A novel simulated annealing trajectory optimization algorithm in an autonomous UAVs-empowered MFC system for medical internet of things devices

Peer-reviewed author version

Asim, Muhammad; CHEN, Junhong; Muthanna, Ammar; Wenyin, Liu; Khan, Siraj & El-Latif, Ahmed A. Abd (2023) A novel simulated annealing trajectory optimization algorithm in an autonomous UAVs-empowered MFC system for medical internet of things devices. In: Wireless networks, 29 (7), p. 3163-3176.

DOI: 10.1007/s11276-023-03370-0
Handle: http://hdl.handle.net/1942/40213

A Novel Simulated Annealing Trajectory Optimization Algorithm in an autonomous UAVs-Empowered MFC System for Medical Internet of Things Devices.

Muhammad Asim[a,b], Chen Junhong[a*,c], Ammar Muthanna[d,e], Liu Wenyin[a], Siraj Khan[f], Ahmed A. Abd El-Latif[b*,g]

[a]*School of Computer Science and Technology, Guangdong University of Technology, Guangzhou 510006, PR, China (Email: asimpk@gdut.edu.cn; liuwy@gdut.edu.cn)*
[b]*EIAS Data Science Lab, College of Computer and Information Sciences, Prince Sultan University, Riyadh 11586, Saudi Arabia (aabdellatif@psu.edu.sa).*
[c]*Expertise Centre for Digital Media, Hasselt University, Hasselt, Belgium (CSChenjunhong@hotmail.com).*
[d]*Department of Telecommunication Networks and Data Transmission, The Bonch-Bruevich Saint-PetersburgState University of Telecommunications, 193232 Saint Petersburg, Russia (muthanna.asa@spbgut.ru)*
[e]*Department of Probability and Informatics, Peoples Friendship University of Russia (RUDN University), 6 Miklukho-Maklaya, 117198 Moscow, Russia*
[f]*School of Software Engineering, South China University of Technology, Guangzhou, P. R. China (sirajdurani@gmail.com).*
[g]*Department of Mathematics and Computer Science, Faculty of Science, Menoufia University, Shebin El-Koom 32511, Egypt (a.rahiem@gmail.com).*

Abstract

This article investigates a new autonomous mobile fog computing (MFC) system empowered by multiple unmanned aerial vehicles (UAVs) in order to serve medical Internet of Things devices (MIoTDs) efficiently. The aim of this article is to reduce the energy consumption of the UAVs-empowered MFC system by designing UAVs' trajectories. To construct the trajectories of UAVs, we need to consider not only the order of SPs but also the association among UAVs, SPs, and MIoTDs. The above-mentioned problem is very complicated and is difficult to be handled via applying traditional techniques, as it is NP-hard, nonlinear, non-convex, and mixed-integer. To handle this problem, we propose a novel simulated annealing trajectory optimization algorithm (SATOA), which handles the problem in three phases. First, the deployment (i.e., number and locations) of stop points (SPs) is updated and produced randomly using variable population sizes. Accordingly, MIoTDs are associated with SPs and extra SPs are removed. Finally, a novel simulated annealing algorithm is proposed to

---

⋆A Novel Simulated Annealing Trajectory Optimization Algorithm in an autonomous UAVs-Empowered MFC System for Medical Internet of Things Devices

∗Chen Junhong and Ahmed A. Abd El-Latif

Email address: `CSChenjunhong@hotmail.com; aabdellatif@psu.edu.sa` (Ahmed A. Abd El-Latif[b*,g])

optimize UAVs' association with SPs as well as their trajectories. The performance of SATOA is demonstrated by performing various experiments on nine instances with 40 to 200 MIoTDs. The simulation results show that the proposed SATOA outperforms other compared state-of-the-art algorithms in terms of saving energy consumption.

Keywords: `Mobile fog computing, simulated annealing algorithm, unmanned aerial vehicle, meta-heuristic algorithm`.

LIST OF ACRONYMS

UAV	Unmanned Aerial Vehicle	MEC	Mobile Edge Computing
QoS	Quality of Service	IoT	Internet of Things
ACO	Ant Colony Optimization	SA	Simulated Annealing
MFC	Mobile Fog Computing	ISA	Improved SA
TSP	Travelling Salesman Problem	DEC	Differential Evolution Clustering
MIoTD	Medical Internet of Things Devices	SATOA-W	SATOA without Remove Operator
SATOA	SA Trajectory Optimization Algorithm	TS	Tabu Search
EC	Energy Consumption	GA	Genetic Algorithm

1. Introduction

The number of resource-intensive and latency-sensitive applications in E-health care are growing day by day due to the advancement of mobile communication systems and health informatics [1][2]. These applications are usually latency-sensitive which require huge-computation resources. Since medical Internet of Things devices (MIoTDs) have some limitations like limited battery and computation resources, therefore, they are not capable of executing and processing these tasks [3].

To handle the above-listed issues, mobile fog computing (MFC) has been introduced for providing services with low latency near to users' end. It processes and executes MIoTDs' tasks at the nearby MFC server and returns the results to MIoTDs quickly [4][5]. Since MFC servers are close to MIoTDs, therefore, it can respond to MIoTDs' requests quickly with less energy consumption (EC) [6][7][8]. However, it is still limited to fulfill the MIoTDs requirements due to the fixed position of MFC servers, that cannot be adjusted according to MIoTDs' requirements. Therefore, it faces issues in providing timely services during natural disasters.

To satisfy the quality of service (QoS) requirements of MIoTDs, the integration of unmanned aerial vehicle (UAV) in MFC systems is esteemed as a promising technologies. UAVs-empowered MFC can handle the above-mentioned issues in MFC. Compared to the conventional MFC systems, UAVs-empowered MFC systems can attain a better QoS due to their flexibility and controllable mobility. In fact, the MFC system performance can be significantly improved with the assistance of UAVs, that can establish the better line-of-sight communication links between MIoTDs and MFC servers.

Recently, UAVs have been integrated and deployed in many fields, for example wireless communication [9] [10][11], military [12] [13], E-Health care [14], and rescue operations [15] [16]. They have also been integrated into MFC systems aiming to enhance their capabilities. For example, Zhang *et al.* [17] proposed a UAV-empowered mobile edge computing (MEC) for efficient multitask scheduling with aim of minimizing completion time of the system. Chen *et al.* [18] established a multilevel MFC offloading model, where UAV and fog node undertakes relay nodes and offloading computing nodes for computation-intensive and latency-critical tasks. Liu *et al.* [19] investigated the task offloading optimization problem of cruising UAV with fixed trajectory, where UAV provides limited-time task offloading services for multi-user nodes and multi-server nodes. Wang *et al.* [20] proposed a UAV-assisted computation offloading scheme based on deep reinforcement learning in a MEC framework. Lu *et al.* [21] proposed a secure communication scheme for the Dual-UAV-MEC system, where UAV server assists ground users in calculating the offloading tasks. Han *et al.* [22] proposed an optimized iterative algorithm to improve the secrecy performance of an UAV-assisted MEC system and assure secrecy transmit. Michailidis *et al.* [23] presented a novel UAV-aided MEC architecture for vehicular networks, where a hovering UAV can serve as an aerial road side units for task processing. Xu *et al.* [24] investigated a UAV-assisted relaying and MEC network. They used UAV as a MEC server to assist computation for the computation-hungry terminal devices, and also as a relay to deliver the sub-tasks to a ground access point for execution. Yang *et al.* [25] designed a multi-UAV deployment for MEC enhanced Internet of Things (IoT) architecture. They deployed multiple UAVs with computing offloading services for ground IoT devices.

Moreover, some researchers have investigated the trajectory planning and designing problem of UAVs in order to fully utilized their capabilities in MFC systems. For instance, Huang *et al.* [26] proposed an energy-efficient trajectory planning algorithm in a multi-UAV-empowered MEC system with the aim of minimizing EC of the system. Zeng *et al.* [27] studied a UAV-empowered wireless communication system. They optimized the trajectory of UAV, including the hovering locations and duration by proposing an efficient algorithm. Asim *et al.* [28] presented a novel algorithm called ETCTMA in order to design UAVs' trajectories in a UAVs-aided MEC system. Li *et al.* [29] investigated a UAV-assisted multi-task MEC networks by considering the quality of experience requirement of time-sensitive tasks of ground nodes. They jointly optimized the trajectory, resource allocation of UAV, and offloading decisions of ground nodes to minimize the total EC of them. Asim *et al.* [3] presented a new multi-UAV-aided MEC system by deploying several UAVs to provide services to users. They aimed to save the EC of the system via proposing a genetic trajectory planning algorithm. Sun *et al.* [30] presented a new UAV-assisted MEC framework. They jointly optimized the trajectory and CPU frequency of a fixed-wing UAV, and the offloading schedule to minimize the MEC of the UAV. Qin *et al.* [31] studied a UAV-assisted Fog-RAN network. They used UAV as mobile remote radio head to help base station forwards signals to the multiple users in the downlink transmissions. They jointly optimized the user scheduling and the UAVs trajectory in order to minimize the maximum transmission delay for all terrestrial users in downlink communication. Zhang *et al.* [32] studied the interference management problem by optimizing the power control and trajectory planning in a UAV-assisted wireless sensor network with the aim of maximizing the sum throughput of the target sensor. Asim *et al.* [33] presented a novel trajectory planning algorithm based on evolutionary algorithms in a MEC system assisted by multiple UAVs. Asim and Abd El-Latif [34] investigated a multi-UAV-empowered MEC system to save the system EC by designing UAVs' trajectories.

From the above-given introduction and related work, we came to know that the deployment of variable number of UAVs is still lacking in the current studies. In addition, the optimization of the number of UAVs and their association with SPs simultaneously have rarely been considered. This article considers the UAVs' trajectories planning problem in a MFC system empowered by multiple UAVs with the aim of minimizing the system EC including the EC of MIoTDs and UAVs at the same time.

The core contributions of this article are given as:

- A new UAVs-empowered MFC system is investigated and formulated in order to reduce EC of the studied systems via planning UAVs' trajectories.
- A novel simulated annealing (SA) trajectory optimization algorithm (SATOA) is proposed, that solves the problem in three phases. First, SPs are generated and updated randomly by varying population sizes. Accordingly, first, MIoTDs are associated with SPs and then extra SPs are ignored via applying a remove operator. Finally, an improved SA (ISA) is proposed to jointly handle UAVs' association with SPs and SPs' order for all UAVs.
- The performance of the proposed SATOA has been validated by performing extensive experiments on a set of nine instances with up to 200 MIoTDs.

The rest of the article is organized as follows. In Section 2, we introduce the system model and its problem formulation. Section 3 discusses our proposed algorithm SATOA. Section 4 and Section 5 present the experimental studies and conclusion this article, respectively.

2. System Model

As depicted in Fig. 1, we have $u \in \mathcal{U} = \{1, 2, ..., U\}$ MIoTDs and $l \in \mathcal{L} = \{1, 2, ..., L\}$ UAVs. All UAVs flyover on the MIoTDs to communicate and collect their data. The UAVs will hover over on some SPs for some time so that the MIoTDs can send their data to UAVs. We suppose that UAVs will stop over $t_l \in \mathcal{T}_l = \{1, 2, ..., T_l\}$ SPs. Thus, one can have

$$a_{ul}[t_l] = \{0, 1\}, \forall u \in \mathcal{U}, \forall t_l \in \mathcal{T}_l, \forall l \in \mathcal{L}, \tag{1}$$

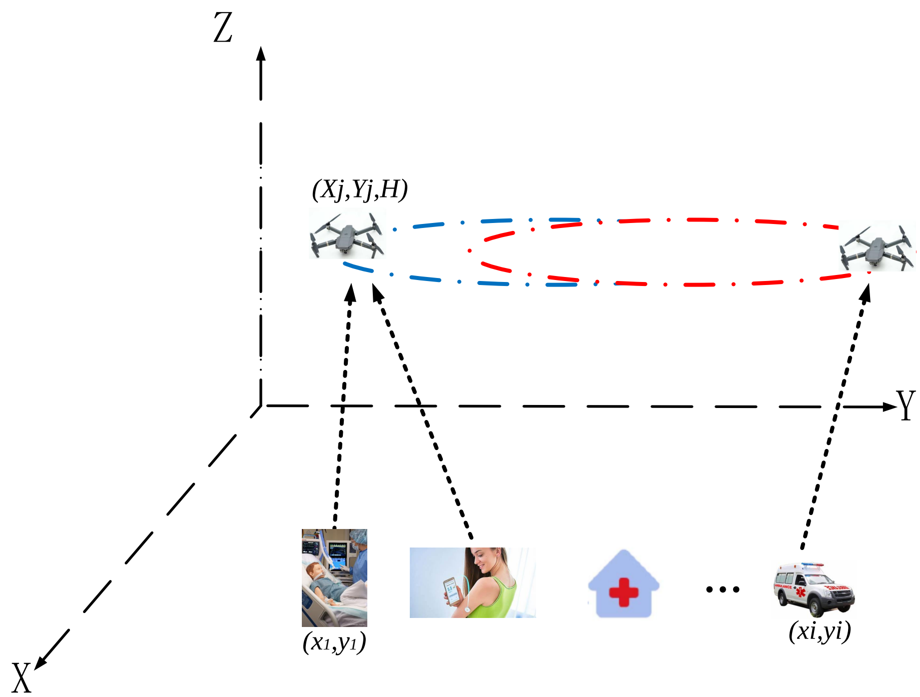


Figure 1: Multi-UAV-empowered MFC system with U MIoTDs and L UAVs

where $a_{ul}[t_l] = 1$ shows that the uth user/MIoTD intends to send its data to lth UAV in the tth time slot, while $a_{ul}[t_l] = 0$ denotes otherwise.

Also, we have

$$\sum_{t=1}^{T_l} \sum_{l=1}^{L} a_{ul}[t_l] = 1, u \in \mathcal{U} \tag{2}$$

Eq. 2 shows that one user/MIoTD can choose one UAV in one time slot. This means that all tasks of users/MIoTDs are indivisible and can not be divided into subtasks.

Assume that at each time slot, lth UAV may communicate with at most u_l MIoTDs, Therefore, we have

$$\sum_{u=1}^{U} a_{ul}[t_l] \leq u_l, t_l \in \mathcal{T}_l, l \in \mathcal{L} \tag{3}$$

It is assumed that uth user/MIoTD sends D_u amount of data to UAV l. The UAV may hover at T_l SPs, where each SP t_l lasts for maximum T_{max} seconds, where T_{max} is a constant value.

Then, the transmission time of sending data from user/MIoTD to UAV in the t_lth time slot is as

$$T_u^{Tr}[t_l] = \frac{D_u}{r_{ul}[t_l]}, \ \ \forall l \in \mathcal{L}, t_l \in \mathcal{T}_l \tag{4}$$

where $r_{ul}[t_l]$ is the data rate defined in (11).

Also, if F_u is the CPU cycles needed to process a task. Then, the process time of processing a task of user/MIoTD at SP t_l by UAV l is given as:

$$T_u^{C}[t_l] = \frac{F_u}{f_{ul}[t_l]}, \ \ \forall l \in \mathcal{L}, \forall t_l \in \mathcal{T}_l \tag{5}$$

where $f_{ul}[t_l]$ denotes the computation capacity of the UAV.

Also, we have

$$T_u[t_l] = T_u^{C}[t_l] + T_u^{Tr}[t_l], u \in \mathcal{U}, t_l \in \mathcal{T}_l \tag{6}$$

Then, one can have

$$T_l^{H}[t_l] = \max_{\boldsymbol{u \in \mathcal{U}}} \{T_u^{C}[t_l] + T_u^{Tr}[t_l], l \in \mathcal{L}, t_l \in \mathcal{T}_l\}, \tag{7}$$

The coordinates of u-th user/MIoTD and l-th UAV are assumed to be (x_u, y_u) and $(X_l[t_l], Y_l[t_l], H)$, respectively. We also assume that the UAV's trajectory is characterized by a sequence of location $q[t] = [X_l[t_l], Y_l[t_l], H]^T$, where the height H of all UAVs is constant. Also, we have

$$||q_l[t_l + 1] - q_l[t_l]||^2 \leq D_{max}^2, t_l = 0, ..., T_l \tag{8}$$

where $D_{max} = V_{max} \cdot T_{max}$ is the maximum horizontal distance of the UAV. V_{max} shows the maximum velocity of the UAV.

The distance of uth user/MIoTD from lth UAV at t_lth SP is defined as:

$$d_{ul}[t_l] = \sqrt{(X_l[t_l] - x_u)^2 + (Y_l[t_l] - y_u)^2 + H^2}, \forall u \in \mathcal{U}, \forall t_l \in \mathcal{T}_l \tag{9}$$

Then, the channel power gain can be given as

$$h_{ul}[t_l] = \frac{\beta_0}{d_{ul}[t_l]^2} \tag{10}$$

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

The data rate of offloading a task from MIoTDs to the UAVs can be defined as:

$$r_{ul}[t_l] = B\log_2\left(1 + \frac{p^u h_{ul}[t_l]}{\sigma^2}\right) \tag{11}$$

where σ^2 denotes noise power and p^u presents transmission power.

The EC for offloading a task of the u-th user/MIoTD to the l-th UAV at SP t_l is defined as

$$E_{ul}^{Tr}[t_l] = p^u T_u^{Tr}[t_l] = \frac{p^u D_u}{r_{ul}[t_l]}, \ \ \forall l \in \mathcal{L}, t_l \in \mathcal{T}_l \tag{12}$$

The total EC of all MIoTDs is given as

$$E_U = \sum_{u=1}^{U} \sum_{l=1}^{L} \sum_{t_l=1}^{T_l} a_{ul}[t_l] E_{ul}^{Tr}[t_l] \tag{13}$$

Since the flying EC of UAVs are directly proportional to their flying distances, thus one can calculate flying EC as

$$E_l^F = P^F \sum_{t_l=1}^{T_l-1} ||q_l[t_l+1] - q_l[t_l]||^2 \tag{14}$$

Also, the hovering EC can be calculated as

$$E_l^H = P^H \sum_{t_l=1}^{T_l-1} T_l^H[t_l], \tag{15}$$

where P^H represents UAV's hovering power.

The total EC of UAVs is expressed as

$$E_{UAV} = \sum_{l=1}^{L} (E_l^F + E_l^H) \tag{16}$$

The total EC denoted by E can be calculated as

$$E = E_{UAV} + \alpha E_U \tag{17}$$

where α is a weighted coefficient between the EC of UAVs and MIoTDs.

Then, the problem formulation is given as.

$$\mathcal{P}: \min_{\boldsymbol{a_{ul}[t_l]}, \boldsymbol{T_l}, \boldsymbol{q_l[t_l]}, \boldsymbol{M}} E \tag{18a}$$

subject to:

$$a_{ul}[t_l] = \{0,1\}, \forall u \in \mathcal{U}, \forall t_l \in \mathcal{T}_l, \forall l \in \mathcal{L}, \tag{18b}$$

$$\sum_{t_l=1}^{T_l} \sum_{l=1}^{L} a_{ul}[t_l] = 1, u \in \mathcal{U}, \tag{18c}$$

$$\sum_{u=1}^{U} a_{ul}[t_l] \leq U_l, t_l \in \mathcal{T}_l, l \in \mathcal{L}, \tag{18d}$$

$$||q_l[t_l+1] - q_l[t_l]||^2 \leq D_{max}^2, t_l = 0, ..., T_l, \tag{18e}$$

$$X_{min} \leq X_l[t_l] \leq X_{max}, \ \forall l \in \mathcal{L}, t_l \in \mathcal{T}_l, \tag{18f}$$

$$Y_{min} \leq Y_l[t_l] \leq Y_{max}, \ \forall l \in \mathcal{L}, t_l \in \mathcal{T}_l. \tag{18g}$$

Where (18f) and (18g) denote the boundaries of X-axis and Y-axis, respectively.

3. The Proposed Algorithm and Challenges

3.1. Challenges

When solving (18(a)), one has to consider the following challenges.

- To solve $\mathcal{P}$, one has to take for the deployment updation of SPs, their association with UAVs, and their order for UAVs, which are strongly coupled with heach other as well. Therefore, it is a complex problem to be handled by traditional optimization techniques.
- $\mathcal{P}$ is a mixed-decision variables problem containing various decision variables, like integer variables (L and T_l), binary decision variable $a_{ul}[t_l]$, and continuous decision variables ($X_l[t_l]$ and $Y_l[t_l]$). Therefore, it is hard/challenging to be handled via applying traditional optimization techniques.

In this article, we presented a novel algorithm named as SATOA to construct the trajectories of UAVs. SATOA solve $\mathcal{P}$ in three phases: 1) the optimization of of SPs' deployment, 2) optimization of the association between MIoTDs and SPs and ignoring extra SPs, and 3) the optimization of the association between SPs and UAVs and the trajectories' construction of UAVs.

3.2. SATOA

Algorithm 1 presents the SATOA' pseudo-code. Firstly, the initial population $\mathcal{POP}$ consisting of SPs is generated randomly which is given as $\mathcal{POP}$ = (X_1, Y_1), (X_2, Y_2),...,(X_{max}, Y_{max}). Subsequently, MIoTDs are associated with SPs by Eq. 19 and extra SPs are ignored via Algorithm 5. Accordingly, a novel ISA is proposed to group SPs to associate SPs with UAVs and construct their

Algorithm 1 Psuedocode of SATOA

$FEs = 0$;
repeat
Initialize the population $\mathcal{POP}$ containing SPs randomly;
Associate MIoTDs with SPs in $\mathcal{POP}$ using 19 and remove extra SPs by Algorithm 5.
Jointly group SPs of $\mathcal{POP}$ into various clusters and construct the trajectories in each cluster using ISA in Algorithm 4;
Evaluate population $\mathcal{POP}$ using $\mathcal{P}$;
$FEs = FEs + 1$;
while $FEs < FEs_{max}$ **do**
  Generate random SPs which form population $\mathcal{POP}_O$;
  for i = 1 : $|\mathcal{POP}_O|$ **do**
    Construct $\mathcal{POP}_1$, $\mathcal{POP}_2$, and $\mathcal{POP}_3$ by using Algorithm 2;
    for l = 1 : 3 **do**
      Associate MIoTDs with SPs in $\mathcal{POP}_l$ using 19 and remove extra SPs by Algorithm 5.
      Jointly group SPs in $\mathcal{POP}_l$ into different clusters and construct their order for UAVs via ISA in Algorithm 4;
    end for
    Evaluate all three populations $\mathcal{POP}_1$, $\mathcal{POP}_2$, and $\mathcal{POP}_3$ using $\mathcal{P}$;
    $FEs = FEs + 3$;
    Replace population $\mathcal{POP}$ with a best feasible population among all three populations $\mathcal{POP}_1$, $\mathcal{POP}_2$, and $\mathcal{POP}_3$.
  end for
end while
Output: Best Solution i.e., Best $\mathcal{POP}$.

Algorithm 2 Generating Three New Populations

$\mathcal{POP}_1 \leftarrow$ add an individual $i \in \mathcal{POP}_O$ to $\mathcal{POP}$;
$\mathcal{POP}_2 \leftarrow$ replace a random individual in $\mathcal{POP}$ with $i \in \mathcal{POP}_O$;
$\mathcal{POP}_3 \leftarrow$ remove a random individual from $\mathcal{POP}$;
Output: $\mathcal{POP}_1$, $\mathcal{POP}_2$, $\mathcal{POP}_3$

order for UAVs. Afterward, $\mathcal{POP}$ is evaluated via $\mathcal{P}$. If $\mathcal{POP}$ is feasible, it is accepted; otherwise, it is regenerated until it becomes feasible or the stopping criteria does not meet. Accordingly, a new population $\mathcal{POP}_O$ is first generated randomly during the evolution. By adopting individuals of $\mathcal{POP}_O$, we construct three new populations $\mathcal{POP}_1$, $\mathcal{POP}_2$, and $\mathcal{POP}_3$ using insertion, deletion, and replacement. Then, the SPs in each $\mathcal{POP}_1$, $\mathcal{POP}_2$, and $\mathcal{POP}_3$ are associated with MIoTDs by using Eq. 19. Afterward, the extra SPs in them are removed using Algorithm 5. Accordingly, the SPs in $\mathcal{POP}_1$, $\mathcal{POP}_2$, and $\mathcal{POP}_3$ are clustered into various clusters as well as designed the trajectories of UAVs in them via ISA in Algorithm 4. Then, $\mathcal{POP}_1$, $\mathcal{POP}_2$, and $\mathcal{POP}_3$ are evaluated using $\mathcal{P}$. Finally, $\mathcal{POP}$ is replaced with the best feasible population among $\mathcal{POP}_1$, $\mathcal{POP}_2$, and $\mathcal{POP}_3$. The process continues until the stopping criteria is satisfied i.e., $FEs \geq FEs_{max}$.

3.3. Optimization of the SPs' Deployment

The optimization of the deployment of SPs is optimized via inserting, replacing, or removing an individual randomly in/from $\mathcal{POP}$. The individuals

Algorithm 3 ISA

Initialize: $POPSize$, minimum tour Min_t, maximum D_{max}, $MaxIt$, $MaxIt\text{-}Inner$, Initial temperature T_0, and Damping rate of Temperature α_0.
$num_{brks} \leftarrow$ Initialize the number of break points randomly;
pop_{rte} = zeros($POPSize$,n);
pop_{brk} = zeros($POPSize$,num_{brks});
for k = 1:$POP - Size$ **do**
  $pop_{rte}(k,:) = randperm(n)$+1;
  $pop_{brk}(k,:) = randbreaks()$;
end for
$BestCost \leftarrow$ Evaluate $\mathcal{POP}$ by using local optimizer in Algorithm 4;
$BestSol \leftarrow$ Best solution for $BestCost$;
for $It = 1 : MaxIt$ **do**
  for It_2 = 1:$MaxIt\text{-}Inner$ **do**
    $New\text{-}Sol \leftarrow$ Create Neighbor solution by applying Swap, Reversion, and Insertion operators;
    $NewCost \leftarrow$ Evaluate $New\text{-}Sol$ by using local optimizer in Algorithm 4;
    if $NewCost \leq Cost$ **then**
      $Sol \leftarrow New\text{-}Sol$;
    else
      δ=$NewCost$-$Cost$;
      $p = e^{\frac{-\delta}{T}}$;
      if $rand \leq p$ **then**
        $Sol \leftarrow New\text{-}Sol$;
      end if
    end if
    if $Cost \leq BestCost$ **then**
      $BestSol \leftarrow Sol$;
    end if
  end for
  $T = \alpha_0 \times T$;
end for
OUTPUT: NEW POP $\mathcal{POP}_N$

used to replace and insert in $\mathcal{POP}$ are taken from a randomly produced population called $\mathcal{POP}_O$. That is the individuals of $\mathcal{POP}_O$ are used to update the parent population $\mathcal{POP}$ using Algorithm 2. Since each individual in $\mathcal{POP}$ denotes a location of SP. Therefore, the whole population $\mathcal{POP}$ denotes a single deployment of SPs. Hence, the size of population is equal to the number of SPs. Three new population $\mathcal{POP}_1$, $\mathcal{POP}_2$, and $\mathcal{POP}_3$ are constructing using Algorithm 2. One can see from Algorithm 2 that the population sizes of all three populations $\mathcal{POP}_1$, $\mathcal{POP}_2$, and $\mathcal{POP}_3$ are different from one another. As a result, the population sizes are kept variable during the optimization of SPs' deployment. Hence, the SPs' locations and their number are optimized via repeating the above process.

3.4. Optimization of the Association between MIoTDs and SPs

This subsection is devoted to associate MIoTDs with SPs and then based on it we remove extra/redundant SPs in order to minimize the EC of the system.

Algorithm 4 TSP Optimizer Algorithm

```
for p = 1:POPSize do
    I_p ← POP(p)
    for r = 1:|I_p| do
        uav_r = I_p(r);
        if uav_r ≠ ∅ then
            d_2 = d_2 + dmat(1, uav_r(1))/Para.V + Th(uav_r(1));
            for k = 1 : length(uav_r) − 1 do
                d_2 = d_2 + dmat(uav_r(k), uav_r(k + 1))/Para.V + Th(uav_r(k + 1));
            end for
            d_2 = d_2 + dmat(uav_r(end), 1)/Para.V;
            if d2 > D_max then
                d2 = d2 + (d2 − maxtour) * penalty-rate;
            end if
        end if
        D(r) ← d_2;
        d ← d + d_2;
    end for
end for
BestCost = d;
```

Algorithm 5 Remove Extra SPs

```
U_assoc ← Unique association between MIoTDs and SPs;
S_D ← The set difference of the index set of SPs/POP and U_assoc;
POP^U ← Update POP via ignoring/deleting SPs having indexes in set S_D;
```

We associate MIoTDs with SPs by the following equation.

$$a_{ul}[t_l] = \begin{cases} 1, & if(u, l, t_l) = \underset{u \in \mathcal{U}, l \in \mathcal{L}}{\operatorname{argmin}}(d_{ul}[t_l]), \\ 0, & \text{otherwise}. \end{cases} \tag{19}$$

Which shows that MIoTDs always communicate and exchange their data with UAVs at the nearest SPs.

After associating MIoTDs with SPs, one can see that there are some SPs having no user/MIoTD to served. Such SPs are called extra/redundant SPs, which need to be removed to avoid extra EC. We remove extra/redundant SPs by an operator given in Algorithm 5. As described in Algorithm 5, we first find the unique association between MIoTDs and SPs. After that, we calculate the set difference S_D between the index set of SPs/POP and unique association set U_{assoc}. Finally, we remove SPs with indexes in S_D from POP to get an updated population POP^U. Hence, by using remove operator, we remove all redundant SPs from POP.

3.5. ISA for handling Clustering of SPs and the trajectories construction of UAVs

SPs are associated with UAVs by clustering SPs into clusters along with their optimal order in clusters in this subsection. Inspired from [35] [36], in SATOA, we proposed a novel improved ISA to jointly handle the optimization of the association between UAVs and SPs and designing of optimal trajectories for all UAVs. In addition, the proposed algorithm can also predict the appropriate number of UAVs. We borrowed the idea of [35] and [36] to handle the multi-UAVs routing problem.

Evolutionary algorithms are population-based algorithms mainly inspired by biological evolution. Recently, various EAs were proposed under the umbrella of evolutionary computing [37] [38]. EC is one of the emerging sub field of artificial intelligence and soft computing have had tackled various benchmark functions and real-world problems. Evolutionary algorithms, SA, and Tabu Search (TS) are general iterative algorithms for combinatorial optimization. SA is a probabilistic meta-heuristic algorithm which is first proposed in [39] [40]. It starts with an initial solution and then search in solution space iteratively for improving the initial solution. Quality of solutions in SA is improved during the search concerning a given measure of quality. It has been used for the solution of various optimization problems.

As described in Algorithm 3, ISA works on the following three inputs, the set of SPs, the matrix of UAVs flying times, and the matrix of their hovering time. In addition, it requires some other inputs, like population size, the maximum-iterations $MaxIt$, the maximum inner-iterations $MaxIt\text{-}Inner$, the tour constraints (i.e., minimum and maximum flight times) for UAVs, initial temperature T_0, and damping rate of temperature α_0.

First, the initial population is initialized via generating UAVs routes randomly. Then, the overall time/fitness values for UAVs are calculated via using a local UAV/TSP solver given in Algorithm 4, where the fitness function can be defined as the sum of flying times and hovering times of UAVs. Subsequently, ISA operators i.e., swap, reversion, and insertion are applied to generate a new solution $New\text{-}Sol$. Accordingly, the new solution $New\text{-}Sol$ is evaluated by the local UAV/TSP solver in Algorithm 4. The new solution is compared with the old solution and the best between them is selected for the next iteration.

The TSP or UAV solver which is also known as local optimizer is described by Algorithm 4. This algorithm optimizes the routes of UAVs independently. Further details of TSP/UAV solver can be fund in [36, 41]. The long routes that exceed the constraint (18d) are realized as uncommonly. Such routes get a penalty and are split into shorter routes by the route/chromosome partition operator. Thus, all UAVs' routes do not exceed the constraint (18d), but the splitting of routes may lead to increase the number of UAVs. As there exists a constraint for the number of UAVs and ISA tries to minimize the number of UAVs, therefore this penalty may have a great impact on the fitness function while solving $\mathcal{P}$ in (18).

4. Simulation Results

Table 1 presents the parameter settings of the studied UAVs-empowered MFC system. In this article, we have tested nine instances range from 40 to 200 MIoTDs to evaluate the effectiveness of SATOA. Similar to [3], it is assumed that the UEs locations are randomly generated in a $1000 \times 1000\ \mathrm{m}^2$ region [3]. In

Table 1: Parameters Setting

Parameter	Value	Parameter	Value
$D_u; (u \in U)$	$[1, 10^3]MB$	P	0.1 W
P^H	1000	V_{max}	20 m/s
P^F	1000	σ^2	-174 dBm
B	1 MHz	α_0	0.98
α	-30 dB	L	4
λ	1	d	$\lambda/2$
β	2.8	H^U	200
X_{max}	1000	Y_{max}	1000
$q_l[0]$	[0 0 200]		

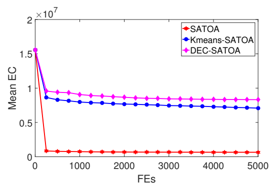
SATOA, parameters were set as: T_0=100, $\alpha_0 = 0.98$, $Min_t = 1$, $MaxIt$=100, $MaxIt$-$Inner$=50, and maximum tour D_{max} = 10000 m. Maximum fitness evaluations (FEs_{max}) was set to 5000 and 20 runs were applied independently on all algorithms. To further validate the effectiveness, we have carried out the Wilcoxon rank-sum test at significant level of 0.05. We have used $\approxeq$, $+$, and $-$ in order to show that SATOA performs similar to, better than, and worse than the compared algorithms, respectively. All the simulations were implemented in MATLAB R2016a on a PC with an Intel(R) Core(TM) i7-8700 CPU @3.20 GHz and 16 GB RAM. Furthermore, the better mean EC obtained by TPaPBA and its competitor are highlighted bold in the tables for the sack of fare comparison and differentiation among SATOA and each compared algorithm in this paper.

In order to validate the performance and effectiveness of SATOA, we design two variants called Kmeans-SATOA and DEC-SATOA. Kmeans-SATOA adopts K-means algorithm [42] for the association between UAVs and SPs and a greedy algorithm for the construction of trajectories of UAVs, while DEC-SATOA adopts a clustering algorithm based on differential evolution denoted by DEC [43] to associate UAVs and SPs and a greedy algorithm for constructing the trajectory of each UAV. K-means requires a predefined clusters number which is assumed to be 6 here, while DEC can clusters SPs without any initialization. The population size is set to 10 and the maximum number of iterations to 50 for DEC in DEC-SATOA. Table 2 lists the mean EC and standard-deviation (STD) obtained by SATOA, Kmeans-SATOA, and DEC-SATOA . Moreover, Figure 2 plots mean EC of SATOA, Kmeans-SATOA, and DEC-SATOA against FEs on nine instances. Table 2 and Figure 2 show that SATOA outperforms Kmeans-SATOA and DEC-SATOA in reducing the system mean EC. Furthermore, SATOA produces better statistical results than Kmeans-SATOA and DEC-SATOA, as can be seen in the last row of Table 2.

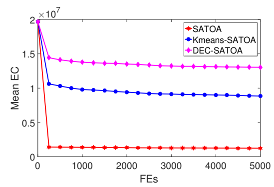
To ignore redundant SPs serving no MIoTDs, a remove operator in Algorithm 5 was designed. The effectiveness of this operator is investigated by testing SATOA with and without this operator, where SATOA without remove operator is named as SATOA-W. Figure 3 depicts the evolution of mean EC of SATOA and SATOA-W on nine instances: N = 40, 60,.., 200, which shows

Table 2: Simulation results of SATOA, Kmeans-SATOA, and DEC-SATOA

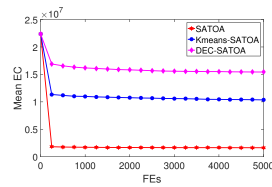
N	SATOA	Kmeans-SATOA	DEC-SATOA
	Mean EC (STD)	Mean EC (STD)	Mean EC (STD)
40	**6.51E+05** (3.25E+04)	7.09E+06 (8.67E+05) +	8.32E+06 (1.13E+06) +
60	**1.24E+06** (9.85E+04)	8.84E+06 (9.38E+05) +	1.30E+07 (8.70E+05) +
80	**1.64E+06** (5.56E+04)	1.04E+06 (7.10E+05) +	1.54E+07 (1.34E+06) +
100	**2.37E+06** (1.61E+05)	1.22E+06 (7.50E+05) +	2.04E+07 (1.12E+06) +
120	**2.58E+06** (1.21E+05)	1.18E+07 (9.37E+05) +	2.17E+07 (1.24E+06) +
140	**3.36E+06** (1.30E+05)	1.26E+07 (1.13E+06) +	2.44E+07 (1.04E+06) +
160	**4.07E+06** (1.44E+05)	1.37E+07 (1.44E+06) +	2.91E+07 (1.46E+06) +
180	**4.67E+06** (9.54E+04)	1.49E+07 (1.33E+06) +	3.07E+07 (1.26E+06) +
200	**5.42E+06** (1.30E+05)	1.58E+07 (1.75E+06) +	3.43E+07 (9.18E+05) +
+, −, ≊		10/0/0	10/0/0



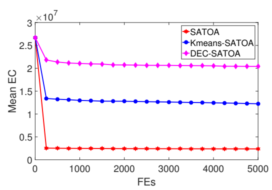
(a) N=40.



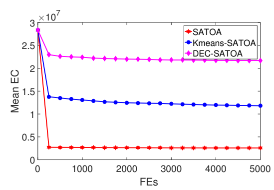
(b) N=60.



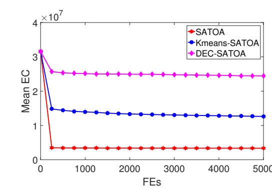
(c) N=80



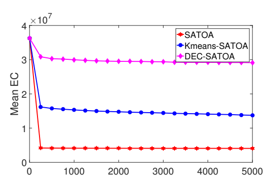
(d) N=100.



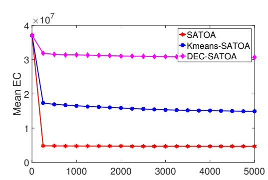
(e) N=120



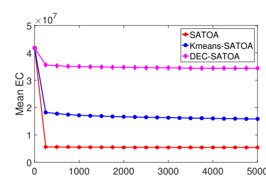
(f) N=140.



(g) N=160

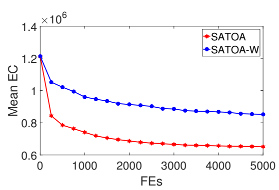


(h) N=180.

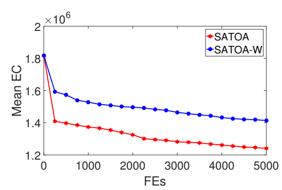


(i) N=200

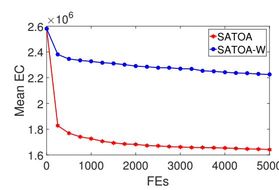
Figure 2: Mean EC comparison of SATOA, Kmeans-SATOA, and DEC-SATOA. Where FEs shows fitness evaluations



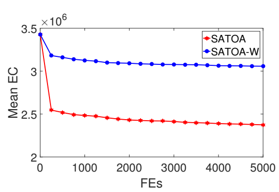
(a) N=40.



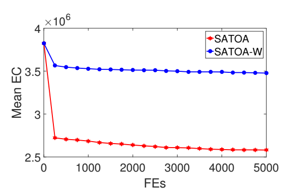
(b) N=60.



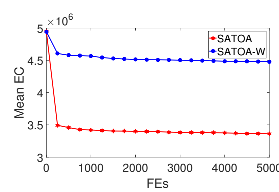
(c) N=80



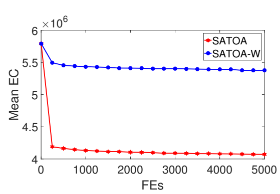
(d) N=100.



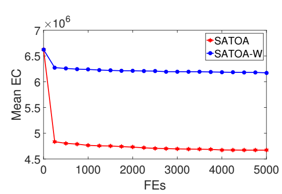
(e) N=120



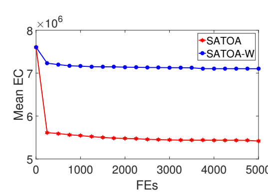
(f) N=140.



(g) N=160



(h) N=180.



(i) N=200

Figure 3: Mean EC comparison of SATOA and SATOA-W

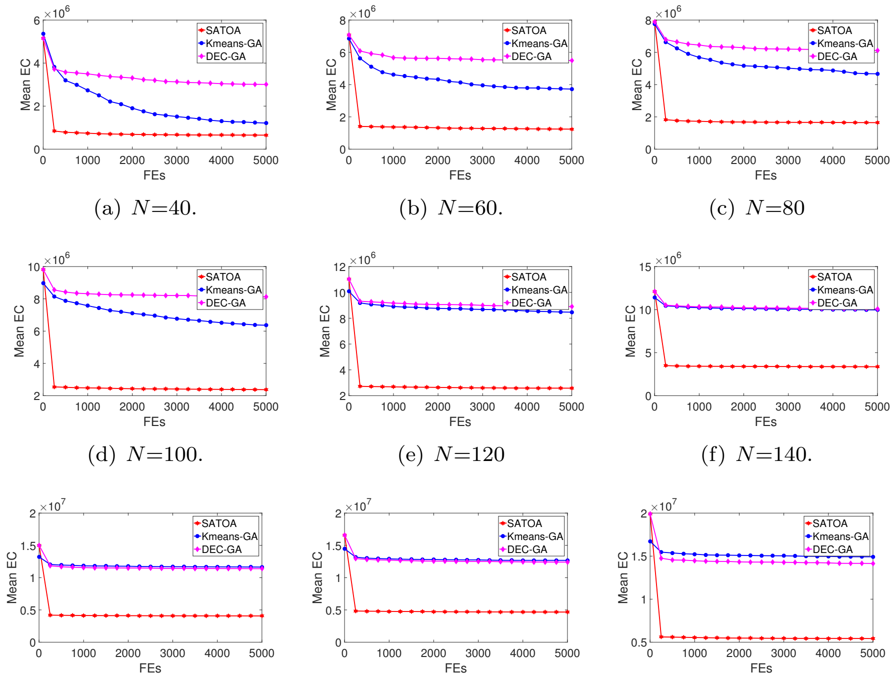
that the SATOA converges faster than SATOA-W in early stages and provides better results. SATOA performs better than SATOA-W due to the following reason: the remove operator restricts UAVs to ignore unnecessary SPs serving no MIoTDs, therefore saving the system EC.

Table 3: Simulation results of SATOA, Kmeans-GA, and DEC-GA

N	SATOA	Kmeans-GA	DEC-GA
	Mean EC (STD)	Mean EC (STD)	Mean EC (STD)
40	**6.51E+05** (3.25E+04)	1.20E+06 (7.20E+05) +	3.01E+06 (4.89E+05) +
60	**1.24E+06** (9.85E+04)	3.72E+06 (1.23E+06) +	5.49E+06 (4.62E+05) +
80	**1.64E+06** (5.56E+04)	4.67E+06 (9.16E+05) +	6.12E+06 (3.89E+05) +
100	**2.37E+06** (1.61E+05)	6.33E+06 (1.13E+06) +	8.13E+06 (3.04E+05) +
120	**2.58E+06** (1.21E+05)	8.45E+06 (7.67E+05) +	8.90E+06 (2.60E+05) +
140	**3.36E+06** (1.30E+05)	9.95E+06 (3.82E+05) +	1.01E+07 (4.60E+05) +
160	**4.07E+06** (1.44E+05)	1.16E+07 (4.58E+05) +	1.14E+07 (3.36E+05) +
180	**4.67E+06** (9.54E+04)	1.27E+07 (3.88E+05) +	1.24E+07 (3.73E+05) +
200	**5.42E+06** (1.30E+05)	1.49E+07 (3.77E+05) +	1.41E+07 (3.66E+05) +
$+, -, \approxeq$		10/0/0	10/0/0

To further validate the effectiveness of the order of SPs in the proposed SATOA, we compare SATOA with other algorithms called Kmeans-GA and DEC-GA. Kmeans-GA and DEC-GA are constructed by replacing greedy algorithm with genetic algorithm (GA) in Kmeans-Greedy and DEC-Greedy, respectively [44]. That is both Kmeans-GA and DEC-GA use GA for the construction of trajectories of UAVs. The mean EC and STD of SATOA, Kmeans-GA, and DEC-GA are listed in Table 3. Table 3 shows that the proposed SATOA outperforms Kmeans-GA and DEC-GA in terms of mean EC. The statistical results of SATOA, Kmeans-GA, and DEC-GA are listed in the last row of Table 3, which reveals that SATOA produces better statistical results compare with Kmeans-GA and DEC-GA. Moreover, Figure 4 depicts mean EC's evolution obtained by SATOA, KMeans-GA, and DEC-GA on nine instances over 20 runs. Figure 4 shows that the SATOA converges faster and provides better results compared with KMeans-GA and DEC-GA. Furthermore, Figure 5 presents the mean running time of SATOA, KMeans-GA, and DEC-GA for all nine instances, which shows that the mean running time of the proposed SATOA is much smaller than that of the compared algorithms.

Overall, the proposed SATOA provides average percentage improvements in mean EC on all instances upto 36.96%, 30.46%, 57.67%, and 12.25% compared with Kmeans-SATOA, DEC-SATOA, Kmeans-SATOA, and DEC-SATOA, respectively. The superiority and better performance of SATOA against Kmeans-SATOA, DEC-SATOA, Kmeans-SATOA, and DEC-SATOA can be attributed to the following aspect: in SATOA-GA, the association problem between SPs and UAVs and UAVs' trajectories problem are jointly solved that leads to better performance. However, in other compared algorithms, they are addressed/optimized independently.



(a) N=40. (b) N=60. (c) N=80

(d) N=100. (e) N=120 (f) N=140.

(g) N=160 (h) N=180. (i) N=200

Figure 4: Mean EC comparison of SATOA, KMeans-GA, and DEC-GA.

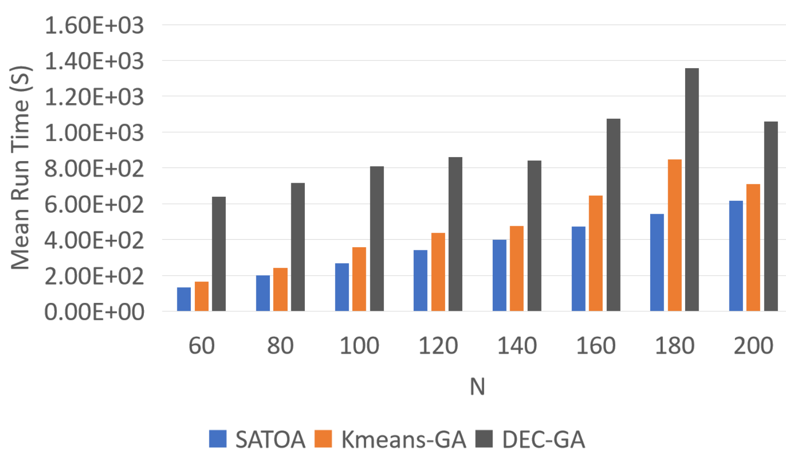


Figure 5: Mean running time of SATOA, KMeans-GA, and DEC-GA.

5. Conclusions

This article has presented an autonomous UAVs-empowered MFC system, where several UAVs have been integrated as MFC servers to provide services MIoTDs. We have formulated the system model as an optimization problem to minimize the system's EC. We proposed a novel simulated annealing trajectory optimization algorithm (SATOA) that consisted of three phases. First, the SPs' deployment was generated and updated randomly with variable population sizes. Accordingly, MIoTDs were associated with SPs and extra SPs were removed via the remove operator. Finally, a novel improved simulated annealing algorithm called ISA was adopted to address association between SPs and UAVs and design UAVs' trajectories aiming to reduce their hovering and flying EC. The simulation results on different numbers of MIoTDs have validated that SATOA provides better results than the compared variants in terms of reducing the mean EC. In the future, we will study some hardware implementations of the proposed algorithm in UAV-assisted MFC systems.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

The studies at St. Petersburg State University of Telecommunications. prof. M.A. Bonch-Bruevich were supported by the Ministry of Science and High Education of the Russian Federation by the grant 075-15-2022-1137.

Data Availability

The data used to support the findings of this study are available from the authors upon request.

References

References

[1] A. S. Panayides, A. Amini, N. D. Filipovic, A. Sharma, S. A. Tsaftaris, A. Young, D. Foran, N. Do, S. Golemati, T. Kurc, K. Huang, K. S. Nikita, B. P. Veasey, M. Zervakis, J. H. Saltz, C. S. Pattichis, Ai in medical imaging informatics: Current challenges and future directions, IEEE Journal of Biomedical and Health Informatics 24 (7) (2020) 1837–1857. doi:10.1109/JBHI.2020.2991043.

[2] S. Rajoria, K. Mishra, A brief survey on 6g communications, Wireless Networks 28 (06 2022). doi:10.1007/s11276-022-03007-8.

[3] M. Asim, W. K. Mashwani, S. B. Belhaouari, S. Hassan, A novel genetic trajectory planning algorithm with variable population size for multi-uav-assisted mobile edge computing system, IEEE Access 9 (2021) 125569–125579. doi:10.1109/ACCESS.2021.3111318.

[4] M. Asim, Y. Wang, K. Wang, P. Q. Huang, A review on computational intelligence techniques in cloud and edge computing, IEEE Transactions on Emerging Topics in Computational Intelligence 4 (6) (2020) 742–763. doi:10.1109/TETCI.2020.3007905.

[5] M. Kumar, S. K. Mothku, K. Anusha, Deep reinforcement learning mechanism for deadline-aware cache placement in device-to-device mobile edge networks, Wireless Networks (10 2022). doi:10.1007/s11276-022-03135-1.

[6] A. Mohajer, F. Sorouri, A. Mirzaei, A. Ziaeddini, K. J. Rad, M. Bavaghar, Energy-aware hierarchical resource management and backhaul traffic optimization in heterogeneous cellular networks, IEEE Systems Journal (2022) 1–12doi:10.1109/JSYST.2022.3154162.

[7] A. Mohajer, M. S. Daliri, A. Mirzaei, A. Ziaeddini, M. Nabipour, M. Bavaghar, Heterogeneous computational resource allocation for noma: Toward green mobile edge-computing systems, IEEE Transactions on Services Computing (2022) 1–14doi:10.1109/TSC.2022.3186099.

[8] F. Nikjoo, A. Mirzaei, A. Mohajer, A novel approach to efficient resource allocation in noma heterogeneous networks: Multi-criteria green resource management, Applied Artificial Intelligence 32 (7-8) (2018) 583–612. doi:10.1080/08839514.2018.1486132.

[9] A. Zaini, L. Xie, Distributed drone traffic coordination using triggered communication, Unmanned Systems 08 (07 2019). doi:10.1142/S2301385020500016.

[10] M. Mozaffari, W. Saad, M. Bennis, Y. Nam, M. Debbah, A tutorial on UAVs for wireless networks: Applications, challenges, and open problems, IEEE Communications Surveys Tutorials 21 (3) (2019) 2334–2360.

[11] Z. Xiao, Y. Chen, H. Jiang, Z. Hu, J. C. Lui, G. Min, S. Dustdar, Resource management in uav-assisted mec: state-of-the-art and open challenges, Wireless Networks 28 (2022) 1–18.

[12] J. E. Low, L. T. S. Win, D. S. B. Shaiful, C. H. Tan, G. S. Soh, 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.

[13] Y. Zeng, R. Zhang, T. J. Lim, Wireless communications with unmanned aerial vehicles: opportunities and challenges, IEEE Communications Magazine 54 (5) (2016) 36–42. doi:10.1109/MCOM.2016.7470933.

[14] R. Gupta, A. Shukla, P. Mehta, P. Bhattacharya, S. Tanwar, S. Tyagi, 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. doi:10.1109/INFOCOMWKSHPS50562.2020.9162738.

[15] K. Gomez, A. Hourani, L. Goratti, R. Riggio, S. Kandeepan, 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. doi:10.1109/EuCNC.2015.7194055.

[16] A. Merwaday, I. Guvenc, UAV assisted heterogeneous networks for public safety communications, in: 2015 IEEE Wireless Communications and Networking Conference Workshops (WCNCW), 2015, pp. 329–334. doi:10.1109/WCNCW.2015.7122576.

[17] B. Zhang, G. Zhang, S. Ma, K. Yang, K. Wang, Efficient multitask scheduling for completion time minimization in UAV-assisted mobile edge computing, Mobile Information Systems 2020 (2020) 1–11.

[18] Z. Chen, X. Nan, D. Han, A multilevel mobile fog computing offloading model based on uav-assisted and heterogeneous network, Wireless Communications and Mobile Computing 2020 (2020) 11.

[19] P. Liu, H. He, T. Fu, H. Lu, A. Alelaiwi, M. W. I. Wasi, Task offloading optimization of cruising uav with fixed trajectory, Computer Networks 199 (2021) 108397. doi:https://doi.org/10.1016/j.comnet.2021.108397. URL https://www.sciencedirect.com/science/article/pii/S1389128621003741

[20] H. Wang, H. Ke, W. Sun, Unmanned-aerial-vehicle-assisted computation offloading for mobile edge computing based on deep reinforcement learning, IEEE Access 8 (2020) 180784–180798. doi:10.1109/ACCESS.2020.3028553.

[21] W. Lu, Y. Ding, Y. Gao, S. Hu, Y. Wu, N. Zhao, Y. Gong, Resource and trajectory optimization for secure communications in dual-uav-mec systems, IEEE Transactions on Industrial Informatics (2021) 1–1doi:10.1109/TII.2021.3087726.

[22] D. Han, T. Shi, Secrecy capacity maximization for a uav-assisted mec system, China Communications 17 (10) (2020) 64–81. doi:10.23919/JCC.2020.10.005.

[23] E. T. Michailidis, N. I. Miridakis, A. Michalas, E. Skondras, D. J. Vergados, D. D. Vergados, Energy optimization in massive MIMO UAV-aided MEC-enabled vehicular networks, IEEE Access 9 (2021) 117388–117403. doi:10.1109/ACCESS.2021.3106495.

[24] Y. Xu, T. Zhang, D. Yang, L. Xiao, Uav-assisted relaying and mec networks: Resource allocation and 3d deployment, in: 2021 IEEE International Conference on Communications Workshops (ICC Workshops), 2021, pp. 1–6. doi:10.1109/ICCWorkshops50388.2021.9473550.

[25] L. Yang, H. Yao, X. Zhang, J. Wang, Y. Liu, Multi-UAV deployment for MEC enhanced IoT networks, in: 2020 IEEE/CIC International Conference on Communications in China (ICCC), 2020, pp. 436–441. doi:10.1109/ICCC49849.2020.9238870.

[26] P. Huang, Y. Wang, K. Wang, Energy-efficient trajectory planning for a multi-UAV-assisted mobile edge computing system, Frontiers of Information Technology & Electronic Engineering 21 (12) (2020) 1713–1725. doi:10.1631/FITEE.2000315.

[27] Y. Zeng, J. Xu, R. Zhang, Energy minimization for wireless communication with rotary-wing UAV, IEEE Transactions on Wireless Communications 18 (4) (2019) 2329–2345.

[28] M. Asim, W. K. Mashwani, A. A. Abd El-Latif, Energy and task completion time minimization algorithm for uavs-empowered mec system, Sustainable Computing: Informatics and Systems 35 (2022) 100698. doi:https://doi.org/10.1016/j.suscom.2022.100698.

[29] W.-T. Li, M. Zhao, Y.-H. Wu, J.-J. Yu, L.-Y. Bao, H. Yang, D. Liu, Collaborative offloading for UAV-enabled time-sensitive MEC networks, EURASIP Journal on Wireless Communications and Networking 2021 (1) (2021) 1–17.

[30] C. Sun, W. Ni, X. Wang, Joint computation offloading and trajectory planning for UAV-assisted edge computing, IEEE Transactions on Wireless Communications 20 (8) (2021) 5343–5358. doi:10.1109/TWC.2021.3067163.

[31] Q. Qin, E. Liu, R. Wang, Trajectory optimization for UAV assisted Fog-RAN network, in: H. Gao, Z. Feng, J. Yu, J. Wu (Eds.), Communications and Networking, Springer International Publishing, Cham, 2020, pp. 344–355.

[32] S. Zhang, S. Shi, S. Gu, X. Gu, Power control and trajectory planning based interference management for uav-assisted wireless sensor networks, IEEE Access 8 (2020) 3453–3464. doi:10.1109/ACCESS.2019.2962547.

[33] M. Asim, W. K. Mashwani, S. Habib, S. B. Belhaouari, An evolutionary trajectory planning algorithm for multi-UAV-assisted MEC system, Soft Computing (2021). doi:https://doi.org/10.1007/s00500-021-06465-y.

[34] M. Asim, A. A. Abd El-Latif, Intelligent computational methods for multi-unmanned aerial vehicle-enabled autonomous mobile edge computing systems, ISA Transactions (2021). doi:https://doi.org/10.1016/j.isatra.2021.11.021.

[35] K. András, Multiple traveling salesmen problem - genetic algorithm, using multi-chromosome representation, Retrieved from https://www.mathworks.com/matlabcentral/fileexchange/48133-multiple-traveling-salesmen-problem-genetic-algorithm-using-multi-chromosome-representation, MATLAB Central File Exchange (2021).

[36] A. Király, J. Abonyi, Redesign of the supply of mobile mechanics based on a novel genetic optimization algorithm using google maps api, Engineering Applications of Artificial Intelligence 38 (2015) 122–130.

[37] M. Asim, W. Khan, O. Yeniay, M. A. Jan, N. Tairan, H. Hussian, G.-G. Wang, Hybrid genetic algorithms for global optimization problems, Hacettepe Journal of Mathematics and Statistics 47 (3) (2018) 539–551.

[38] W. Khan Mashwani, A. Salhi, A decomposition-based hybrid multiobjective evolutionary algorithm with dynamic resource allocation, Applied Soft Computing 12 (9) (2012) 2765–2780. `doi:https://doi.org/10.1016/j.asoc.2012.03.067`.

[39] S. Kirkpatrick, C. Gelatt, M. Vecchi, Optimization by simulated annealing, in: M. A. Fischler, O. Firschein (Eds.), Readings in Computer Vision, Morgan Kaufmann, San Francisco (CA), 1987, pp. 606–615. `doi:https://doi.org/10.1016/B978-0-08-051581-6.50059-3`.

[40] V. Cerny, Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm, Journal of Optimization Theory and Applications 45 (1985) 41–51. `doi:https://doi.org/10.1007/BF00940812`.

[41] A. Király, J. Abonyi, Optimization of Multiple Traveling Salesmen Problem by a Novel Representation Based Genetic Algorithm, Springer Berlin Heidelberg, Berlin, Heidelberg, 2011, pp. 241–269.

[42] A. K. Jain, Data clustering: 50 years beyond k-means, Pattern Recognition Letters 31 (8) (2010) 651 – 666.

[43] H. K. Mostapha, Evolutionary data clustering in matlab, https://yarpiz.com/64/ypml101-evolutionary-clustering, Yarpiz (2015).

[44] K. Joseph, Traveling salesman problem - genetic algorithm, Retrieved from https://www.mathworks.com/matlabcentral/fileexchange/13680-traveling-salesman-problem-genetic-algorithm, MATLAB Central File Exchange (2021).