al. [18] proposed an energy-efficient trajectory optimization scheme for UAV-assisted Internet of Things networks. They deployed a single UAV powered by both solar energy and charging stations, resulting in sustainable communication services. They optimized the trajectory design of UAV by jointly considering the average data rate, the total energy consumption, and the fairness of coverage for the Internet of Things terminals. Wang et al. [19] introduced a multi-agent deep reinforcement learning-based trajectory planning algorithm for UAV-aided MEC framework, where several UAVs having different trajectories fly over the target area and support the ground IoT systems. They aimed to jointly optimize the geographical fairness among all the IoT systems, the fairness of each UAV’s UE-load, and the overall energy consumption of IoT systems. They proposed a multi-agent deep reinforcement learning-based trajectory control algorithm for managing the trajectory of each UAV independently. Given the UAVs’ trajectories, a low-complexity approach is introduced for optimizing the offloading decisions of IoT systems. Liu et al. [20] investigated the UAV-enabled wirelessly powered cooperative MEC system, where a UAV is deployed with an energy transmitter and a MEC server providing both energy and computing services to sensor devices. They formulated an optimization problem to minimize the required energy of UAV by mutually optimizing the CPU frequencies, the offloading amount, the transmit power, and the UAV’s trajectory. They proposed a successive convex approximation-based algorithm and a decomposition and iteration-based algorithm to handle the non-convex problem. Tun et al. [21] studied the problems of energy-efficient UAV trajectory optimization, resource allocation, and task offloading in a UAV-assisted MEC system. They aimed at minimizing not only the energy consumption of mobile devices but also UAV’s propulsion and computing power. They introduced a block successive upper-bound minimization algorithm, which is a powerful tool for non-convex and nonsmooth problem. Wu and Zhang [22] considered a practical scenario of UAVs in an orthogonal frequency-division multiple access (OFDMA) system. They proposed an iterative block coordinate descent approach for optimizing the UAV’s trajectory and OFDMA resource allocation to maximize the minimum average throughput of IoT systems. Diao et al. [23] optimized joint trajectory and data allocation to minimize the maximum energy consumption. Jeong et al. [24] studied the bit allocation and trajectory planning under latency and energy budget constraints. Hu et al. [25] developed a UAV-assisted relaying and MEC system, where the UAV can act as the MEC server or the relay. They proposed a joint task scheduling and trajectory optimization algorithm to minimize the weighted sum energy consumption of UAVs and IoT systems subject to task constraints. Yang et al. [26] presented the sum power minimization problem for a UAV-enabled MEC network. To solve the non-convex sum power minimization problem, they proposed a low-complexity algorithm with solving three subproblems iteratively. They proposed a compressive sensing-based algorithm for the UE association subproblem. For the computation capacity allocation subproblem, the optimal solution is obtained in closed form. They used a one-dimensional search method for the optimal solution of the location planning subproblem. Finally, they proposed a fuzzy C-means clustering-based algorithm to obtain a feasible solution. Zhang et al. [27] investigated the computation efficiency maximization problem in the multi-UAV assisted MEC network. They jointly optimized user association, computation, and communication resource allocation as well as trajectory scheduling of UAVs. To tackle the problem, they proposed an iterative optimization algorithm with a double-loop structure. In the outer loop, they have adopted Dinkelbach method to find the optimal computation efficiency, while in the inner loop, they proposed a joint optimization algorithm for user association, resource allocation, and trajectory scheduling. Huang et al. [28] studied a multi-UAV-assisted MEC system, where the UAVs act as edge servers to provide computing services for IoT systems. They presented an energy-efficient trajectory planning algorithm (TPA) to minimize the energy consumption of the
system. Zeng et al. [29] proposed an efficient algorithm to optimize the trajectory of UAV, including the hovering locations and duration. They formulated the problem as a traveling salesman problem to minimize the energy consumption of UAV.

From the above introduction and related literature, it can be seen that the deployment of multiple UAVs in MEC systems remains scarce in current studies. In fact, collaboration among multiple UAVs can improve the capabilities of the system. In addition, variable number of UAVs have rarely been considered in current studies. In this paper, we consider the trajectory planning problem in a multi-UAV-assisted MEC system with a variable number of UAVs. Compared with the conventional trajectory planning problems, this research problem is more challenging due to the fact that the deployment of the stop points (SPs) of UAVs is unknown in prior. Furthermore, in case of a multi-UAV-assisted MEC system, we need to consider the association between UAVs and SPs.

The main contributions of this paper are summarized as follows:

- A new multi-UAV-assisted MEC system is proposed and formulated with the aim of minimizing the sum of the hovering energy and flying energy of the system by considering the deployment including the number and locations of SPs, the number of UAVs, and their association with SPs, and the order of SPs.  
- A GA trajectory planning algorithm with variable population size (GTPA-VP) is proposed, which consists of two phases. First, operators of continuous genetic algorithm (GA) [30] with variable population size are proposed to optimize the deployment of SPs. Subsequently, MCGA is adopted to associate UAVs with SPs and predict the optimal number of UAVs as well as construct the order of SPs in each cluster.  
- Extensive experiments have been carried out on a set of ten instances with up to 200 IoTDs. The experimental results show the effectiveness of GTPA-VP.

The remainder of this paper is organized as follows. Section II introduces the system model, including the problem formulation of the proposed system. Section III presents the details of our proposed algorithm GTPA-VP. In Section IV, the simulation results are discussed. Finally, Section V concludes this paper.

**II. SYSTEM MODEL**

As shown in Fig. 1, we consider there are \( i \in \mathcal{N} = \{1, 2, \ldots, N\} \) IoTDs and \( j \in \mathcal{M} = \{1, 2, \ldots, M\} \) UAVs. UAVs fly over all the IoTDs to collect the data. We assume that the UAVs will stop at SPs for some time and the IoTDs can send the sensing data to the UAVs. We assume UAVs will hover over \( t \in \mathcal{T} = \{1, 2, \ldots, T\} \) SPs at the air in which each SP may last for \( T^{\text{max}} \) seconds, where \( T^{\text{max}} \) is the fixed value. Therefore, one has

\[
a_{ij}[t] = \begin{cases} 1, & \forall i \in \mathcal{N}, \forall t \in \mathcal{T}, \forall j \in \mathcal{M}, \\ 0, & \text{otherwise}. \end{cases}
\]  

(1)

where \( a_{ij}[t] = 1 \) denotes that the \( i \)-th IoTD decides to send its sensing data to \( j \)-th UAV at \( t \)-th SP, while \( a_{ij}[t] = 0 \) indicates otherwise. Then, one has

\[
\sum_{t=1}^{T} \sum_{j=1}^{M} a_{ij}[t] = 1, \quad i \in \mathcal{N}
\]  

(2)

which denotes that one IoTD should choose one UAV at each SP to send its sensing data.

We assume that the UE always sends data to the closest UAV at each SP. Then, one has

\[
a_{ij}[t] = \begin{cases} 1, & \text{if } (i, j, t) = \text{argmin}_{i \in \mathcal{N}, j \in \mathcal{M}}(d_{ij}[t]), \\ 0, & \text{otherwise}. \end{cases}
\]  

(3)

Assume that at each SP \( t \), \( j \)-th UAV can accept at most \( U_{j} \) IoTDs. Therefore, one has

\[
\sum_{i=1}^{N} a_{ij}[t] \leq U_{j}, \quad t \in \mathcal{T}, \quad j \in \mathcal{M}
\]  

(4)

We assume that \( i \)-th IoTD may collect \( D_{i} \) amount of data which intends to send it to the UAV. Then, the transmission time \( (T_{i}^{Tr}) \) to send the data from IoTD to UAV at \( t \)-th SP is as

\[
T_{i}^{Tr}[t] = \frac{D_{i}}{r_{j}[t]}, \quad \forall j \in \mathcal{M}, \quad t \in \mathcal{T}
\]  

(5)

where \( r_{j}[t] \) is the data rate which is given by (14).

The processing time \( (T_{i}^{C}) \) of the data in UAV can be obtained as:

\[
T_{i}^{C}[t] = \frac{F_{i}}{f_{j}[t]}, \quad \forall j \in \mathcal{M}, \quad \forall t \in \mathcal{T}
\]  

(6)

where \( F_{i} \) is the CPU cycles that a task may need to process and \( f_{j}[t] \) is the computation capacity of the UAV assigned to each data processing procedure, where

\[
\sum_{i=1}^{N} f_{i}[t] \leq f_{\text{max}}, \quad j \in \mathcal{M}, \quad t \in \mathcal{T}
\]  

(7)
where \( f_{\text{max}} \) is the maximal computing power the UAV can provide to each IoTD. Also, the total time \( T_i[r] \) for transmitting and processing task of UE \( i \) is given as:

\[
T_i[r] = T_i^C[r] + T_i^F[r], \quad i \in \mathcal{N}, \ t \in \mathcal{T}
\]  

(8)

Then, one can have

\[
T_i[r] \leq T_{\text{max}}
\]  

(9)

Assume that the coordinate of \( i \)-th IoTD is \((x_i, y_i)\) and the coordinate of the \( j \)-th UAV at the \( t \)-th SP is \((X_j[r], Y_j[r], H)\). Additionally, assume that the UAV’s trajectory can be characterized by a sequence of location \( q[t] = [X_j[r], Y_j[r], H]^T \), where \( H \) is a fixed value. In addition, all UAVs take off from the same initial position \( q[0] \) and land in at the same position \( q[0] \). Also, we have

\[
||q[t+1] - q[t][2|| \leq S_{\text{max}}, \quad t = 0, \ldots, T
\]  

(10)

where \( S_{\text{max}} = \sqrt{V_{\text{max}} \cdot T_{\text{max}}} \) is the maximum horizontal distance that the UAV can travel and \( V_{\text{max}} \) is the maximum speed.

Then, the horizontal distance between the \( i \)-th IoTD and the \( j \)-th UAV at \( t \)-th SP is measured as

\[
d_{ij}[t] = \sqrt{(X_j[r] - x_i)^2 + (Y_j[r] - y_i)^2}, \quad \forall i \in \mathcal{N}, \ \forall t \in \mathcal{T}
\]  

(11)

Also, the distance between the \( i \)-th IoTD and the \( j \)-th UAV at \( t \)-th SP is measured as

\[
d_{ij}[t] = \sqrt{R_{ij}[t]^2 + H^2}, \quad \forall i \in \mathcal{N}, \ \forall t \in \mathcal{T}
\]  

(12)

Then, the channel power gain can be given as

\[
h_{ij}[t] = \frac{\beta_0}{d_{ij}[t]^2}
\]  

(13)

where \( \beta_0 \) denotes the channel power gain at the reference distance 1m.

If IoTD \( i \) decides to offload data to the UAV \( j \) at SP \( t \), then the data rate can be given as

\[
r_{ij}[t] = B \log_2 \left( 1 + \frac{p_{ij}^F h_{ij}[t]}{\sigma^2} \right)
\]  

(14)

where \( \sigma^2 \) is the noise power and \( p_{ij}^F \) is the transmission power, which is constrained by

\[
p_{ij}^F \leq P_{\text{max}}
\]  

(15)

Assume the flying energy of the UAV is proportional to the flying distance, then the flying energy can be calculated as

\[
E_j^F = P^F \sum_{t=1}^{T_j-1} ||q[t+1] - q[t][2||^2
\]  

(16)

Also, for the hovering energy, one can have

\[
E_j^H = P^H \cdot T_j \cdot T_{\text{max}}
\]  

(17)

In this study, we aim to optimize the trajectories of UAVs in order to minimize the energy consumption of the system. Thus, we can have the optimization problem as follows.

\[
\mathcal{P} : \min_{T, \{q[t]\}, M} \sum_{j=1}^{M} (E_j^F + E_j^H)
\]  

(18a)

subject to:\n
\[
a_{ij}[t] = \{0, 1\}, \ \forall i \in \mathcal{N}, \ \forall t \in \mathcal{T}, \ \forall j \in \mathcal{M},
\]  

(18b)

\[
\sum_{i=1}^{T} \sum_{j=1}^{M} a_{ij}[t] = 1, \ i \in \mathcal{N},
\]  

(18c)

\[
\sum_{i=1}^{N} a_{ij}[t] \leq U_j, \ t \in \mathcal{T}, \ j \in \mathcal{M},
\]  

(18d)

\[
\sum_{i=1}^{N} f_{ij}[t] \leq f_{\text{max}}, \ j \in \mathcal{M}, \ t \in \mathcal{T}
\]  

(18e)

\[
T_i[r] \leq T_{\text{max}},
\]  

(18f)

\[
||q[t+1] - q[t][2|| \leq S_{\text{max}}, \ t = 0, \ldots, N,
\]  

(18g)

\[
p_{ij}^F \leq P_{\text{max}}
\]  

(18h)

\[
X_{\text{min}} \leq X_j[r] \leq X_{\text{max}}, \ \forall j \in \mathcal{M}, \ t \in \mathcal{T}
\]  

(18i)

\[
Y_{\text{min}} \leq Y_j[r] \leq Y_{\text{max}}, \ \forall j \in \mathcal{M}, \ t \in \mathcal{T}
\]  

(18j)

where the objective function is the sum of hovering energy and flying energy of UAVs and (18(i)) and (18(j)) present the lower and upper bounds of the X-axis and Y-axis, respectively.

### III. PROPOSED ALGORITHM

#### A. MOTIVATION

By analyzing the proposed system model and problem formulation in Section II, it is clear that (18(a)) is a non-convex, NP-hard, and nonlinear optimization problem that cannot be solved by traditional optimization methods. Evolutionary algorithms (EAs) are a kind of population-based heuristic gradient-free optimization algorithms that have the potential to address the above-mentioned problem (18(a)). However, EAs still face some issues in solving (18(a)).

- To solve (18(a)), we need to consider the number of UAVs, the number of SPs and their locations, which one UAV will visit which specific SPs, and in what order the UAV will visit the assigned SPs. Therefore, it is a complicated/complex problem to be tackled by the EAs directly.
- (18(a)) contains integer decision variable M and number of SPs \( T_j \) for UAV j, binary variable \( a_{ij}[t] \), and continuous variables \((X_j \) and \( Y_j)\). Therefore, it is a mixed decision variable problem, which is challenging to be solved by the EAs [13], [31].
- Since, the number of UAVs is unknown in prior, the clustering of SPs into different clusters requires an
unsupervised scheme (i.e., free of initialization/parameter-free clustering algorithm) that can group closely spaced SPs into different clusters automatically and can also simultaneously find an optimal number of clusters/UAVs [32].

- Since the number of SPs is unknown in prior, thus the length of individual is not fixed. The commonly used crossover and mutation operators of EAs are constructed for fixed-length individuals [33]. Therefore, the direct use of EAs would be ineffective.

In this paper, we proposed an algorithm called GTPA-VP to design the trajectories of UAVs. The proposed algorithm consists of two phases: the deployment of SPs and the association between UAVs and SPs and the order of SPs.

The main technical advantages of the proposed algorithm are given as:

- Considering the strong coupling among the deployment of SPs, the association between UAVs and SPs, and the order of SPs. GTPA-VP plans the trajectories of UAVs at each iteration through two phases: updating the deployment of SPs and the association between UAVs and SPs and constructing the optimal order of SPs for UAVs.

- In GTPA-VP, the deployment of SPs is addressed by proposing a GA with variable population size. Each individual represents the location of an SP; thus, the whole population represents a whole deployment, rather than a set of deployments. Since the length of the individual is fixed, we can directly adopt the commonly used crossover and mutation operators for updating the deployment of SPs.

- In GTPA-VP, the association between UAVs and SPs and the order of SPs were jointly addressed by adopting MCGA [34], [35]. MCGA can associate UAVs with SPs without knowing the number of clusters in prior as well as can predict the optimal number of clusters/UAVs. In addition, it can also construct the order of SPs for all UAVs.

### B. GTPA-VP

The framework of GTPA-VP is given in Algorithm 1. In the initialization, the locations of SPs are produced randomly, forming an initial population \( POP = (X_1, Y_1), (X_2, Y_2), \ldots, (X_{max}, Y_{max}) \). Accordingly, MCGA is adopted to group SPs into different clusters (i.e., UAVs are associated with SPs) and construct the order of SPs in each cluster. After that, \( POP \) is evaluated via Eq. (18(a)); if it is feasible, the initial population is generated successfully; otherwise, the initialization is repeated until it is feasible or the number of fitness evaluations (\( FEs \)) is not less than maximum \( FEs \) (\( FEs_{max} \)). Accordingly, an offspring population \( POP_C \) is first produced via continuous GA in Algorithm 2 during the evolution. After that, we construct three new populations \( POP_1, POP_2, \) and \( POP_3 \) using Algorithm 3. Then, the SPs in \( POP_1, POP_2, \) and \( POP_3 \) are grouped into different clusters along with the construction of the order of SPs in each cluster by using MCGA in Algorithm 4. Accordingly, the three new populations \( POP_1, POP_2, \) and \( POP_3 \) are evaluated using Eq. (18(a)). Finally, we replace \( POP \) with the feasible population among one of \( POP_1, POP_2, \) and \( POP_3 \) with the greatest performance improvement against \( POP \), if at least one feasible population exists among \( POP_1, POP_2, \) and \( POP_3 \) then.

![Algorithm 1 General Framework of GTPA-VP](image)

### C. DEPLOYMENT OF SPs

The deployment of SPs is updated by using operators of continuous GA [30], which is a simple, most popular, and effective EA and has been successfully applied in many fields [36], [37]. More specifically, random selection, continuous crossover, and continuous mutation operators were adopted in GTPA-VP to generate an offspring population \( POP_C \) (i.e., locations of new SPs). The individuals of \( POP_C \) are adopted to update parent population \( POP \) (i.e., locations of SPs can be updated). Thus, the locations of SPs can be updated by using the above process. Since each individual in GA represents a location of SP. Therefore,
Algorithm 2 Updating Deployment of SPs Using Continuous GA

1: Initialize: \( POP_C = \emptyset \), crossover probability \( P_c \), mutation probability \( P_m \);
2: for \( k = 1: 2; \) \( [POP] \) do
3: \( x_1, x_2 \) ← Apply random selection to select parents;
4: \( y_1, y_2 \) ← Apply continuous crossover \((x_1, x_2)\);
5: \( o_1, o_2 \) ← Apply continuous mutation to generate \( O_i \);
6: \( O_i \leftarrow \{o_1, o_2\} \);
7: \( POP_C = POP_C \cup O_i \);
8: end for
9: \( POP_C \);
10: Use individuals of \( POP_C \) to generate three new populations in Algorithm 3;

Algorithm 3 Generating Three New POP

1: \( POP_C \);
2: \( POP_1 \) ← insert the \( i \)th individual in \( POP_C \) to \( POP_P \);
3: \( POP_2 \) ← replace a random individual in \( POP \) by using the \( i \)th individual in \( POP_C \);
4: \( POP_3 \) ← delete a random individual in \( POP \).

![Chromosome Diagram](image)

**FIGURE 2.** Example of route planning system with 4 UAVs and 15 SPs, where \( \{1, 2, \ldots, 15\} \) represent SPs.

the whole population represents the locations of all SPs. Hence, the number of SPs is equal to the population size. Thus, the population size is kept variable during evolution while updating the number of SPs i.e., the population size can be increased, kept unchanged, or reduced.

By using Algorithm 3, we construct three new population \( POP_1, POP_2, \) and \( POP_3 \) by inserting, replacing, and removing an individual in/from population \( POP \), respectively. More specifically, \( POP_1 \) is constructed by inserting an individual \( i \) from offspring \( POP_C \) to \( POP \). \( POP_2 \) is constructed by replacing \( i \)th individual in \( POP_C \) with an individual in \( POP \), and \( POP_3 \) is constructed by removing a random individual from \( POP \). Therefore, the population sizes of \( POP_1, POP_2, \) and \( POP_3 \) are one more than, the same as, and one less than that of \( POP \), respectively. Thus, the population size is varying during the updation of SPs by using \( POP_1, POP_2, \) or \( POP_3 \).

D. ASSOCIATION BETWEEN UAVS AND SPs AND THE ORDER OF SPs

In this subsection, we associate UAVs with SPs (i.e., SPs are grouped into different clusters and then a UAV is assigned to each cluster to visit its SPs) and construct the order of SPs for UAVs. In GTPA-VP, we used MCGA [34], [35], [38] to jointly handle association between UAVs and SPs (i.e., grouped closely SPs into the same cluster and a UAV is assigned to visit its SPs) and the order of SPs for UAVs. Moreover, this algorithm can also predict the optimal number of UAVs.

There are two sets of mutation operators, the so-called In-route mutations and the cross-route mutations. In-route

![Algorithm 4 MCGA Algorithm](image)

**Algorithm 4 MCGA Algorithm**

1: Initialize: distance matrix \( dmat \), population size \( POPSize = 8 \), minimum tour for each UAV \( mintour = 2 \), maximum iteration \( maxIter = 200 \), maximum tour for each UAV \( S_{max} = 1000 \), and penalty rate \( penalty-rate = 100 \);
2: for \( iter = 1: Max-Iter \) do
3: for \( p = 1: POPSize \) do
4: total cost \( d = 0 \);
5: for \( m = 1: length \) of individual \( p \) do
6: UAV \( m \) = pick route of UAV \( m \) from individual \( p \);
7: \( d_m = 0 \);
8: if UAV \( m \) ≠ \( \emptyset \) then
9: \( d_m = d_m + dmat(1, UAV_m(1))/max \);
10: for \( i = 1:length(UAV_m)-1 \) do
11: \( d_m = d_m + dmat(UAV_m(t), UAV_m(t + 1))/max \);
12: end for
13: \( d_m = d_m + dmat(UAV_m(end), 1)/max \);
14: if \( d_m > S_{max} \) then
15: \( d_m = d_m + (d_m - S_{max}) * rate - penalty \);
16: end if
17: end if
18: \( d = d + d_m \);
19: end for
20: \( total - dist(p) = d \);
21: end for
22: POP\(_N\) ← Generate New POP by using Algorithm 5;
23: end for

Algorithm 5 Operators of GA

1: rand-grouping = randperm(pop-size);
2: for \( p = 8:8:pop-size \) do
3: rpop = select 8 random individuals;
4: best-rpop = find best individual among 8;
5: for \( k = 1:8 \) do
6: Flip ← Apply Flip to flip 2 genes of chromosome;
7: Swap ← Apply Swap to transpose genes from two random chromosome;
8: Slide ← Apply Slide operator to slide the genes in/among random chromosomes;
9: Crossover ← Apply one point crossover;
10: end for
11: end for
12: OUTPUT: NEW POP \( POP_N \)
TABLE 1. Parameters setting.

| Parameter | Value         | Parameter | Value         |
|-----------|---------------|-----------|---------------|
| $D_i$ (i ∈ M) | [1, 10]$^n$ M | $P$       | 0.1 W         |
| $P^H$     | 1000          | $V_{max}$ | 20 m/s        |
| $P^F$     | 1000          | $\sigma^2$| -250 dBm      |
| $\alpha$  | -30 dB        | $Q(0)$   | [0 0 200]     |
| $\beta$   | 2.8           | $H^G$    | 200           |
| $X_{max}$ | 1000          | $Y_{max}$ | 1000          |
| $B$       | 1 MHz         | $U_i$     | 5             |

Table: Parameters setting.

### IV. SIMULATION RESULTS

#### A. PARAMETER SETTINGS

The parameter settings of the proposed multi-UAV-assisted MEC system are presented in Table 1 [28]. We have used the same parameters for all compared algorithms used in this paper for a fair comparison. The parameter settings were kept the same as used in other articles such as: [28]. We have tested ten instances with up to 200 IoTDs to evaluate the performance of GTPA-VP. We assume that all the UEs are distributed randomly in a 1000 m × 1000 m square region. In GTPA-VP, we set the following parameters: $P_m = 0.95$ and $P_m = 0.01$. The maximum number of fitness evaluations ($FES_{max}$) is set to 5000 and 20 runs are implemented independently on each algorithm. The mean energy consumption and the standard deviation of the proposed system over 20 runs are denoted by mean EC and Std Dev, respectively. In addition, we performed the Wilcoxon rank-sum test at 0.05 significant level. In the simulation results, we used $\mid$, $\triangleright$, and $\approx$ to show that GTPA-VP performs significantly better than, worse than, and similar to its competitors. Since most of the existing solutions are designed for different systems and problems, therefore, are not applicable to our proposed system. However, we have compared our algorithm with some existing approaches such as TPA [28], etc. In addition, we have compared each phase of the proposed GTPA-VP with an existing method to show its effectiveness.

In order to show the effectiveness of GTPA-VP, we designed algorithms called Kmeans-Greedy, GAC-Greedy, and TPA [28]. Kmeans-Greedy uses K-means algorithm [39] for clustering and a greedy algorithm for the order of SPs. GAC-Greedy uses GA clustering(GAC) [40] and a greedy algorithm for the order of SPs. K-means algorithm needs the number of UAVs in advance which was set to 4 in this paper, while GAC algorithm does not require the number of UAVs in prior. In GAC, the number of iterations and population size were set to 50 and 10, respectively. The deployment of SPs was kept the same in both Kmeans-Greedy and GAC-Greedy as used in GTPA-VP. The parameters of TPA were kept unchanged. TPA also requires the number of UAVs for clustering SPs, which was set to 4. The mean EC and Std Dev of GTPA-VP, Kmeans-Greedy, GAC-Greedy, and TPA are presented in Table 2. In addition, Figure 3 presents the evolution of the mean EC by GTPA-VP, Kmeans-Greedy, GAC-Greedy, and TPA on ten instances. One can see from Table 2 and Figure 3 that the proposed GTPA-VP performs better than Kmeans-Greedy, GAC-Greedy, and TPA in terms of mean EC. In addition, the statistical test results of GTPA-VP, Kmeans-Greedy, GAC-Greedy, and TPA are summarized at the bottom of Table 2, which show that GTPA-VP is significantly better than Kmeans-Greedy, GAC-Greedy, and TPA. The superiority of GTPA-VP against compared algorithms can be attributed as 1) in GTPA-VP, the association between UAVs and SPs and the trajectories of UAVs are jointly addressed which may lead to good performance and 2) in GTPA-VP, MCGA is used to construct the order of SPs for UAVs, which is a known famous EA for its good convergence.

#### B. EFFECTIVENESS OF DEPLOYMENT OF SPs

In GTPA-VP, GA operators with variable population sizes are proposed for updating the deployment of SPs. To prove its effectiveness, we compare the proposed algorithm with two
TABLE 2. Experimental results of GTPA-VP, Kmeans-Greedy, GAC-Greedy, and TPA in terms of average EC over 20 runs.

| N   | GTPA-VP Mean EC (Std Dev) | Kmeans-Greedy Mean EC (Std Dev) | GAC-Greedy Mean EC (Std Dev) | TPA Mean EC (Std Dev) |
|-----|---------------------------|---------------------------------|-------------------------------|-----------------------|
| 20  | 1.30E+05 (1.24E+04)       | 1.31E+05 (1.31E+04)            | 2.99E+06 (8.07E+05)          | 1.42E+05 (2.58E+04)   |
| 40  | 2.57E+05 (1.33E+04)       | 1.37E+06 (1.26E+05)            | 6.39E+06 (8.50E+05)          | 2.52E+06 (1.57E+06)   |
| 60  | 5.88E+05 (2.18E+04)       | 2.83E+06 (1.04E+05)            | 1.11E+07 (1.94E+06)          | 3.93E+06 (1.53E+06)   |
| 80  | 5.48E+05 (2.27E+04)       | 5.12E+06 (1.35E+06)            | 1.30E+07 (7.70E+05)          | 5.12E+06 (1.39E+06)   |
| 100 | 4.67E+05 (2.19E+04)       | 5.95E+06 (1.71E+06)            | 1.73E+07 (1.17E+06)          | 6.63E+06 (1.90E+06)   |
| 120 | 8.22E+05 (3.68E+04)       | 7.42E+06 (3.10E+06)            | 1.96E+07 (8.72E+05)          | 7.64E+06 (2.34E+06)   |
| 140 | 9.79E+05 (3.71E+04)       | 8.45E+06 (2.14E+06)            | 2.21E+07 (1.03E+06)          | 8.39E+06 (2.58E+06)   |
| 160 | 1.12E+06 (4.12E+04)       | 9.91E+06 (2.52E+06)            | 2.67E+07 (1.39E+06)          | 8.94E+06 (2.19E+06)   |
| 180 | 1.24E+06 (3.34E+04)       | 1.78E+07 (3.03E+06)            | 2.75E+07 (1.26E+06)          | 1.18E+07 (2.63E+06)   |
| 200 | 1.39E+06 (3.91E+04)       | 1.08E+07 (2.78E+06)            | 3.23E+07 (1.12E+06)          | 1.20E+07 (2.57E+06)   |

†, ‡,  ≈ 10/00 10/00 10/00

FIGURE 3. Evolution of the mean EC by GTPA-VP, Kmeans-Greedy, GAC-Greedy, and TPA on nine instances over 20 runs. Where N shows the number of IoTDs.

other variants named RAN-GTPA and DE-GTPA. In RAN-GTPA, the locations of SPs are updated randomly, while in DE-GTPA, operators of DE [41] are used to update the deployment of SPs. Table 3 presents the mean EC and Std Dev of GTPA-VP, RAN-GTPA, and DE-GTPA on ten instances, which shows that the mean EC of GTPA-VP is significantly smaller than RAN-GTPA and DE-GTPA on all instances. In addition, the statistical test results of GTPA-VP, RAN-GTPA, and DE-GTPA are summarized at the bottom of Table 3. It is clear that GTPA-VP provides significantly better results than RAN-GTPA and DE-GTPA. The better performance of the deployment of SPs is due to the following reason: since GTPA-VP can simultaneously update the number and locations of SPs and the dimension of the search space is always fixed, therefore, it can achieve better performance.
TABLE 3. Experimental results of GTPA-VP, RAN-GTPA, and DE-GTPA in terms of average EC over 20 runs.

| N   | GTPA-VP Mean EC (Std Dev) | RAN-GTPA Mean EC (Std Dev) | DE-GTPA Mean EC (Std Dev) |
|-----|---------------------------|----------------------------|---------------------------|
| 20  | 1.30E+05 (1.28E+04)       | 1.34E+05 (9.71E+03)       | 1.33E+05 (1.69E+04)       |
| 40  | 2.57E+05 (1.33E+04)       | 2.58E+05 (1.68E+04)       | 2.63E+05 (1.66E+04)       |
| 60  | 3.88E+05 (2.18E+04)       | 3.91E+05 (2.27E+04)       | 3.95E+05 (1.85E+04)       |
| 80  | 5.48E+05 (2.27E+04)       | 5.54E+05 (2.09E+04)       | 5.60E+05 (3.06E+04)       |
| 100 | 6.67E+05 (2.19E+04)       | 6.73E+05 (3.13E+04)       | 6.64E+05 (2.89E+04)       |
| 120 | 8.22E+05 (3.68E+04)       | 8.31E+05 (3.10E+04)       | 8.25E+05 (2.42E+04)       |
| 140 | 9.79E+05 (3.75E+04)       | 9.87E+05 (4.27E+04)       | 9.84E+05 (4.36E+04)       |
| 160 | 1.12E+06 (4.12E+04)       | 1.12E+06 (3.38E+04)       | 1.12E+06 (3.70E+04)       |
| 180 | 1.24E+06 (3.34E+04)       | 1.24E+06 (3.60E+04)       | 1.24E+06 (4.12E+04)       |
| 200 | 1.39E+06 (3.91E+04)       | 1.60E+06 (9.68E+05)       | 1.89E+06 (1.23E+06)       |

**FIGURE 4.** Evolution of the mean EC obtained by GTPA-VP, GAC-GA, DEC-GA, and Kmeans-GA on nine instances over 20 runs.

C. EFFECTIVENESS OF THE ASSOCIATION BETWEEN UAVs AND SPs AND THE ORDER OF SPs

GTPA-VP adopted an MCGA to jointly handle the association between UAVs and SPs and the order of SPs for UAVs. In order to show the effectiveness of MCGA used in GTPA-VP, we designed algorithms called GAC-GA, DEC-GA, and Kmeans-GA. GAC-GA, DEC-GA, and Kmeans-GA use GAC [40], DE clustering (DEC) [40], and K-means clustering [39], respectively for associating SPs with UAVs. All the above-mentioned algorithms use GA [42] for constructing the order of SPs. The deployment of SPs was kept the same for all of them as used in GTPA-VP. The number of iterations and population size were set to 50 and 10, respectively in DEC and GAC.

The mean EC and Std Dev of GTPA-VP, GAC-GA, DEC-GA, and Kmeans-GA are presented in Table 4. In addition, Figure 4 presents the evolution of the mean EC by GTPA-VP, GAC-GA, DEC-GA, and Kmeans-GA on ten instances.
One can see from Table 4 and Figure 4 that the proposed GTPA-VP performs better than GAC-GA, DEC-GA, and Kmeans-GA in terms of mean EC. In addition, the statistical test results of GTPA-VP, GAC-GA, DEC-GA, and Kmeans-GA are summarized at the bottom of Table 4, which shows that GTPA-VP is significantly better than GAC-GA, DEC-GA, and Kmeans-GA. The superiority of GTPA-VP against compared algorithms can be attributed as: in GTPA-VP, MCGA (which is known for its good convergence) is used to jointly address the association between UAVs and SPs and the order of SPs for UAVs, which may lead to good performance, while in GAC-GA, DEC-GA, and Kmeans-GA, they are addressed independently.

### V. CONCLUSION

In this paper, a multi-UAV-assisted MEC system has been studied, where multiple UAVs have been used to serve IoTDs. We aimed to optimize the sum of hovering and flying energies of UAVs in the system. The problem is complicated to be solved by traditional optimization methods. We have proposed a genetic trajectory planning algorithm with variable population size called GTPA-VP, which consists of two phases. In the first phase, a GA with a variable population size called GTPA-VP, consists of two evolved algorithms: insert, replace, and delete. Accordingly, MCGA is adopted to associate SPs with UAVs, predict an optimal number of UAVs, and construct the optimal order of SPs for UAVs, in order to minimize the flying distances (i.e. flying energies) of UAVs as well as improved the mean EC of the system. The experimental results on ten instances up to 200 IoTDs have shown that GTPA-VP performs better than other compared variants in terms of energy consumption. In the future, we intend to improve the complexity of the proposed algorithm keeping in view the current industrial applications and demands of MEC systems across the globe.

### ACKNOWLEDGMENT

All authors are thankful to Qatar National Library (QNL), Qatar, for support to conduct study in this paper.

### REFERENCES

[1] M. Asim, Y. Wang, K. Wang, and P.-Q. Huang, “A review on computational intelligence techniques in cloud and edge computing,” *IEEE Trans. Emerg. Topics Comput. Infell.,*, vol. 4, no. 6, pp. 742–763, Dec. 2020.
[2] A. Guo, Y. Hu, W. Liang, Y. Lin, L. Li, and X. Li, “A QoE-oriented scheduling scheme for energy-efficient computation offloading in UAV cloud system,” *IEEE Access*, vol. 7, pp. 68656–68668, 2019.
[3] A. H. Zaini and L. Xie, “Distributed drone traffic coordination using triggered communication,” *Unmanned Syst.*, vol. 8, no. 1, p. 1–20, Jan. 2020.
[4] M. Mozaffari, W. Saad, M. Bennis, Y.-H. Nam, and M. Debbah, “A tutorial on UAVs for wireless networks: Applications, challenges, and open problems,” *IEEE Commun. Surveys Tuts.*, vol. 21, no. 3, pp. 2334–2360, 3rd Quart., 2019.
[5] J. E. Low, L. T. S. Win, D. S. B. Sahul, C. H. Tan, G. S. Soh, and S. Fong, “Design and dynamic analysis of a transformable hovering rotorcraft (THOR),” in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2017, pp. 6389–6396.
[6] Y. Zeng, R. Zhang, and T. J. Lim, “Wireless communications with unmanned aerial vehicles: Opportunities and challenges,” *IEEE Commun. Mag.*, vol. 54, no. 5, pp. 36–42, May 2016.
[7] P.-M. Olsson, J. Kvarnstrom, P. Doherty, O. Burdakov, and K. Holmberg, “Generating UAV communication networks for monitoring and surveillance,” in *Proc. 11th Int. Conf. Control Autom. Robot. Yuc.*, Dec. 2010, pp. 1070–1077.
[8] C. Yuan, K. A. Gharny, Z. Liu, and Y. Zhang, “Unmanned aerial vehicle based forest fire monitoring and detection using image processing technique,” in *Proc. IEEE Chin. Guid., Navigat. Control Conf. (CGNCC)*, Aug. 2016, pp. 1870–1875.
[9] M. Ayaz, M. Ammad-Uddin, Z. Sharif, A. Mansour, and E. M. Aggoune, “Internet-of-Things (IoT)-based smart agriculture: Toward making the fields talk,” *IEEE Access*, vol. 7, pp. 129551–129583, 2019.
[10] R. Gupta, A. Shukla, P. Mehta, S. Batra, N. Kumar, “VHAKA: A blockchain-based outdoor delivery scheme using UAV for healthcare 4.0 services,” in *Proc. IEEE Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, Jul. 2020, pp. 255–260.
[11] K. Gomez, A. Hourani, L. Goratti, R. Riggio, S. Kandeepan, and I. Bucaille, “Capacity evaluation of aerial LTE base-stations for public safety communications,” in *Proc. Eur. Conf. Netw. Commun. Comput. Commun. (EurCN)*, Jun. 2015, pp. 133–138.
[12] A. Murwaday and I. Guvenc, “UAV assisted heterogeneous networks for public safety communications,” in *Proc. IEEE Wireless Commun. Netw. Conf. Workshops (WCNCW)*, Mar. 2015, pp. 329–334.
[13] Y. Wang, Z.-Y. Ru, K. Wang, and P.-Q. Huang, “Joint deployment and task scheduling optimization for large-scale mobile users in multi-UAV-enabled mobile edge computing,” *IEEE Trans. Cybern.*, vol. 50, no. 9, pp. 3984–3997, Sep. 2020.
[14] B. Zhang, G. Zhang, S. Ma, K. Yang, and K. Wang, “Efficient multistage scheduling for completion time minimization in UAV-assisted mobile edge computing,” *Mobile Inf. Syst.*, vol. 2020, pp. 1–11, Jun. 2020.
[15] S. Garg, A. Singh, S. Batra, N. Kumar, and L. T. Yang, “UAV-empowered edge computing environment for cyber-threat detection in smart vehicles,” *IEEE Netw.*, vol. 32, no. 3, pp. 42–51, May/Jun. 2018.
[16] L. Zhang, Z. Zhao, Q. Wu, H. Zhao, H. Xu, and X. Wu, “Energy-aware dynamic resource allocation in UAV assisted mobile edge computing over social internet of vehicles,” *IEEE Access*, vol. 6, pp. 56700–56715, 2018.
[17] Y. Du, K. Yang, K. Wang, G. Zhang, Y. Zhao, and D. Chen, “Joint resources and workflow scheduling in UAV-enabled wirelessly-powered MEC for IoT systems,” *IEEE Trans. Veh. Technol.*, vol. 68, no. 10, pp. 10187–10200, Oct. 2019.
[18] L. Zhang, A. Celik, S. Dang, and B. Shihada, “Energy-efficient trajectory optimization for UAV-assisted IoT networks,” IEEE Trans. Mobile Comput., early access, Apr. 22, 2021, doi: 10.1109/TMC.2021.3075083.

[19] L. Wang, K. Wang, C. Pan, W. Xu, N. Aslam, and L. Hanzo, “Multi-agent deep reinforcement learning-based trajectory planning for multi-UAV assisted mobile edge computing,” IEEE Trans. Cogn. Commun. Netw., vol. 7, no. 1, pp. 73–84, Mar. 2021.

[20] Y. Liu, K. Xiong, Q. Ni, P. Fan, and K. B. Letaief, “FAIR data allocation,” IEEE Internet Things J., vol. 7, no. 4, pp. 2777–2790, Apr. 2020.

[21] Y. K. Tun, Y. M. Park, N. H. Tran, W. Saad, S. R. Pandey, and C. S. Hong, “Energy-efficient resource management in UAV-assisted mobile edge computing,” IEEE Access, vol. 12, pp. 249–253, Jan. 2021.

[22] Q. Wu and R. Zhang, “Common throughput maximization in UAV-enabled OFDMA systems with delay consideration,” IEEE Trans. Commun., vol. 66, no. 12, pp. 6614–6627, Dec. 2018.

[23] X. Xiao, J. Zheng, Y. Cai, Y. Wu, and A. Anpalagan, “Fair data allocation and trajectory optimization for UAV-assisted mobile edge computing,” IEEE Commun. Lett., vol. 23, no. 12, pp. 2357–2361, Dec. 2019.

[24] S. Jeong, O. Simeone, and J. Kang, “Tireless edge computing via a UAV-mounted cloudlet: Optimization of bit allocation and path planning,” IEEE Trans. Veh. Technol., vol. 67, no. 3, pp. 2049–2063, Mar. 2018.

[25] X. Hu, K. K. Wong, K. Yang, and Z. Zheng, “UAV-assisted relaying and edge computing: Scheduling and trajectory optimization,” IEEE Trans. Wireless Commun., vol. 18, no. 10, pp. 4738–4752, Oct. 2019.

[26] Z. Yang, C. Pan, K. Wang, and M. Shikh-Bahaei, “Energy efficient resource allocation in UAV-enabled mobile edge computing networks,” IEEE Trans. Wireless Commun., vol. 18, no. 9, pp. 4576–4589, Sep. 2019.

[27] J. Zhang, L. Zhou, F. Zhou, B.-C. Seet, H. Zhang, Z. Cai, and J. Wei, “Communication-efficient offloading and trajectory scheduling for multi-UAV assisted mobile edge computing,” IEEE Trans. Veh. Technol., vol. 69, no. 2, pp. 2114–2125, Feb. 2020.

[28] P.-Q. Huang, Y. Wang, and K.-Z. Wang, “Energy-efficient trajectory planning for a multi-UAV-assisted mobile edge computing system,” Frontiers Inf. Technol. Electron. Eng., vol. 21, no. 1, pp. 1713–1725, Dec. 2020.

[29] Y. Zeng, J. Xu, and R. Zhang, “Energy minimization for wireless communication with rotary-wing UAV,” IEEE Trans. Wireless Commun., vol. 18, no. 4, pp. 2329–2345, Apr. 2019.

[30] M. Seyedali. (2020). The Continuous Version of the Genetic Algorithm. [Online]. Available: https://www.mathworks.com/matlabcentral/fileexchange/867694-the-continuous-version-of-the-genetic-algorithm

[31] T. Liao, K. Socha, M. A. M. de Oca, T. Stützle, and M. Dorigo, “Ant colony optimization for mixed-variable optimization problems,” IEEE Trans. Evol. Comput., vol. 18, no. 4, pp. 503–518, Aug. 2014.

[32] K. P. Sinaga and M.-S. Yang, “Unsupervised K-means clustering algorithm,” IEEE Access, vol. 8, pp. 80716–80727, 2020.

[33] M. Ryerkerk, R. Averill, K. Deb, and E. Goodman, “A survey of evolutionary algorithms using metameric representations,” Genet. Program. Evolvable Mach., vol. 20, no. 4, pp. 494–506, Dec. 2019.

[34] K. Andráš, (2021). Multiple Traveling Salesmen Problem—Genetic Algorithm, Using Multi-Chromosome Representation, MATLAB Central File Exchange. [Online]. Available: https://www.mathworks.com/matlabcentral/fileexchange/48133-multiple-traveling-salesmen-problem-genetic-algorithm-using-multi-chromosome-representation

[35] A. Kiruly and J. Abonyi, “Redesign of the supply of mobile mechanics based on a novel genetic optimization algorithm using Google maps API,” Eng. Appl. Artif. Intell., vol. 38, pp. 122–130, Feb. 2015.

[36] M. Asim, W. Khan, O. Yeniay, M. A. Jan, N. Tairan, H. Hussian, and G.-G. Wang, “Hybrid genetic algorithms for global optimization problems,” Hacettepe J. Math. Statist., vol. 47, no. 3, pp. 539–551, 2018.

[37] A. E. Eiben and J. E. Smith, Introduction to Evolutionary Computing, 2nd ed. Springer, 2015.

[38] A. Kiruly and J. Abonyi, “Optimization of multiple traveling salesmen problem by a novel representation based genetic algorithm,” in Intelligent Computational Optimization in Engineering. Berlin, Germany: Springer, 2011, pp. 241–269.

[39] A. K. Jain, “Data clustering: 50 years beyond K-means,” Pattern Recognit. Lett., vol. 31, no. 8, pp. 651–666, 2010.

[40] M. Yann. (2019). Distance-Based Clustering of a Set of XY Coordinates. [Online]. Available: https://www.mathworks.com/matlabcentral/fileexchange/56150-distance-based-clustering-of-a-set-of-xy-coordinates

[41] P. Huang, Y. Wang, K. Wang, and K. Yang, “Differential evolution with a variable population size for deployment optimization in a UAV-assisted IoT data collection system,” IEEE Trans. Emerg. Topics Comput. Intell., vol. 4, no. 3, pp. 324–335, Jun. 2020.

[42] K. Joseph. (2021). Traveling Salesman Problem—Genetic Algorithm, MATLAB Central File Exchange. [Online]. Available: https://www.mathworks.com/matlabcentral/fileexchange/13680-traveling-salesman-problem-genetic-algorithm

MUHAMMAD ASIM received the M.S. degree in mathematics from the University of Peshawar, Khyber Pakhtunkhwa, Pakistan, in 1996, and the Ph.D. degree in mathematics from the University of Essex, U.K., in 2012. He is currently pursuing the Ph.D. degree in computer science and technology with Central South University, Changsha, China. His current research interests include computational intelligence techniques, cloud computing, and edge computing.

WALI KHAN MASHWANI received the M.Sc. degree in mathematics from the University of Peshawar, Khyber Pakhtunkhwa, Pakistan, in 2000, and the Ph.D. degree in mathematics from the Federal Polytechnic School of Lausanne, EPFL, Switzerland, in 2006. He is currently an Associate Professor with Qatar Foundation, Division of Information and Communication Technologies, College of Science and Engineering, Hamad Bin Khalifa University (HBKU). During last years, he also holds several research and teaching positions at Innopolis University, Russia; Alfaisal University, Saudi Arabia; the University of Sharjah, United Arab Emirates; Universiti Teknologi PETRONAS, Malaysia; and EPFL Federal Swiss School. His research interests include evolutionary computation, hybrid evolutionary multi-objective algorithms, and decomposition-based evolutionary methods for multi-objective optimization, mathematical programming, numerical analysis, and artificial neural networks.

SAMIR BRAHIM BELHAOUARI (Senior Member, IEEE) received the master’s degree in telecommunications and network from the Institut Nationale Polytechnique of Toulouse, France, in 2000, and the Ph.D. degree in mathematics from the Federal Polytechnic School of Lausanne, EPFL, Switzerland, in 2006. He is currently an Associate Professor with Qatar Foundation, Division of Information and Communication Technologies, College of Science and Engineering, Hamad Bin Khalifa University (HBKU). During last years, he also holds several research and teaching positions at Innopolis University, Russia; Alfaisal University, Saudi Arabia; the University of Sharjah, United Arab Emirates; Universiti Teknologi PETRONAS, Malaysia; and EPFL Federal Swiss School. His research interests include various areas from applied mathematics, statistics, and data analysis until artificial intelligence, and image signal processing due to both mathematics and computer science backgrounds.

SAIMA HASSAN received the M.Sc. degree in computer science from the University of Peshawar, Pakistan, in 2003, and the M.Ph. and Ph.D. degrees in information technology from Universiti Teknologi PETRONAS, Malaysia, in 2013 and 2016, respectively. She is currently an Assistant Professor with the Institute of Information Technology, Kohat University of Science & Technology, Kohat, Pakistan. Her research interests include time series forecasting, artificial neural networks, and application of computational intelligence techniques to load forecasting.

VOLUME 9, 2021