An Evolutionary Trajectory Planning Algorithm for Multi-UAV-Assisted MEC System

Muhammad Asim, Wali Khan Mashwani, and Samir Brahim Belhaouari

***Abstract*—This paper presents a multi-unmanned aerial vehicle (UAV)-assisted mobile edge computing (MEC) system, where multiple UAVs are used to serve mobile users (MUs). We aim to minimize the overall energy consumption of the system by planning the trajectories of UAVs. To plan the trajectories of UAVs, we need to consider the deployment of hovering points (HPs) of UAVs, their association with UAVs, and their order for each UAV. Therefore, the problem is very complicated, as it is non-convex, nonlinear, NP-hard, and mixed-integer. To solve the problem, this paper proposed an evolutionary trajectory planning algorithm (ETPA), which comprises three phases. In the first phase, variable-length GA is adopted to update the deployments of HPs for UAVs. Accordingly, redundant HPs are removed by the remove operator. Subsequently, differential evolution clustering is adopted to cluster HPs into different clusters without knowing the number of HPs in advance. Finally, a GA is proposed to construct the order of HPs for UAVs. The experimental results on a set of eight instances show that the proposed ETPA outperforms other compared 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 resource-intensive and latency-sensitive applications are emerging, such as virtual reality, online gaming, and so on. Such applications are usually sensitive to latency and require huge computational resources. however, due to limitations on mobile users (MUs) devices, 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 near or at MUs. It can execute tasks of MUs at the nearby edge cloud and sends back the results to MUs [1]. Due to the shorter physical distance between MEC's server/edge cloud and MUs, it consumes less energy as compared to mobile cloud computing. However, it is still lacking in fulfilling the requirements of MUs, as the location of the edge cloud is usually fixed and cannot be adjusted flexibly according to the requirements of MUs. Therefore, it cannot provide timely services during a natural disaster as the terrestrial communication link may be broken/lost.

To satisfy this ever-increasing demand, unmanned aerial vehicle (UAV) is regarded as one of the most promising technologies to achieve these ambitious goals. Compared to the traditional communication systems that utilize the terrestrial fixed base stations, UAV-aided communication 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 system performance (e.g., data rate and latency) can be significantly enhanced by establishing the line-of-sight communication links between UAVs and MUs. In addition, through dynamically adjusting the flying and hovering location, UAVs are capable of improving communication performance in wireless communications.

Recently, due to the above-mentioned advantages, UAVs have been extensively used in various fields, such as wireless communication [2] [3], military [4] [5], surveillance and monitoring [6] [7], delivery of medical supplies [8], and rescue operations [9] [10]. Very recently, UAVs have been used to enhance the capabilities of MEC systems. For example, Wang *et al.* [11] studied a multi-UAV-enabled MEC system, where several UAVs are deployed as flying edge clouds for large-scale MUs. Zhang *et al.* [12] proposed a UAV-assisted MEC for efficient multitask scheduling to minimize completion time. Garg *et al.* [13] studied the application of a UAV-empowered MEC system in cyber-threat detection of smart vehicles.

Moreover, to fully exploit the potential of UAV-assisted MEC systems, some researchers have studied appropriate path planning and trajectory designing of UAVs. For instance, Wang *et al.* [14] proposed 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 MUs. Wu *et al.* [15] studied 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 MUs. Diao *et al.* [16] optimized joint trajectory and data allocation to minimize the maximum energy consumption. Jeong *et al.* [17] studied the bit allocation and trajectory planning under latency and energy budget constraints. Hu *et al.* [18] 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

M. Asim is with the School of Computer Science and Engineering, Central South University, Changsha 410083, China (Email: asimpk@csu.edu.cn)

W.K. Mashwani is with Institute of Numerical Sciences, Kohat University of Science & Technology, Pakistan (Email:mashwanigr8@gmail.com)

Samir Brahim Belhaouari is with College of Science and Engineering, Hamad Bin Khalifa University, Qatar (Email:sbelhaouari@hbku.edu.qa)

of UAVs and MUs subject to task constraints. Yang *et al.* [19] presented the sum power minimization problem for a UAV-enabled MEC network. Huang *et al.* [20] studied multi-UAV-assisted MEC system, where the UAVs act as edge servers to provide computing services for Internet of Things devices. Zeng *et al.* [21] 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, it is clear that variable numbers of UAVs have rarely been considered in the current studies. The deployment of an appropriate number of UAVs can improve the system's performance. The main contributions of this paper are summarized as follows:

- A new multi-UAV-assisted MEC system is proposed and formulated to minimize the energy consumption of the system by considering the deployment including the number and locations of hovering points (HPs), the number of UAVs, and their association with HPs, and the order of HPs.
- The deployment of HPs is addressed by proposing a genetic algorithm (GA) with a variable length individual. Specifically, evolutionary operators like crossover and mutation are modified to handle variable-length individuals.
- An evolutionary trajectory planning algorithm (ETPA) is proposed, that consists of four phases. First, a variable-length GA (VLGA) [22] is adopted to optimize the deployment of HPs. Subsequently, redundant HPs which have no MUs to be served, are removed by using the remove operator. After that, UAVs are associated with HPs via differential evolution clustering (DEC) algorithm [23]. Accordingly, a GA is adopted to construct the order of HPs for UAVs.
- Extensive experiments have been carried out on a set of ten instances with up to 200 MUs. The experimental results show the effectiveness of the proposed ETPA.

The remainder of this paper is organized as follows. In Section II, we introduce the system model, including the problem formulation of the proposed system. Section III presents the details of our proposed algorithm ETPA. In Section IV, the experimental studies are discussed. Finally, Section V concludes this paper.

II. System Model

As shown in Fig. 1, we consider there are $i \in \mathcal{N} = \{1, 2, ..., N\}$ MUs and $j \in \mathcal{M} = \{1, 2, ..., M\}$ UAV. UAV flies over all the MUs to collect the data. We assume that the UAV will hover at some points for some time and the MU can send the sensing data to the UAV. We assume UAV will hover over $t \in \mathcal{T}_j = \{1, 2, ..., T_j\}$ HPs. Therefore, one has

$$a_{ij}[t] = \{0, 1\}, \forall i \in \mathcal{N}, \forall t \in \mathcal{T}_j, \forall j \in \mathcal{M}, \tag{1}$$

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

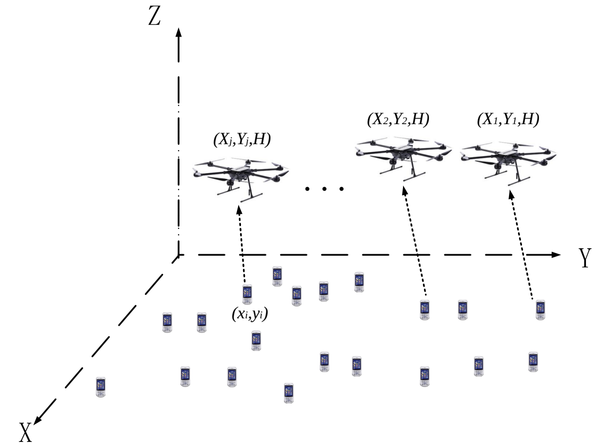


Fig. 1: Collection Framework of multi-UAV-assisted MEC system

$$\sum_{t=1}^{T_j}\sum_{j=1}^{M} a_{ij}[t] = 1, i \in \mathcal{N} \tag{2}$$

which denotes that one MU should choose one UAV at each HP to send its sensing data.

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

$$a_{ij}[t] = \begin{cases} 1, & if(i,j,t) = \underset{i \in \mathcal{N}, j \in \mathcal{M}}{\text{argmin}}\ (d_{ijt}), \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

Assume that at each HP, j-th UAV can accept at most U_j MUs, Therefore, one has

$$\sum_{i=1}^{N} a_{ij}[t] \leq U_j, t \in \mathcal{T}_j, j \in \mathcal{M} \tag{4}$$

We assume that i-th MU may collect D_i amount of data which intend to send it to the UAV. The UAV may stop at T points at the air in which each stop may last for T^{max} seconds, where T^{max} is the fixed value.

Then, the time to send the data from MU to UAV at the t-th HP is as

$$T_i^{Tr}[t] = \frac{D_i}{r_{ij}[t]}, \ \ \forall j \in \mathcal{M}, t \in \mathcal{T}_j \tag{5}$$

where $r_{ij}[t]$ is the data rate which is given by (14). Also, define F_i as the CPU cycles which this task may need to process. Then, one can have the process time of the data in UAV as

$$T_i^{C}[t] = \frac{F_i}{f_{ij}[t]}, \ \ \forall j \in \mathcal{M}, \forall t \in \mathcal{T}_j \tag{6}$$

where $f_{ij}[t]$ is the computation capacity of the UAV assigned to each data processing procedure, where we have

$$\sum_{i=1}^{N} f_{ij}[t] \leq f_{max}, j \in \mathcal{M}, t \in \mathcal{T}_j \tag{7}$$

which f_{max} is the maximal computing power the UAV can provide to each MU. Also, we have

$$T_i[t] = T_i^C[t] + T_i^{Tr}[t], i \in \mathcal{N}, t \in \mathcal{T} \quad (8)$$

Then, one can have

$$T_j[t]^H = \max_{i \in \mathcal{N}} \{T_i^C[t] + T_i^{Tr}[t], j \in \mathcal{M}, t \in \mathcal{T}\}, \quad (9)$$

Assume that the coordinate of i-th MU is as (x_i, y_i) and the coordinate of the j-th UAV at t-th HP is as $(X_j[t], Y_j[t], H)$. Also, assume the UAV's trajectory can be characterized by a sequence of location $q_j[t] = [X_j[t], Y_j[t], H]^T$, where H is a fixed value. In addition, all UAVs start from the same initial position $q[0]$ and finally come back to the same initial position $q[0]$ after visiting all the HPs. Also, we have

$$||q_j[t+1] - q_j[t]||^2 \leq S_{max}^2, t = 0, ..., T_j, \quad (10)$$

where $S_{max} = V_{max} \cdot T^{max}$ is the maximum horizontal distance which the UAV can travel and V_{max} is the maximum speed.

Then, the horizontal distance between the i-th MU and the UAV is as

$$R_{ij}[t] = \sqrt{(X_j[t] - x_i)^2 + (Y_j[t] - y_i)^2}, \forall i \in \mathcal{N}, \forall t \in \mathcal{N} \quad (11)$$

Also, the distance between the i-th MU and the UAV at the t-th HP is as

$$d_{ij}[t] = \sqrt{R_{ij}[t]^2 + H^2}, \forall i \in \mathcal{N}, \forall t \in \mathcal{N} \quad (12)$$

Then, the channel power gain can be given as

$$h_{ij}[t] = \frac{\beta_0}{d_{ij}[t]^2} \quad (13)$$

where β_0 denotes the channel power gain at the reference distance $1m$.

If MUs decide to offload to the UAVs, the data rate can be given as

$$r_{ij}[t] = B\log_2\left(1 + \frac{p_i{}^{ue} h_{ij}[t]}{\sigma^2}\right) \quad (14)$$

where σ^2 is the noise power and p_i^{ue} is the transmission power, which is constrained by

$$p_i{}^{ue} \leq P^{max} \quad (15)$$

The energy consumption of the i-th MU for sending data to the j-th UAV at t-th HP is given by

$$E_{ij}^{Tr}[t] = p_i^{ue} T_i^{Tr}[t] = \frac{p_i^{ue} D_i}{r_{ij}[t]}, \ \forall j \in \mathcal{M}, t \in \mathcal{T}_j \quad (16)$$

The whole energy consumption of all MUs is expressed as

$$E_{MU} = \sum_{i=1}^{N}\sum_{j=1}^{M}\sum_{t=1}^{T_j} a_{ij}[t] E_{ij}^{Tr}[t] \quad (17)$$

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

$$E_j^F = \frac{P^F}{V} \sum_{t=1}^{T_j - 1} ||q_j[t+1] - q_j[t]||^2 \quad (18)$$

where V is the velocity of UAVs.

Also, for the hovering energy, one can have

$$E_j^H = P^H \sum_{t=1}^{T_j - 1} T_j^H[t], \quad (19)$$

where P^H denotes the hovering power of the UAV.

The whole energy consumption of all UAVs is expressed as

$$E_{UAV} = \sum_{j=1}^{M} (E_j^F + E_j^H + C) \quad (20)$$

where C is the fixed cost including take off, land in, and maintenance cost for adding UAVs.

Then, we can have the optimization problem as follows.

$$\mathcal{P}: \min_{\boldsymbol{a_{ij}[t], T_j, q_j[t], p_i^{ue}, f_{ij}[t], M}} \ (E_{UAV} + \alpha E_{MU}) \quad \text{(21a)}$$

subject to:

$$a_{ij}[t] = \{0, 1\}, \forall i \in \mathcal{N}, \forall t \in \mathcal{T}_j, \forall j \in \mathcal{M}, \quad \text{(21b)}$$

$$\sum_{t=1}^{T_j}\sum_{j=1}^{M} a_{ij}[t] = 1, i \in \mathcal{N}, \quad \text{(21c)}$$

$$\sum_{i=1}^{N} a_{ij}[t] \leq U_j, t \in \mathcal{T}_j, j \in \mathcal{M}, \quad \text{(21d)}$$

$$\sum_{i=1}^{N} f_{ij}[t] \leq f_{max}, j \in \mathcal{M}, t \in \mathcal{T}_j \quad \text{(21e)}$$

$$T_i[t] \leq T^{max}, \quad \text{(21f)}$$

$$||q_j[t+1] - q_j[t]||^2 \leq S_{max}^2, t = 0, ..., N, \quad \text{(21g)}$$

$$p_i{}^{ue} \leq P^{max}, \quad \text{(21h)}$$

$$X_{min} \leq X_j[t] \leq X_{max}, \ \forall j \in \mathcal{M}, t \in \mathcal{T}_j, \quad \text{(21i)}$$

$$Y_{min} \leq Y_j[t] \leq Y_{max}, \ \forall j \in \mathcal{M}, t \in \mathcal{T}_j. \quad \text{(21j)}$$

Where the objective function is the sum of hovering energy and flying energy of UAVs and *C8* and *C9* present the lower and upper bounds of 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 (21(a)) is a non-convex, NP-hard, and nonlinear optimization problem. (21(a)) can not be solved by traditional optimization methods due to the following challenges.

- To solve (21(a)), we need to consider the number of UAVs, the number of HPs and their locations, which MU will send data to which HP, which UAV will visit which HPs, and in which order the UAV will visit the assigned HPs. Therefore, it is a complicated/complex problem to be tackled.
- (21(a)) contains integer decision variable M and the number of HPs T_j for UAV j, binary variable a_{ij}, and continuous variables (X_j and Y_j). Therefore, it is a mixed decision variable problem, which is challenging to be solved [11] [24].

- Since, the number of UAVs is unknown in prior, the clustering of HPs into different clusters requires an unsupervised scheme (i.e., free of initialization/parameter-free clustering algorithm) that can group closely spaced HPs into different clusters automatically and can also simultaneously find an optimal number of clusters/UAVs [25].

In this paper, we proposed an algorithm called ETPA to design the trajectories of UAVs. The proposed algorithm consists of four phases: the deployment of HPs, removing redundant HPs, the association between UAVs and HPs, and the order of HPs for UAVs.

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

- Considering the strong coupling among the deployment of HPs, the association between UAVs and HPs, and the order of HPs. ETPA plans the trajectories of UAVs at each iteration through four phases: updating the deployment of HPs, removing redundant HPs, the association between UAVs and HPs, and constructing the optimal trajectories for UAVs.
- In ETPA, the deployment of HPs is solved by using VLGA in [22]. Each individual represents the whole deployment; thus, the whole population represents a set of deployments. Since the length of individuals is variable, we modified the common crossover and mutation operators to handle variable-length individuals for updating the deployment of HPs.
- The optimization problem (21(a)) includes mixed decision variables i.e., integer, binary and continuous decision variables. By analyzing the problem, we transformed it into subproblems so that there is no mixed variables involved. We solved each subproblem independently by proposing an efficient algorithm.

B. ETPA

The framework of ETPA is given in Algorithm 1. In the initialization, the locations of HPs are produced randomly, forming an initial population POP = (X_1, Y_1), (X_2, Y_2),...,(X_{max}, Y_{max}). Subsequently, redundant HPs are removed to restrict UAVs from visiting HPs having no MU by using the algorithm given in Algorithm 3. Accordingly, DEC algorithm in Algorithm 4 is adopted to group HPs into different clusters and a UAV is assigned to each cluster. Afterward, GA in Algorithm 5 is adopted to construct the order of HPs in each cluster. After that, POP is evaluated via Eq. (21(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_{off} is first produced via VLGA in Algorithm 2 during the evolution. Accordingly, redundant HPs are removed by using the algorithm in Algorithm 3. Then, the HPs in the individuals of POP_{Off} are grouped into different clusters by using DEC algorithm in Algorithm 4. Accordingly, the trajectories of UAVs are constructed via GA in Algorithm 5. Accordingly, the new population POP_{Off} is evaluated using Eq. (21(a)). Finally, we select the best feasible individuals among the individuals of POP and POP_{Off} with the greatest performance improvement. This process continues until $FEs \geq FEs_{max}$.

Algorithm 1 General Framework of ETPA

```
1: FEs = 0;
2: repeat
3: Generate Random population POP;
4: Update HPs by removing HPs with no MU using Algorithm 3;
5: Determine the association between UAVs and HPs in via DEC in Algorithm 4;
6: Construct the order of HPs for each UAV by using GA given in Algorithm 5;
7: Evaluate POP via Eq. (21(a));
8: FEs = FEs + 1;
9: while FEs < FEs_max do
10:    Produce an offspring population POP_Off via GA given in Algorithm 2;
11:    for i=1:|POP_Off| do
12:        Update HPs by removing HPs with no MU using Algorithm 3;
13:        Determine the association between UAVs and HPs in via DEC in Algorithm 4;
14:        Construct the order of HPs for each UAV by using GA given in Algorithm 5;
15:        Evaluate POP_Off via Eq. (21(a));
16:        FEs = FEs + 1;
17:    end for
18:    Select best feasible individuals from POP and POP_Off with the greatest performance improvement;
19: end while
20: Output: the best solution.
```

Algorithm 2 Updating Deployment of HPs

```
1: POP_Off = ∅;
2: for k = 1: 2: |POP| do
3:    x1, x2 ← Apply Tournament selection to select parents;
4:    P ← random probability;
5:    if P ≤ Pc × Pc then
6:        y1, y2 ← Apply SBX crossover on x1, x2;
7:    else if P ≤ 1 − Pc then
8:        y1, y2 ← Apply Single point crossover;
9:    else
10:       y1, y2 ← Apply both SBX and single point crossover;
11:   end if
12:   {O1, O2} ← Apply polynomial mutation on y1, y2;
13:   POP_Off = POP_Off ∪ {O1, O2};
14: end for
15: Output: POP_Off;
```

Algorithm 3 Removing HPs with no MU

```
1: U ← Find unique association between MUs and HPs;
2: D ← Find the set difference between the index set of HPs/POP and U;
3: Updated POP ← Update HPs by removing HPs from POP with indexes D;
```

C. The Deployment of HPs

For the deployment of HPs, a variable-length GA (VLGA) in [22] is adopted. GA is a simple, most popular, and effective

EA and has been successfully applied in many fields [26]. More specifically, different from [22], tournament selection [27], simulated binary crossover (SBX) [28] [29] [30], and polynomial mutation [31] operators were adopted in ETPA to generate an offspring population POP_{off} (i.e., locations of new HPs). The individuals of POP_{off} is adopted to update parent population POP (i.e., locations of HPs can be updated). Thus, locations of HPs can be updated by using the above process.

Since each individual in GA represents a location of HP. Therefore, the whole population represents the locations of all HPs. Hence, the number of HPs is equal to the length of the individual in the population. Thus, the length of individuals is kept variable during evolution while updating the number of HPs i.e., the individual length can be increased, kept unchanged, or reduced. By using Algorithm 2, we construct the offspring population POP_{off}. More specifically, we designed special crossovers operators to handle variable-length individuals.

If the new population was composed of the newly created descendants only, the old population's best individual may be lost. To eliminate this deficiency, a new operator, the so-called elitism was introduced. This operator ensures that the previous population's best individual will get into the new population without any modification, thus the best solution found so far will survive during the whole evolutionary process.

D. Removing Redundant HPs

After association MUs with closest HPs via Eq. 3, we have some redundant HPs which have no MU associated with them. We update the number of HPs by removing redundant HPs that have no MU to be served by using Algorithm 3. First, we find unique association U between MUs and HPs (line 1), then we calculate the set difference D between the index set of HPs/POP (index set of POP = 1 to size(POP)) and U (line 2), and finally remove HPs from the POP with indexes given in D (line 3). By removing redundant HPs, we restrict UAVs from visiting redundant HPs, as a result, the flying energy can be saved. In addition, it can shorten the running time of ETPA.

E. Association between UAVs and HPs

In this section, we group HPs into different clusters, and then a UAV is associated with the HPs of each cluster. However, since the number of UAVs is unknown, therefore we need a clustering algorithm that does not require the number of clusters/UAVs prior. Clustering can be stated as a particular kind of NP-hard grouping optimization problem [32]. Therefore, it can be solved by optimization algorithms and metaheuristics. Specifically, evolutionary algorithms are widely used for solving NP-hard problems, which provide near-optimal solutions to such problems in a reasonable time [33]. Therefore, a large number of EAs for solving clustering problems have been proposed in the past. EAs are based on the optimization of some objective function (i.e., the so-called fitness function) that guides the evolutionary search [33]. ETPA adopted a DEC algorithm in [23] to automatically cluster HPs into different clusters. Specifically, DE/rand/1 and binomial crossover [34], [35] are used to produce offspring.

Like other EAs, it is also based on a fitness function. The fitness function is computed using the Davies-Bouldin index (DBI) [36]. The DBI is a function of the ratio of the sum of within-cluster scatter to between-cluster separation [37]. The scatter within C_i cluster is computed as

$$S_{i,q} = \left(\frac{1}{\|C_i\|} \sum_{t \in C_i} \{ ||x - z_i||_2^q \} \right)^{\frac{1}{q}}, \tag{22}$$

where $S_{i,q}$ is the qth root of the qth moment of the HPs in cluster i with respect to their mean, and is a measure of the dispersion of the HPs in cluster i. Specifically, $S_{i,1}$ is the average Euclidean distance of the vectors in class i to the centroid of class i, z_i is the centroid of C_i , and is defined as

$$z_i = 1/n_i \sum_{x \in C_i} x \tag{23}$$

, and n_i is the cardinality of C_i, i.e., the number of HPs in cluster C_i . The Minkowski distance of order t between cluster C_i and C_j is defined as

$$d_{ij,t} = \left\{ \sum_{s=1}^{p} |z_{is} - z_{js}|^t \right\}^{\frac{1}{t}} = \|z_i - z_j\|_t, \tag{24}$$

The DBI is then defined as

$$DBI = \frac{1}{K} \sum_{i=1}^{K} R_{i,qt}, \tag{25}$$

where

$$R_{i,qt} = \max_{j, j \neq i} \left\{ \frac{S_{i,q} + S_{j,q}}{d_{ij,t}} \right\}, \tag{26}$$

The objective is to minimize the DBI for getting proper clustering of the HPs.

The DEC algorithm is explained in Algorithm 4. First, for each individual in the population POP, a random number j in the range $[j_{min}; j_{max}]$ is generated. This individual is assumed to present the centers of j clusters. For initializing these centers, j HPs are chosen randomly from the set of HPs. These HPs are distributed randomly in the POP. After that, the DBI is calculated by using Eq. (25). Subsequently, the offspring population is generated by using DE operators. Accordingly, the new population is evaluated by using Eq. (25). Population with minimum DBI is selected as a parent population for the next iteration. This process continues until the maximum number of iterations $MaxIter$ is reached. Finally, the best solution with minimum DBI is selected as the best solution, hence the number of clusters with proper clustering is obtained (i.e., C_j clusters are obtained, where j represents the number of clusters).

F. The order of HPs

In this subsection, we design the optimal trajectories for all UAVs. In fact, this problem can be dealt with as a traveling salesmen problem. In ETPA, we proposed GA to construct the optimal order of HPs for all UAVs. GA is a

Algorithm 4 DEC Algorithm

Initialize: Set of HPs POP, $nPop$=10, [j_{min}; j_{max}], $MaxIter$=50, number of HPs taken from POP K=10, andP_c=0.2;
for For each individual in POP **do**
Generate j in the range [j_{min}; j_{max}]
Choose j HPs randomly from POP;
Distributes these j HPs randomly in the individual;
end for
for each individual in POP **do**
Extract j centers stored in it;
cluster each point by assigning it to the cluster corresponding to the closest center;
$F_{POP} \leftarrow$ Compute DBI via Eq. (25);
end for
$BestFitness \leftarrow$ Select best fitness from F_{POP};
$BestSolution \leftarrow$ Best solution from POP having fitness value $BestFitness$;
for iter = 1:$Max - Iter$ **do**
for i=1:nPop **do**
A $\leftarrow$ randperm($nPOP$);
a, b, c $\leftarrow$ select three random numbers from A;
$POP_O \leftarrow$ Apply DE operators to produce offspring population;
for each individual in POP_O **do**
Extract j centers stored in it;
cluster each point by assigning it to the cluster corresponding to the closest center;
$F_O \leftarrow$ Compute DBI via Eq. (25);
end for
if $F_O < BestFitness$ **then**
$POP \leftarrow POP_O$;
$BestFitness \leftarrow F_O$;
$BestSolution \leftarrow$ Best solution from POP_O having fitness F_O;
end if
end for
end for
Output: C_j clusters and associate a UAV with each cluster ($\forall j \in \mathcal{M}$);

popular EA that ensures good convergence in solving traveling salesman problem [38]. Specifically, Swap, Flip, and Slide operators are used in GA to produce offspring populations. The implemented operators are given below.

- Swap: selects two HPs and swaps them. Selected HPs can belong to the same or different routes.
- Flip/Inversion: selects a sub-route and reverses the visiting order of the HPs/UAVs belonging to it.
- Slide/Insertion: selects an HP and inserts it in another place. The route where it is inserted is selected randomly. It is possible to create a new itinerary with this single customer, with probability.

It can be seen from Algorithm 5, the algorithm requires two input sets, the coordinates of the locations of HPs, and the distance matrix which contains traveling distances among HPs. Furthermore, it requires some parameter determination, like population size, maximum iteration number, and some additional constraints. After these steps, the initial population can be created, which consists of randomly created individuals. The fitness function simply summarizes the overall route lengths for each UAV inside an individual. The selection is tournament selection, where tournament size i.e. the number

Algorithm 5 Local Optimization Algorithm for UAV Route Optimization

Initialize: Cluster C_j, $POPSize$, $MaxIter$, Distance Matrix $Dmat$, and $maxtour$;
for iter = 1:$Max - Iter$ **do**
n=$Size(T, 1)$;
for p = 1:$POPSize$ **do**
d = 0;
d = d + $Dmat(1, End)$;
for k = 2:n **do**
d = d + $Dmat(POP(p, k-1), POP(p, k))$;
end for
$d(p)$ = d;
end for
$MinDist$ = min($d(p)$);
$POP_{New} \leftarrow$ Generate New POP by using GA operators i.e., flip, swap, and slide given in Algorithm 6;
end for

Algorithm 6 GA Operator with flip, slide, and swap

for p = 8:8:pop-size **do**
Select 8 individuals from POP;
for k = 1:8 **do**
Flip $\leftarrow$ Apply Flip to flip 2 HPs;
Swap $\leftarrow$ Apply Swap to transpose HPs from two random individuals;
Slide $\leftarrow$ Apply Slide operator to slide the HPs of random individual;
end for
end for
OUTPUT: NEW POP POP_N

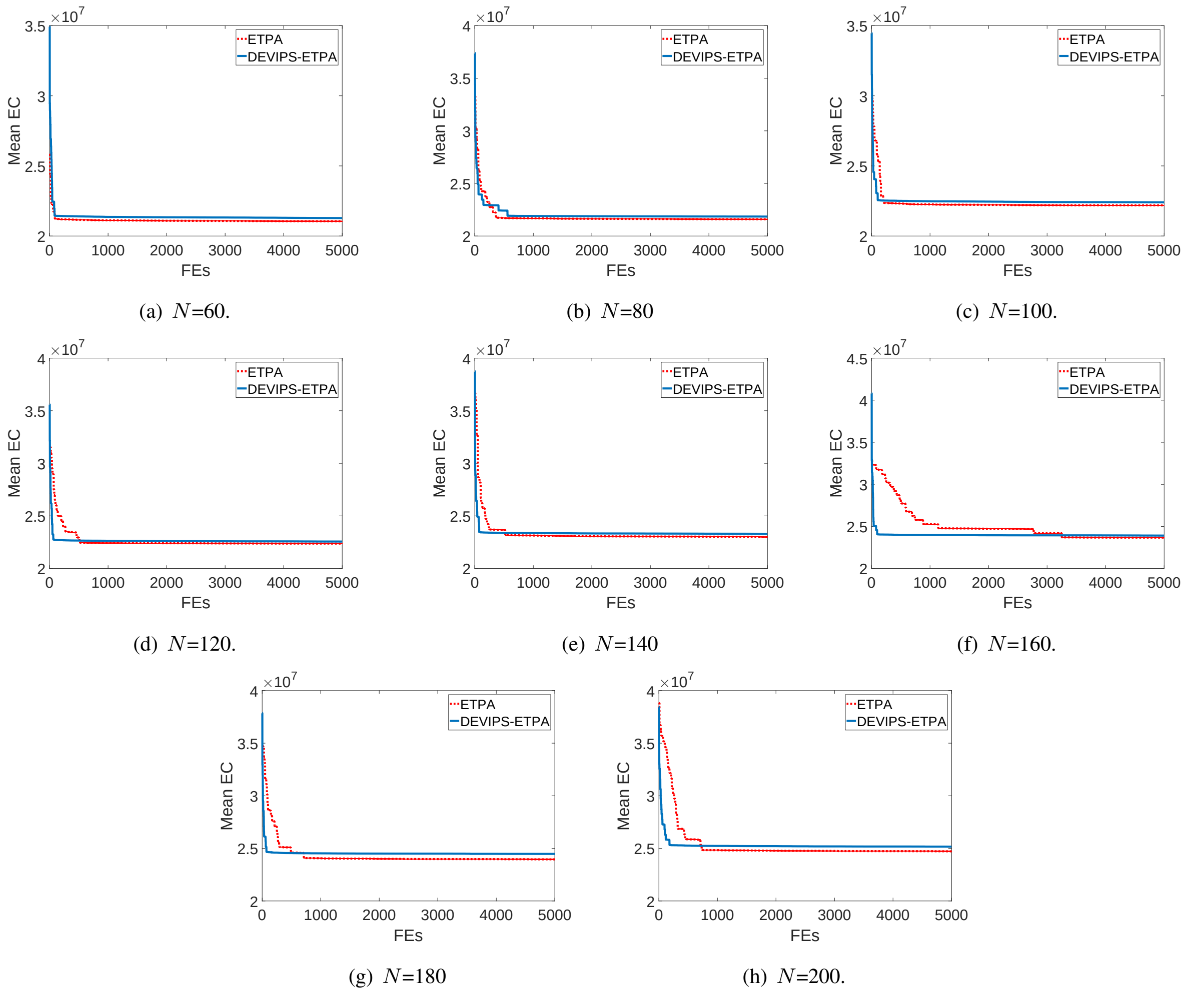
of individuals who compete for survival is 8. Therefore population size must be divisible by 8. The winner of the tournament is the member with the smallest fitness, this individual is selected for a new individual creation, and this member will get into the new population without any modification. After selecting parents from the population, GA's operators i.e., Swap, Flip, and Slide are applied to produce offspring population. The population with minimum tour (i.e., minimum distance) is selected as a parent population for the next iteration. Finally, the best routes/solutions are obtained for UAVs.

IV. Simulation Results

1) Experimental Settings: The parameter setting of the proposed multi-UAV-assisted MEC system is presented in Table I. We have tested ten instances with up to 200 MUs to evaluate the performance of ETPA. We assumed that all the MUs are distributed randomly in a 1000 m $\times$ 1000 m square

TABLE I: Parameters Setting

Parameter	Value	Parameter	Value
$D_i; (i \in M)$	$[1, 10^3]MB$	P	0.1 W
P^H	1000	V_{max}	20 m/s
P^F	1000	σ^2	-174 dBm
B	1 MHz	α	10000
β_0	2.8	H^U	200
X_{max}	1000	Y_{max}	1000



(a) N=60. (b) N=80 (c) N=100.

(d) N=120. (e) N=140 (f) N=160.

(g) N=180 (h) N=200.

Fig. 2: Evolution of the mean EC obtained by ETPA and DEVIPS-ETPA on eight instances over 20 runs.

TABLE II: Experimental results of ETPA and DEVIPS-ETPA in terms of mean EC over 20 runs

N	ETPA	DEVIPS-ETPA
60	**2.11E+07** (3.61E+04)	2.13E+07 (9.17E+04) ↑
80	**2.16E+07** (4.57E+04)	2.18E+07 (9.08E+04) ↑
100	**2.22E+07** (9.28E+04)	2.24E+07 (9.20E+04) ↑
120	**2.24E+07** (5.69E+04)	2.26E+07 (9.50E+04) ↑
140	**2.30E+07** (7.88E+04)	2.33E+07 (8.17E+04) ↑
160	**2.37E+07** (8.38E+04)	2.39E+07 (1.09E+05) ↑
180	**2.40E+07** (9.74E+04)	2.45E+07 (1.02E+05) ↑
200	**2.47E+07** (1.26E+05)	2.52E+07 (1.30E+05) ↑
↑/↓/≈		8/0/0

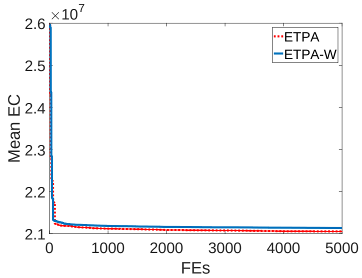
TABLE III: Experimental results of ETPA and ETPA-W in terms of mean EC over 20 runs

N	ETPA	ETPA-W
60	**2.11E+07** (3.61E+04)	2.12E+07 (8.54E+04) ↑
80	**2.16E+07** (4.57E+04)	2.18E+07 (1.21E+05) ↑
100	**2.22E+07** (9.28E+04)	2.24E+07 (1.61E+05) ↑
120	**2.24E+07** (5.69E+04)	2.27E+07 (1.60E+05) ↑
140	**2.30E+07** (7.88E+04)	2.37E+07 (1.71E+05) ↑
160	**2.37E+07** (8.38E+04)	2.44E+07 (2.79E+05) ↑
180	**2.40E+07** (9.74E+04)	2.48E+07 (1.82E+05) ↑
200	**2.47E+07** (1.26E+05)	2.56E+07 (2.43E+05) ↑
↑/↓/≈		8/0/0

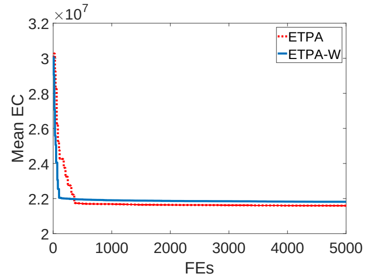
region. The maximum number of fitness evaluations (FEs_{max}) is set to 5000 and 20 runs are implemented independently on each algorithm. The mean energy consumption and the standard deviation of the proposed system over 20 runs are denoted by mean EC and Std, respectively. Furthermore, we performed the Wilcoxon rank-sum test at 0.05 significant level. In the experimental results, we used ↑, ↓, and ≊ to show that ETPA performs significantly better than, worse than, and similar to its competitors.

A. Effectiveness of The Deployment of HPs

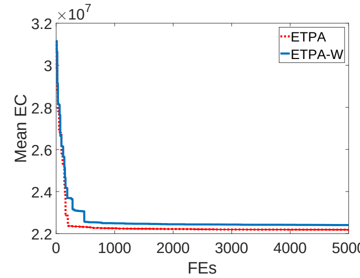
The deployment of HPs is addressed by proposing a GA with variable-length individuals. To prove its effectiveness, we replaced the proposed GA in ETPA with DEVIPS [35] and developed a variant called DEVIPs-ETPA. In DEVIPS-ETPA, the deployment of HPs is updated by using DEVIPS in [35]. The experimental results of ETPA and DEVIPS-ETPA are presented in Table II, which show that the proposed ETPA outperforms DEVIPS-ETPA in terms of mean EC. Further-



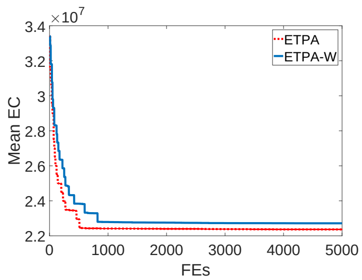
(a) N=60.



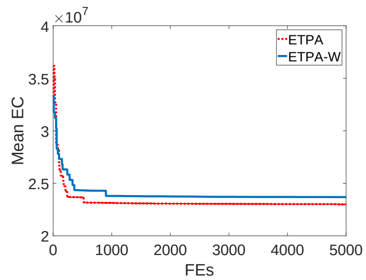
(b) N=80



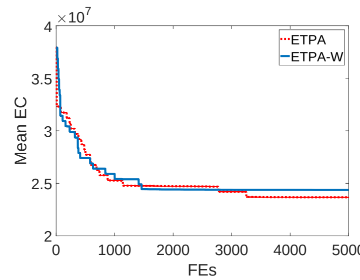
(c) N=100.



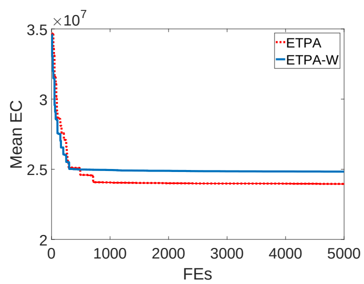
(d) N=120.



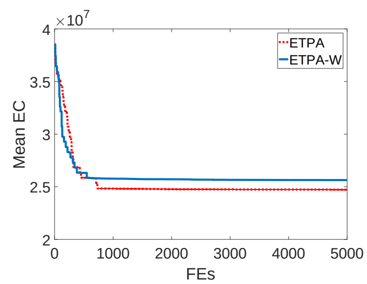
(e) N=140



(f) N=160.



(g) N=180



(h) N=200.

Fig. 3: Evolution of the mean EC obtained by ETPA and ETPA-W on eight instances over 20 runs.

more, as summarized at the bottom of Table II, ETPA provides better statistical results than DEVIPS-ETPA. Moreover, Figure 2 presents the evolution of the mean EC obtained by ETPA and DEVIPS-ETPA on four instances. Figure 2 shows that ETPA converges faster than DEVIPS-ETPA and maintains better performance during evolution. The better performance of ETPA is attributed as: since variable length GA in ETPA can always predict the optimal number of HPs quickly, thus leading to the performance improvement.

TABLE IV: Experimental results of ETPA and Kmeans-ETPA in terms of mean EC over 20 runs

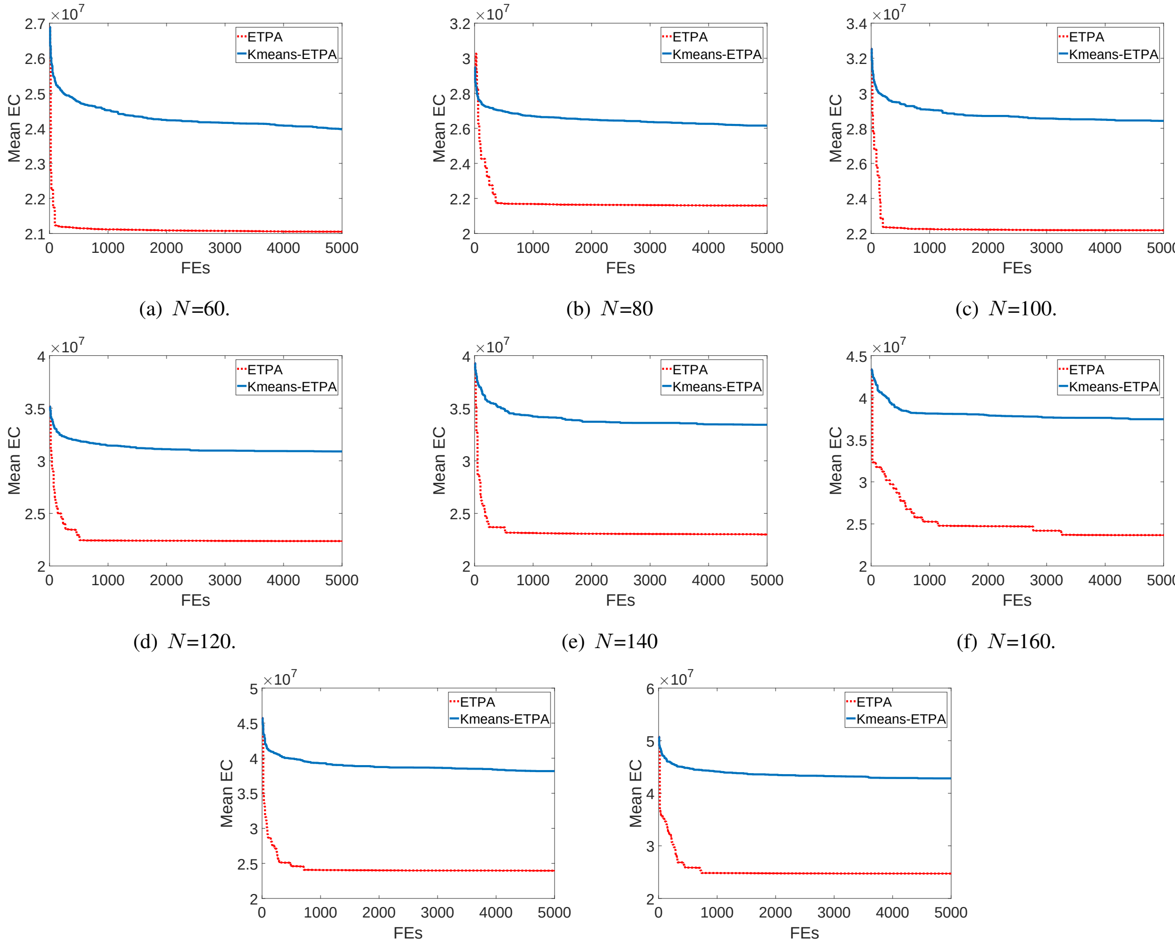
N	ETPA	Kmeans-ETPA
60	**2.11E+07** (3.61E+04)	2.40E+07 (4.03E+05) ↑
80	**2.16E+07** (4.57E+04)	2.62E+07 (2.55E+05) ↑
100	**2.22E+07** (9.28E+04)	2.84E+07 (6.18E+05) ↑
120	**2.24E+07** (5.69E+04)	3.09E+07 (4.98E+05) ↑
140	**2.30E+07** (7.88E+04)	3.34E+07 (7.99E+05) ↑
160	**2.37E+07** (8.38E+04)	3.74E+07 (1.32E+06) ↑
180	**2.40E+07** (9.74E+04)	3.82E+07 (1.06E+06) ↑
200	**2.47E+07** (1.26E+05)	4.28E+07 (1.39E+06) ↑
↑/↓/≈		8/0/0

TABLE V: Experimental results of ETPA and ETPA-Greedy in terms of mean EC over 20 runs

N	ETPA	ETPA-Greedy
60	**2.11E+07** (3.61E+04)	2.85E+07 (5.85E+05) ↑
80	**2.16E+07** (4.57E+04)	3.13E+07 (4.59E+05) ↑
100	**2.22E+07** (9.28E+04)	3.58E+07 (9.96E+05) ↑
120	**2.24E+07** (5.69E+04)	3.83E+07 (6.68E+05) ↑
140	**2.30E+07** (7.88E+04)	4.05E+07 (6.93E+05) ↑
160	**2.37E+07** (8.38E+04)	4.64E+07 (1.14E+06) ↑
180	**2.40E+07** (9.74E+04)	4.55E+07 (1.08E+06) ↑
200	**2.47E+07** (1.26E+05)	4.99E+07 (9.61E+05) ↑
↑/↓/≈		8/0/0

B. Effectiveness of Removing Redundant HPs

To restrict UAVs from visiting redundant HPs, we design an operator called remove redundant HPs in Algorithm 3. To show the effectiveness of this operator, we have tested ETPA with and without remove operator, where ETPA without remove operator is denoted by ETPA-W. The experimental results of ETPA and ETPA-W are listed in Table III, which show that the performance of ETPA is better than ETPA-W in terms of mean EC on all eight instances. In addition,



(a) N=60. (b) N=80 (c) N=100.

(d) N=120. (e) N=140 (f) N=160.

(g) N=180 (h) N=200.

Fig. 4: Evolution of the mean EC obtained by ETPA and Kmeans-ETPA on eight instances over 20 runs.

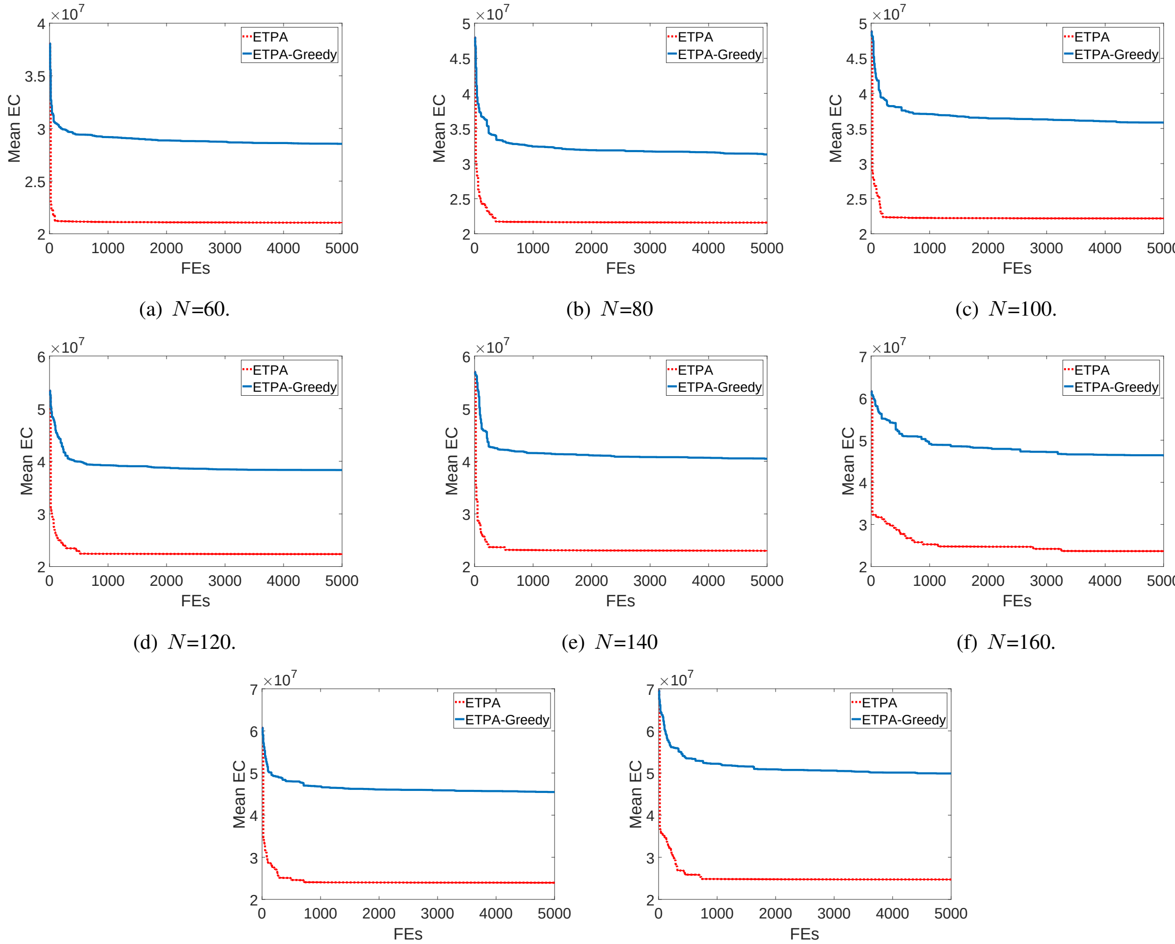
ETPA provides statistically better results than ETPA-W, as can be seen at the bottom of Table III. To further evaluate its effectiveness, Figure 3 presents the evolution of the mean EC of ETPA and ETPA-W on eight instances, which shows that ETPA converges faster than ETPA-W and maintains better performance during evolution. The reason why ETPA performs better than ETPA-W is straightforward: removing redundant HPs is to restrict UAVs from visiting extra/redundant HPs, thus saving the flying energy of the system.

C. Effectiveness of The Association between UAVs and HPs

To associate UAVs with HPs, this paper adopted DEC algorithm given in Algorithm 4. To show the effectiveness of the association between UAVs and HPs, we have replaced DEC with K-means algorithm [39] and designed an algorithm called Kmeans-ETPA. The experimental results of ETPA and Kmeans-ETPA are listed in Table IV, which show that the performance of ETPA is better than Kmeans-ETPA in terms of mean EC on all eight instances. In addition, ETPA provides statistically better results than ETPA-W, as can be seen at the bottom of Table IV. To further evaluate its effectiveness, Figure 4 presents the evolution of the mean EC of ETPA and Kmeans-ETPA on eight instances, which shows that ETPA converges faster than Kmeans-ETPA and maintains better performance during evolution. The reason why ETPA performs better than Kmeans-ETPA is straightforward: DEC algorithm in ETPA can group closely spaced HPs into the same cluster automatically without knowing the number of clusters, that reduces the EC of the system. In addition, it can also predict the optimal number of UAVs, which reduces the extra cost and improves the system EC.

D. Effectiveness of GA

To construct the order of HPs for UAVs, this paper adopted GA in Algorithm 5. To show the effectiveness of GA, we have replaced GA with a greedy algorithm and designed an algorithm called ETPA-Greedy. The experimental results of ETPA and ETPA-Greedy have listed in Table V, which show that the performance of ETPA is better than ETPA-Greedy in terms of mean EC on all eight instances. In addition, ETPA provides statistically better results than ETPA-W, as can be seen at the bottom of Table V. To further evaluate its effectiveness, Figure 5 presents the evolution of the mean EC of ETPA and Kmeans-ETPA on eight instances, which shows



(a) N=60. (b) N=80 (c) N=100.

(d) N=120. (e) N=140 (f) N=160.

(g) N=180 (h) N=200.

Fig. 5: Evolution of the mean EC obtained by ETPA and ETPA-Greedy on eight instances over 20 runs.

that ETPA converges faster than ETPA-Greedy and maintains better performance during evolution. The reason why ETPA performs better than ETPA-Greedy is straightforward: GA in ETPA is a famous evolutionary algorithm that is known for its good convergence in solving NP-hard problems.

V. Conclusion

This paper has presented a multi-UAV-assisted MEC system, where multiple UAVs have been used to serve MUs. A trajectory planning problem was formulated as an optimization problem with the aim of minimizing the system energy consumption. To solve the problem, we have proposed an evolutionary trajectory planning algorithm that consisted of three phases. In the first phase, a genetic algorithm with variable length individual in population was adopted for the deployment of HPs. This algorithm updates the number and location of HPs by using genetic operators designed for variable-length individuals. Afterward, the association between UAVs and HPs was determined by adopting DEC algorithm. Finally, a GA was adopted to construct the trajectories of all UAVs with the aim of reducing their flight distances. The experimental results on eight instances up to 200 MUs have shown that the proposed ETPA performs better than other compared variants in terms of minimizing the system energy consumption.

Ethical Approval

There is no need for ethical approval while conducting the study in this manuscript.

Funding Details

The authors are thankful to Qatar National Library Qatar for support to conduct study in this manuscript.

Conflict of Interest

The authors declare no conflict of interest.

Informed Consent

All authors have read this manuscript and are willing to process it for possible evaluation and publication.

Authorship Contributions

Muhammad Asim conceived the idea of this study. Wali Khan Mashwani guided the research and refined the idea. Muhammad Asim performed the research and drafted the manuscript. Samir Brahim Belhaouari discussed the results. Muhammad Asim and Wali Khan Mashwani revised and finalized the 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 Transactions on Emerging Topics in Computational Intelligence*, vol. 4, no. 6, pp. 742–763, 2020.

[2] A. Zaini and L. Xie, "Distributed drone traffic coordination using triggered communication," *Unmanned Systems*, vol. 08, 07 2019.

[3] M. Mozaffari, W. Saad, M. Bennis, Y. Nam, and M. Debbah, "A tutorial on UAVs for wireless networks: Applications, challenges, and open problems," *IEEE Communications Surveys Tutorials*, vol. 21, no. 3, pp. 2334–2360, 2019.

[4] J. E. Low, L. T. S. Win, D. S. B. Shaiful, C. H. Tan, G. S. Soh, and S. Foong, "Design and dynamic analysis of a transformable hovering rotorcraft (thor)," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 6389–6396.

[5] Y. Zeng, R. Zhang, and T. J. Lim, "Wireless communications with unmanned aerial vehicles: opportunities and challenges," *IEEE Communications Magazine*, vol. 54, no. 5, pp. 36–42, 2016.

[6] P. Olsson, J. KvarnstrÃűm, P. Doherty, O. Burdakov, and K. Holmberg, "Generating UAV communication networks for monitoring and surveillance," in *2010 11th International Conference on Control Automation Robotics Vision*, 2010, pp. 1070–1077.

[7] C. Yuan, K. A. Ghamry, Z. Liu, and Y. Zhang, "Unmanned aerial vehicle based forest fire monitoring and detection using image processing technique," in *2016 IEEE Chinese Guidance, Navigation and Control Conference (CGNCC)*, 2016, pp. 1870–1875.

[8] R. Gupta, A. Shukla, P. Mehta, P. Bhattacharya, S. Tanwar, S. Tyagi, and N. Kumar, "Vahak: A blockchain-based outdoor delivery scheme using UAV for healthcare 4.0 services," in *IEEE INFOCOM 2020 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2020, pp. 255–260.

[9] 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 *2015 European Conference on Networks and Communications (EuCNC)*, 2015, pp. 133–138.

[10] A. Merwaday and I. Guvenc, "UAV assisted heterogeneous networks for public safety communications," in *2015 IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*, 2015, pp. 329–334.

[11] 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 Transactions on Cybernetics*, vol. 50, no. 9, pp. 3984–3997, 2020.

[12] B. Zhang, G. Zhang, S. Ma, K. Yang, and K. Wang, "Efficient multitask scheduling for completion time minimization in UAV-assisted mobile edge computing," *Mobile Information Systems*, vol. 2020, pp. 1–11, 2020.

[13] 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 Network*, vol. 32, no. 3, pp. 42–51, 2018.

[14] 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 Transactions on Cognitive Communications and Networking*, pp. 1–1, 2020.

[15] Q. Wu and R. Zhang, "Common throughput maximization in UAV-enabled OFDMA systems with delay consideration," *IEEE Transactions on Communications*, vol. 66, no. 12, pp. 6614–6627, 2018.

[16] X. Diao, J. Zheng, Y. Cai, Y. Wu, and A. Anpalagan, "Fair data allocation and trajectory optimization for UAV-assisted mobile edge computing," *IEEE Communications Letters*, vol. 23, no. 12, pp. 2357–2361, 2019.

[17] S. Jeong, O. Simeone, and J. Kang, "Mobile edge computing via a UAV-mounted cloudlet: Optimization of bit allocation and path planning," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 3, pp. 2049–2063, 2018.

[18] X. Hu, K. Wong, K. Yang, and Z. Zheng, "UAV-assisted relaying and edge computing: Scheduling and trajectory optimization," *IEEE Transactions on Wireless Communications*, vol. 18, no. 10, pp. 4738–4752, 2019.

[19] Z. Yang, C. Pan, K. Wang, and M. Shikh-Bahaei, "Energy efficient resource allocation in UAV-enabled mobile edge computing networks," *IEEE Transactions on Wireless Communications*, vol. 18, no. 9, pp. 4576–4589, 2019.

[20] P. Huang, Y. Wang, and K. Wang, "Energy-efficient trajectory planning for a multi-UAV-assisted mobile edge computing system," *Frontiers of Information Technology & Electronic Engineering*, vol. 21, no. 12, pp. 1713–1725, 2020.

[21] Y. Zeng, J. Xu, and R. Zhang, "Energy minimization for wireless communication with rotary-wing UAV," *IEEE Transactions on Wireless Communications*, vol. 18, no. 4, pp. 2329–2345, 2019.

[22] C.-K. Ting, C.-N. Lee, H.-C. Chang, and J.-S. Wu, "Wireless heterogeneous transmitter placement using multiobjective variable-length genetic algorithm," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 39, no. 4, pp. 945–958, 2009.

[23] H. K. Mostapha, "Evolutionary data clustering in matlab," *https://yarpiz.com/64/ypml101-evolutionary-clustering, Yarpiz*, 2015.

[24] T. Liao, K. Socha, M. A. Montes de Oca, T. Stutzle, and M. Dorigo, "Ant colony optimization for mixed-variable optimization problems," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 4, pp. 503–518, 2014.

[25] K. P. Sinaga and M. Yang, "Unsupervised k-means clustering algorithm," *IEEE Access*, vol. 8, pp. 80 716–80 727, 2020.

[26] 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 Journal of Mathematics and Statistics*, vol. 47, no. 3, pp. 539–551, 2018.

[27] D. E. Goldberg and K. Deb, "A comparative analysis of selection schemes used in genetic algorithms," in *Foundations of Genetic Algorithms*. Morgan Kaufmann, 1991, pp. 69–93.

[28] K. Deb and R. Agrawal, "Simulated binary crossover for continuous search space," *Complex Syst.*, vol. 9, 1995.

[29] K. Deb and H. georg Beyer, "Real-coded genetic algorithms with simulated binary crossover: Studies on multi-modal and multi-objective problems," *Complex Systems*, pp. 431–454, 1995.

[30] K. Deb, K. Sindhya, and T. Okabe, "Self-adaptive simulated binary crossover for real-parameter optimization," in *Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation*, ser. GECCO '07. New York, NY, USA: Association for Computing Machinery, 2007, pp. 1187–1194.

[31] K. Deb and H. georg Beyer, "A combined genetic adaptive search (geneas) for engineering design," *Comput. Sci. Inf*, vol. 26, no. 4, pp. 30–45, 1996.

[32] E. Falkenauer, *Genetic Algorithms and Grouping Problems*. USA: John Wiley & Sons, Inc., 1998.

[33] E. R. Hruschka, R. J. G. B. Campello, A. A. Freitas, and A. C. Ponce Leon F. de Carvalho, "A survey of evolutionary algorithms for clustering," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 39, no. 2, pp. 133–155, 2009.

[34] A. K. Qin, V. L. Huang, and P. N. Suganthan, “Differential evolution algorithm with strategy adaptation for global numerical optimization,” *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 2, pp. 398–417, 2009.

[35] 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 Transactions on Emerging Topics in Computational Intelligence*, vol. 4, no. 3, pp. 324–335, 2020.

[36] D. L. Davies and D. W. Bouldin, “A cluster separation measure,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-1, no. 2, pp. 224–227, 1979.

[37] S. Bandyopadhyay and U. Maulik, “Genetic clustering for automatic evolution of clusters and application to image classification,” *Pattern Recognition*, vol. 35, no. 6, pp. 1197–1208, 2002. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S003132030100108X

[38] P. Larrañaga, C. M. H. Kuijpers, R. H. Murga, I. Inza, and S. Dizdarevic, “Genetic algorithms for the travelling salesman problem: A review of representations and operators,” *Artif. Intell. Rev.*, vol. 13, no. 2, p. 129âĂŞ170, Apr. 1999. [Online]. Available: https://doi.org/10.1023/A:1006529012972

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