Abdel-Basset *et al.*
Journal of Cloud Computing (2024) 13:35
https://doi.org/10.1186/s13677-024-00594-z

Journal of Cloud Computing: Advances, Systems and Applications

RESEARCH **Open Access**

Multiobjective trajectory optimization algorithms for solving multi-UAV-assisted mobile edge computing problem

Mohamed Abdel-Basset[1], Reda Mohamed[1], Ibrahim M. Hezam[4], Karam M. Sallam[2,3*], Abdelaziz Foul[4] and Ibrahim A. Hameed[5*]

Abstract

The Internet of Things (IoT) devices are not able to execute resource-intensive tasks due to their limited storage and computing power. Therefore, Mobile edge computing (MEC) technology has recently been utilized to provide computing and storage capabilities to those devices, enabling them to execute these tasks with less energy consumption and low latency. However, the edge servers in the MEC network are located at fixed positions, which makes them unable to be adjusted according to the requirements of end users. As a result, unmanned aerial vehicles (UAVs) have recently been used to carry the load of these edge servers, making them mobile and capable of meeting the desired requirements for IoT devices. However, the trajectories of the UAVs need to be accurately planned in order to minimize energy consumption for both the IoT devices during data transmission and the UAVs during hovering time and mobility between halting points (HPs). The trajectory planning problem is a complicated optimization problem because it involves several factors that need to be taken into consideration. This problem is considered a multi-objective optimization problem since it requires simultaneous optimization of both the energy consumption of UAVs and that of IoT devices. However, existing algorithms in the literature for this problem have been based on converting it into a single objective, which may give preference to some objectives over others. Therefore, in this study, several multiobjective trajectory planning algorithms (MTPAs) based on various metaheuristic algorithms with variable population size and the Pareto optimality theory are presented. These algorithms aim to optimize both objectives simultaneously. Additionally, a novel mechanism called the cyclic selection mechanism (CSM) is proposed to manage the population size accurately, optimizing the number of HPs and the maximum function evaluations. Furthermore, the HPs estimated by each MTPA are associated with multiple UAVs using the k-means clustering algorithm. Then, a low-complexity greedy mechanism is used to generate the order of HPs assigned to each UAV, determining its trajectory. Several experiments are conducted to assess the effectiveness of the MTPAs with variable population size and cyclic selection mechanisms. The experimental findings demonstrate that the MTPAs with the cyclic selection mechanism outperform all competing algorithms, achieving better outcomes.

Keywords Multi-UAVs, Mobile edge computing, Trajectory planning, Non-dominated solutions, Pareto optimality, *K*-means clustering algorithm

*Correspondence:
Karam M. Sallam
KSallam@sharjah.ac.ae; karam.sallam@canberra.edu.au
Ibrahim A. Hameed
ibib@ntnu.no
Full list of author information is available at the end of the article

Introduction

Mobile edge computing (MEC) technology has recently emerged to enhance the delivery of services to end users, with a focus on minimizing response time, particularly for real-time applications, while also supporting mobility and location awareness [1]. However, IoT devices face limitations in terms of storage capacity and computing power, making them unsuitable for resource-intensive tasks. To address these limitations, MEC technology has been employed to provide storage and computing services to IoT devices, enabling them to execute resource-intensive IoT tasks with lower EC and reduced latency [2]. Nevertheless, the fixed location of the edge servers poses a significant constraint on the effectiveness of this technology, potentially hindering the provision of desired services to end users. To overcome this limitation, unmanned aerial vehicles (UAVs) have recently been employed in conjunction with edge servers to provide computing power for IoT devices [2]. However, careful planning of UAV trajectories is essential to minimize EC during data transmission for IoT devices, as well as reduce energy usage during UAV hovering and mobility between halting points (HPs). The trajectory planning problem (TPP) is a complex optimization challenge due to multiple factors that must be considered, including the number and positioning of HPs, the relationship between HPs and UAVs, and the sequencing of HPs for each UAV. The number of HPs should be variable during the optimization process because it is unknown a priori. This presents a significant difficulty for the traditional gradient-based techniques because a gradient vector is not explicitly defined [3]. Metaheuristic algorithms are the best alternative to solve this problem because they are gradient-free. Those algorithms optimize a population of particles or individuals, and each individual typically represents either an entire deployment (number of HPs and their positions) for the UAV or an HP in the entire deployment.

Several recent studies have proposed various approaches to enhance the performance of MEC systems utilizing UAVs. For instance, Zhang et al. [4] addressed the resource allocation problem in UAV-supported MEC systems, aiming to optimize the transmit power of the vehicle and the trajectory of the UAV in order to minimize the total EC. They formulated this problem as an optimization task and employed a search algorithm to find a near-optimal trajectory. In another study, Asim et al. [5] presented a trajectory planning approach based on evolutionary algorithms for determining the best trajectories for multiple UAVs in multi-UAV-assisted MEC systems. This approach consisted of four stages. Firstly, the genetic algorithm (GA) was used to search for the near-optimal deployment of halting points (HPs) for UAVs. Secondly, duplicated HPs were eliminated using a removal operator. Thirdly, a clustering algorithm based on differential evolution was employed to assign the HPs to different clusters. Finally, in the last stage, GA was utilized to establish the order of HPs for each UAV. These studies contribute to the improvement of MEC systems through UAV integration, with a focus on optimizing EC and trajectory planning.

Li et al. [6] proposed an optimization approach for jointly optimizing the resource allocation and UAV trajectory in the UAV-powered MEC systems. This approach based on the improved atomic orbital search (AOS) was used to minimize the total EC by jointly optimizing the transmit power allocation, CPU frequency allocation, time division, and flight trajectory. Wang et al. [7] proposed an optimization framework for the joint optimization of the geographical fairness between the end users (EU), the justice of each UAV's EU-load, and the EU's overall EC. This algorithm was based on multi-agent deep reinforcement learning to search for the optimal trajectory for each UAV separately. Savkin et al. [8] proposed a path-planning optimization technique for minimizing the overall EC and maximizing the number of computing tasks.

Asim et al. [9] proposed a trajectory planning technique based on GA with a variable population size (VPs) for minimizing the total EC of multi-UAV-aided MEC systems. This approach was comprised of two stages: the first stage included employing the GA with VPs for finding out the near-optimal deployment of HPs for UAV; and the second stage utilizing the multi-chrome GA to determine the relationship between UAVs and HPs, the UAV's near-optimal number, and the near-optimal order of HPs for UAVs. Zhang et al. [10] proposed an iterative optimization technique with the double-loop structure to search for jointly optimizing the partial computation offloading, power and spectrum resources, allocation of CPU, user association, and UAVs' trajectory to minimize the total EC and maximize the computation efficiency in MEC systems. Sun et al. [11] proposed an optimization approach based on the successive convex approximation for jointly optimizing the UAV's trajectory and the offloading mode in the hope of minimizing the overall EC.

Wu et al. [12] enhanced the tabu search algorithm and introduced a new robust path-planning approach. This approach efficiently optimized the number of UAVs and their path planning with the goal of minimizing total EC. In [2], a three-stage trajectory planning algorithm (TPA) was proposed to minimize the overall EC of UAVs. In the first stage, both the number and positions of halting points (HPs) were simultaneously updated using the differential evolution algorithm combined with a virtual particles (VPs) strategy.

In the second stage, the k-means clustering technique was applied to divide the provided HPs into a number of groups equal to the number of UAVs, with each group containing HPs visited by the same UAV. Finally, in the last stage, a low-complexity greedy strategy was employed to determine the order of HPs in each cluster, thereby generating the trajectory for each UAV. Additionally, there are several other studies in the literature that propose trajectory optimization approaches for UAV-assisted MEC systems.

M. J. Sobouti [13] proposed a new exact approach to solve the multiple 3D trajectory problem for flying base stations (FBS). This approach takes into consideration several constraints when used to solve this problem. Those constraints are FBS energy consumption, flight distance limits, inter-cell interference, and operation time. Broadly speaking, this approach has been split into two stages: FBS trajectory and FBS placement. In this approach, the problem is divided into numerous snapshots. First, the minimal number of FBSs needed and their precise 3D placements in each snapshot are estimated. Then, the trajectory phase is carried out between every two snapshots. During this phase, a binary linear model, which takes into account FBS flight distance and EC limits, is used to determine the ideal path between the origin and destination of each FBS. Afterwards, the shortest path for each FBS, which takes into consideration some constraints such as obstacles and collision avoidance, is computed. The experimental findings showed that this approach could be applied to the real world.

J. Lin [14] proposed a new encoding strategy, namely the cutting and padding encoding, to be employed with the differential evolution for presenting a multiobjective trajectory optimization technique to find the near-optimal flight trajectory of a UAV. This technique was verified using a set of up to 400 IoT devices and compared to some existing optimization techniques to reveal its effectiveness. In [15], several metaheuristic-based optimization algorithms, including the sine cosine algorithm (SCA), salp swarm algorithm (SSA), and flower pollination algorithm (FPA), were developed for precisely optimizing the whole deployment of a UAV utilizing an effective encoding technique. These algorithms were evaluated using numerous instances with a set of up to 700 IoT devices and compared to one another and to some other optimizers to demonstrate their performance. The experimental findings revealed that SCA outperformed in the majority of test instances.

Huang et al. [3] proposed a new encoding strategy for encoding solutions to this problem to avoid excessive dimensionality and mixed variables. This strategy was employed with differential evolution (DE) to offer a new deployment optimization technique, namely DEVIPS. This technique was evaluated using seven instances with several IoT devices ranging between 100 and 700. Furthermore, it was compared to several optimization approaches in order to demonstrate the superiority of this encoding scheme. Zhang [16] used an encoding mechanism to adapt the backtracking search algorithm (BSA) to solve this problem. This encoding mechanism makes each individual responsible for an HP and the population liable for the whole deployment. In addition, BSA was improved by utilizing opposition-based learning in conjunction with the population adjustment mechanism to present a better variant called BSADP. This approach was assessed using numerous instances and contrasted with some competing optimizers to demonstrate its superiority.

Zhang et al. [17] presented a new optimization approach based on integrating an elite-driven DE (EDDE) and DE with a dynamic population (DPDE). This approach was abbreviated EDDE-DPDE and used to optimize the deployment problem of a UAV with the purpose of saving the travel time required while collecting data from IoT devices. The experimental findings show the effectiveness of this algorithm compared to four competitors. Abu-Baker et al. [18] developed a new approach to gathering data from wireless sensor networks using UAVs. This approach incorporates two well-known metaheuristic algorithms, namely GA and particle swarm optimization (PSO). GA was utilized to solve the clustering problem, whilst the other was used to discover the UAV's near-optimal deployment. This approach's performance was measured using two metrics: throughput and lifetime. Some of these include drone swarm path planning [19], a multi-objective trajectory optimization algorithm for a single UAV [14], and reinforcement learning techniques [20–22]. These studies contribute to the field by offering various approaches for optimizing UAV trajectories in UAV-assisted MEC systems, with a focus on minimizing EC and improving overall efficiency.

The TPP in UAV-assisted MEC systems is considered a multi-objective optimization challenge since it requires optimizing the EC objectives of both UAVs and IoT devices. However, most algorithms proposed in the literature have approached this problem by converting it into a single-objective optimization, potentially favoring one objective over the other. To address this limitation and optimize both objectives simultaneously, this study presents several multi-objective trajectory planning algorithms (MTPAs). These MTPAs are developed by adapting various metaheuristic algorithms, including the artificial gorilla troops optimizer (GTO) [23], gradient-based optimizer (GBO) [24], teaching–learning-based optimization (TLBO) [25], nutcracker optimization algorithm (NOA) [26], slime mould algorithm (SMA) [27], spider wasp optimizer (SWO) [28], differential evolution (DE) [2], RUN [29], and INFO [30]. The MTPAs utilize a variable population size mechanism and leverage the Pareto optimality theory. Furthermore, a novel mechanism called the cyclic selection mechanism (CSM) is introduced in this study to effectively manage the population size, precisely optimize the number of halting points (HPs), and ensure accurate utilization of function evaluations. The HPs estimated by each MTPA are associated with multiple UAVs using the k-means clustering algorithm. The greedy mechanism is then employed to determine the order of HPs assigned to each UAV, thereby generating its trajectory. To evaluate the efficacy of the MTPAs with variable population size and cyclic selection mechanism, several experiments are conducted. The experimental results demonstrate that MTPAs employing the cyclic selection mechanism outperform other competing approaches.

The main contributions of this study are listed as follows:

- Presenting multiobjective trajectory planning algorithms (MTPAs) for finding the near-optimal trajectories for various UAVs in the UMEC systems
- Those MTPAs are based on several metaheuristic algorithms encoded using the variable population size and adapted for the multiobjective problems using the Pareto optimality.
- Presenting a novel mechanism, namely the cyclic selection mechanism, to manage the population size for optimizing the number of HPs for each UAV more accurately
- Investigating the effectiveness of the Pareto optimality and CSM with several metaheuristic algorithms for solving several instances with several IoT devices ranging between 80 and 400.
- The experimental findings show that dealing with this problem as a multiobjective could achieve better total EC than converting it into a single objective. Furthermore, the CSM could significantly improve the performance of the optimization algorithms
- Among all the studied algorithms, both MTPA-VTLBO and MTPA-VGBO could achieve superior outcomes.

The following is a summary of the remaining parts of this paper: Problem formulation section explains the problem formulation in a concise manner; the proposed algorithms are described in Proposed algorithm section; the experimental settings are illustrated in Experimental settings section. Results and discussion are presented in Results and discussions section, and the conclusion and recommendations for further research are discussed in Conclusion and future work section.

Problem formulation

The UAVs could be loaded with edge servers to be mobile for quickly receiving the data from the IoT devices for improving the quality of services presented to the end users. For example, Fig. 1 shows a multi-UAV-aided MEC system with n IoT devices and m UAVs. In this study, the IoT devices are denoted as $N = \{1, 2, \ldots, n\}$, and the m UAVs with edge servers are denoted as $M = \{1, 2, \ldots, m\}$. This system involves IoT devices that need to execute resource-intensive tasks. The *ith* task of the *ith* device is represented in this study using a two-tuple (D_i, S_i), where D_i represents the size of the *ith* task and S_i represents the required resources for processing a single bit in the *ith* task. The IoT devices could not process those tasks due to their limited resources. Therefore, those tasks are first submitted to the MEC servers to be processed, and then the findings are submitted back to the IoT devices.

The UAVs could update their stop positions to decrease the distance with the IoT devices; those positions are referred to as stop points or halting points (HPs). Each *jth* UAV has a set of k_j HPs, which is denoted as $K_j = \{1, 2, \ldots, k_j\}$. Furthermore, the *jth* UAV's trajectory is comprised of a sequence of HPs that are visited by UAVs for receiving the data from the IoT devices with the purpose of minimizing the total EC. In this study, the *jth* UAV's trajectory is denoted as $g_j = \left\{\left(X_j^l, Y_j^l, H_j^l\right)\right\}$, such that $l \in K_j$. The altitude H_j^l of each stop point for the trajectory of this UAV is set to a fixed value as discussed in [2]. The distance between the *ith* device, which is found at $(x_i, y_i, 0)$, and the *jth* HPs could be defined according to the following:

$$d_{ijl} = \sqrt{\left(X_j^l - x_i\right)^2 + \left(Y_j^l - y_i\right)^2 + \left(H_j^l - 0\right)^2} \quad (1)$$

where H_j^l is the fixed altitude of the *jth* UAV. The IoT devices always chose the nearest HP for sending their tasks to be processed to minimize EC and transmission time. To assign each IoT device to the nearest HPs, a binary variable b_{ijl} is used to determine whether the *ith* task is allocated to the *lth* HP of the *jth* UAV or not [3]. This variable b_{ijl} is assigned a value of 1 when the *lth* HP of the *jth* UAV is the nearest to the *ith* IoT device; otherwise, it is set to 0, as defined mathematically in the following mathematical equation:

$$C_1 : b_{ijl} = \begin{cases} 1, if\left(j, l\right) = \arg\ min_{j \in M, l \in K_j} d_{iji}, \\ 0,\ otherwise \end{cases} \quad (2)$$

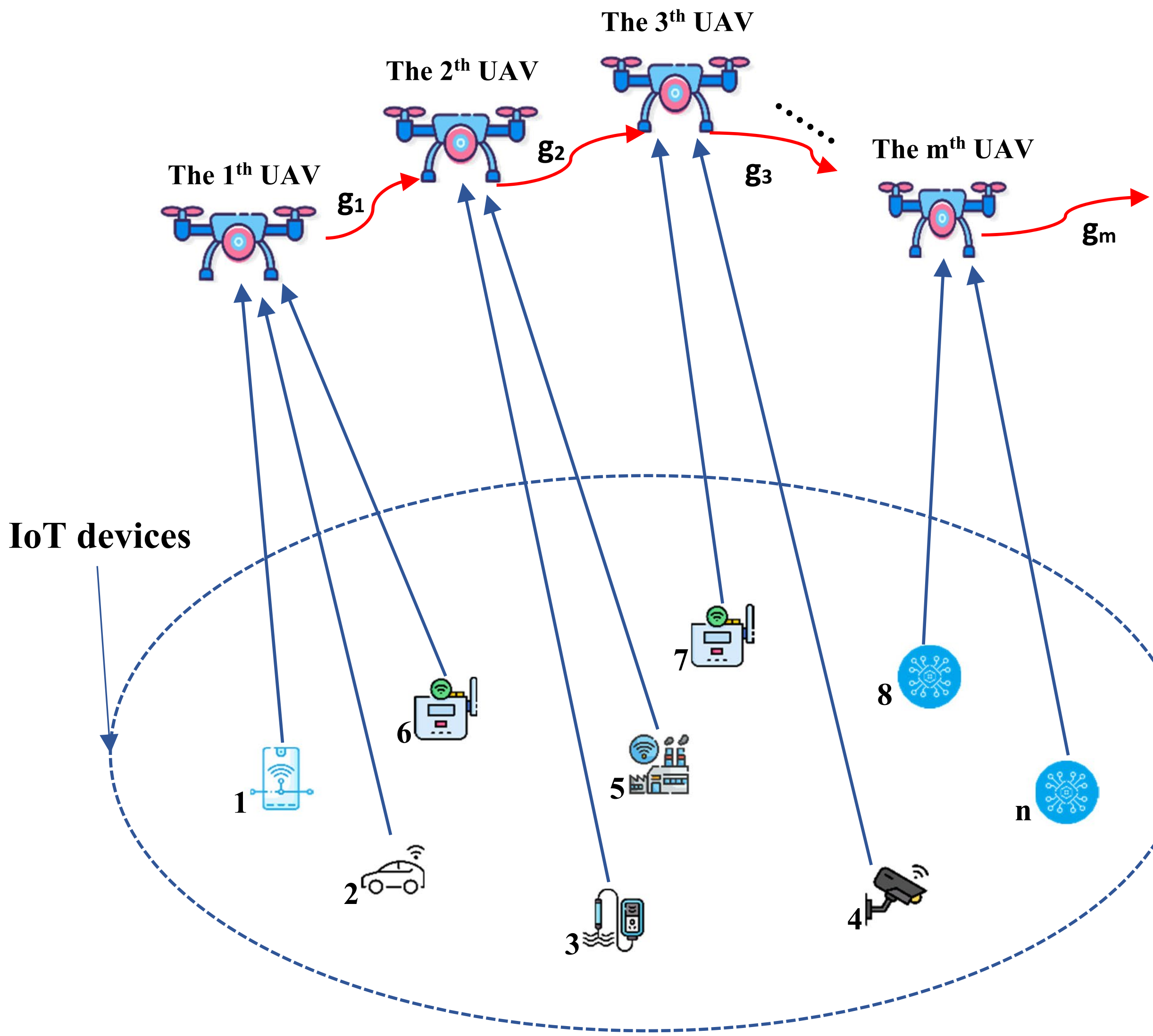

Fig. 1 Multi-UAV-aided MEC system including m UAVs and n IoT devices

The next constraint should be achieved to ensure that the *ith* task is assigned to only a UAV at only one HP:

$$C_2 : \sum_{j=1}^{m} \sum_{l=1}^{kj} b_{ijl} = 1, \forall i \in N \tag{3}$$

The *jth* UAV at the *lth* HP is able to simultaneously process the tasks of δ IoT devices at most due to its limited bandwidth, so the following constraint should be satisfied:

$$C_3 : \sum_{i=1}^{n} b_{ijl} \leq \delta, \forall j \in M, \forall l \in K_j \tag{4}$$

In addition to this, each UAV located at each HP provides service to at least one IoT device; hence, the total number of HPs for all UAVs, which is referred to as k, must adhere to the following constraint:

$$C_4 : k_{min} \leq k \leq k_{max} \tag{5}$$

where k_{min} is equal to $\left[\frac{n}{M}\right]$, and k_{max} is equal to n. During the data transmission, the channel power gain between the *ith* IoT device and the *jth* UAV at the *lth* HP could be defined as follows:

$$G_{ijl} = G_0 d_{ijl}^{-2} = \frac{G_0}{\left(X_j^l - x_i\right)^2 + \left(Y_j^l - y_i\right)^2 + {H_j}^2} \tag{6}$$

The mathematical model that could be used for computing the data transfer rate from the *ith* IoT device to the *jth* UAV at the *lth* HP is defined as follows:

$$R_{ijl} = B\log_2\left(1 + \frac{P_i G_{ijl}}{\beta^2}\right), \forall i \in N, j \in M, l \in K_j \quad (7)$$

where B refers to the bandwidth, P_i is the transmission power of the ith device, and β^2 indicates the white Gaussian noise power. Time taken and power used by the ith IoT device to deliver data to the jth UAV at the lth HP are calculated as [3]:

$$T_{ijl}^{iot} = \frac{D_i}{R_{ijl}}, \forall i \in N, j \in M, l \in K_j \quad (8)$$

$$E_{ijl} = P_i T_{ijl}^{iot} = \frac{P_i D_i}{R_{ijl}}, \forall i \in N, j \in M, l \in K_j \quad (9)$$

The total amount of energy consumed by all IoT devices is stated as [2]:

$$F1 = E_{iot} = \sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{l=1}^{k_j} b_{ijl} E_{ijl} \quad (10)$$

UAVs begin carrying out their missions after they have received the necessary input data. The amount of time it takes to complete the ith task on the jth UAV at the lth HP, given that the computing resource is c_{ijl}, may be calculated as:

$$T_{ijl}^{u} = \frac{S_i D_i}{c_{ijl}}, \forall i \in N, j \in M, l \in K_j \quad (11)$$

The jth UAV will wait until the lth HP has finished all of its assigned tasks before proceeding to the next HP. So, the amount of time the jth UAV spends hovering over the lth HP is equal to the total amount of time it takes to complete all assigned tasks. The hovering time of the jth UAV over the lth HP could be defined using the following equation:

$$T_{jl}^{u} = \max_{i \in N}\left\{ b_{ijl}\left(T_{ijl}^{u} + T_{ijl}^{iot}\right)\right\} \quad (12)$$

Afterwards, the energy consumption by the jth UAV over the lth HP until all assigned tasks are completed could be computed using the following formula:

$$E_j^u = \sum_{l=1}^{k_j} p^u T_{jl}^u, \forall j \in M \quad (13)$$

where p^u stands for the jth UAV's hovering power. The jth UAV will visit all HPs in the estimated trajectory g_j, so the time taken to fly among those SPs could be estimated as follows:

$$T_j^F = \frac{1}{v}\sum_{l=2}^{k_j}\sqrt{\left(X_j^l - X_j^{l-1}\right)^2 + \left(Y_j^l - Y_j^{l-1}\right)^2 + H_j^2}, \forall j \in M \quad (14)$$

where v represents the jth UAV's flight velocity. Then, the energy consumed by the jth UAV within the flight time could be estimated according to the following equation:

$$E_j^F = p^F T_j^F, \forall j \in M \quad (15)$$

where p^F represents the jth UAV's flight power. The total EC of all UAVs is based on the energy consumed in hovering time and flight time, as defined in the following formula:

$$F2 = E_{uav} = \sum_{j=1}^{m}\left(E_j^F + E_j^u\right) \quad (16)$$

In this paper, we seek to present a new optimization technique that could optimize the UAV trajectories to minimize the energy consumed by the UAVs and IoT devices in the multi-UAV-aided MEC system. The energy consumption of IoT and that of IoT devices is the total energy consumption of this system. This means that this problem contains two objectives that need to be simultaneously optimized. However, several works in the literature have dealt with this problem using a weighted variable to relate two objectives together. This strategy is not effective for optimizing both objectives together because some objectives might have the highest effect on the objective function, and hence the small improvement in this objective might give preference to the generated trajectory even though it affects negatively the other objectives with low influence. Finally, the mathematical model of the multiobjective trajectory planning problem (MTPP) could be defined as follows:

$$\begin{gathered} Min\ F = \{F1, F2\} \\ S.t. C_1 : b_{ijl} = \begin{matrix} 1, if\left(j, l\right) = \arg\ min_{j \in M, l \in k_j} d_{iji}, \\ 0,\ otherwise \end{matrix} \\ C_2 : \sum_{j=1}^{m}\sum_{l=1}^{k_j} b_{ijl} = 1, \forall 1 \in N \\ C_3 : \sum_{i=1}^{n} b_{ijl}\sum_{l=1}^{k_j} b_{ijl} \leq \delta, \forall j \in M, \forall l \in K_j \\ C_4 : k_{min} \leq k \leq k_{max} \\ C_5 : X_{min} \leq X_j^l \leq X_{max}, \forall j \in M, \forall l \in K_j \\ C_6 : Y_{min} \leq Y_j^l \leq Y_{max}, \forall j \in M, \forall l \in K_j \end{gathered}$$

Proposed algorithm

Pareto optimality theory

The trajectory planning optimization problem could be classified as a multiobjective optimization problem (MOP) because it contains two objectives that need to be simultaneously optimized. In addition, the MOP might involve a set of constraints that need to be met by the solutions obtained within the optimization process. For simplification, a multiobjective optimization problem that needs to be minimized is modeled as follows:

$$Min\ F(\overrightarrow{x_i}) = \{F1(\overrightarrow{x_i}), F2(\overrightarrow{x_i}), \ldots, FL(\overrightarrow{x_i})\}, L \geq 2 \tag{17}$$

$$Subject\ to\ nq_i\ \ (\overrightarrow{x_i}) \geq 0, i = 1, 2, 3, \ldots, z \tag{18}$$

$$q_i(\overrightarrow{x_i}) = 0, i = 1, 2, 3, \ldots, S \tag{19}$$

$$\overrightarrow{L} \leq \overrightarrow{x_i} \leq \overrightarrow{U}, i = 1, 2, 3, \ldots, n_1 \tag{20}$$

where L refers to the number of objectives in the tackled MOP,nq_i is the ith inequality constraint, z is the number of the inequality constraint, q_i is the ith equality constraint, S is the number of the equality constraint, n_1 is the number of solutions obtained by an optimization algorithm, $\overrightarrow{U}$ is the upper bound of all dimensions, and $\overrightarrow{L}$ is the lower bound. The multi-objective optimization problems are more complicated than the single-objective problems (SOPs) because the solutions obtained for these problems might improve some objectives and deteriorate others, and hence the methodology for choosing the best solution is hard to be achieved easily. In contrast, SOP only has one objective that the optimization algorithms could directly optimize in order to find a single solution that could either minimize or maximize it depending on its nature. From that, it is clear that SOP could be easily and directly solved by the majority of the optimization techniques, while for a MOP, the optimization technique needs to be first adapted by Pareto optimality to make them relevant for optimizing all objectives simultaneously. As aforementioned, the trajectory planning problem has two objectives that need to be simultaneously optimized using the Pareto optimality theory. This theory is based on estimating a set of solutions, known as non-dominated solutions. Those solutions are called nondominated because they could minimize at least one objective without deteriorating any one of the others [31]. For more clarification, in Fig. 2, the solution $\overrightarrow{x}_1$ is dominated by the solution $\overrightarrow{x}_2$ because it has one objective that is inferior and one objective that is equal. For solving the multiobjective trajectory planning problem of the UAVs in the multi-UAV-assisted MEC systems, a metaheuristic algorithm will be executed to generate a set of solutions, which are compared with each other under this theory, and the non-dominated solutions are stored in an archive to be compared with the solutions generated in the rest of the optimization process for searching for better non-dominated solutions.

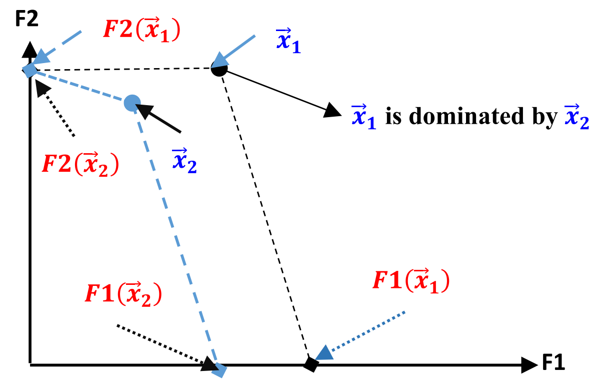

Fig. 2 An illustrative example for the non-dominated solutions

Association between HPs and UAVs

In this stage, the HPs generated by the optimization process at generation t need to be associated with the UAVs to determine the HPs for each UAV to be visited. To do that, according to [2], the K-means clustering (KMC) algorithm, which is classified as an unsupervised machine learning technique, is used to cluster the HPs into m clusters, denoted as C_j, where All HPs in the jth cluster are visited by only the jth UAV. In more detail, the KMC algorithm will first initialize m clusters with m HPs selected randomly from the current population; those HPs are used to represent the centers of the clusters. Then, the Euclidian distance is used to compute the distance between those centers and the other HPs in the current population and assign each HP to the cluster with the closest distance. In algorithm 1, we describe the steps of the KMC algorithm for assigning the different HPs to the best UAV based on computing the distance between each HP and the centroid of various clusters and assigning this HP to the nearest cluster. This process is continued until the termination condition is satisfied. The termination condition in this state is achieved when the centroid in each cluster is unchanged. It is worth stating that the centroid $\left(\check{X}_j, \check{Y}_j\right)$ for each cluster is derived by averaging the HPs allocated to it, as defined by the following formulas:

$$\check{Y}_j = \frac{1}{|C_j|} \sum\nolimits_{(X_l, Y_l) \in C_j} Y_l \tag{21}$$

$$\check{Y}_j = \frac{1}{|C_j|} \sum\nolimits_{(X_l, Y_l) \in C_j} X_l \tag{22}$$

Initialize $C_j, \forall j \in M$ with a HP selected randomly from the current HPs
while ($Termination\ condition\ is\ false$)
  for $i = 1: k$
    for $j = 1: m$
      Compute the Euclidian distance d_{ij} between the ith HP and the centroid of all HPs in the jth cluster.
    end for
    J'=$\arg\min_{j \in M} d_{ij}$
    Add the ith HP to the $J'th$ cluster ($C_{J'}$)
  end for
end while
Output: Return $C_{J'}$

Algorithm 1 The pseudocode of KMC algorithm

Low-complexity greedy mechanism for the order of HPs

In [2], a new simple greedy method for generating the order of HPs for each UAV was proposed to be a strong alternative to the traditional and modern optimization techniques, which have high computational costs. This method was called the low-complexity greedy method. For each UAV, this method selects an HP randomly from its corresponding cluster and then sets this HP to the first cell in the trajectory array of this UAV. Afterwards, the distance between this HP and all HPs in the corresponding cluster is computed, and the nearest HP is set to the second cell in the trajectory array. Likewise, the distance between the last added HP to the trajectory and all remaining HPs in the corresponding cluster is computed, and the nearest HP is added to the next cell in the trajectory array. This procedure is continued until all HPs in all clusters are assigned to the trajectory of each UAV. The pseudocode of this mechanism is presented in Algorithm 2.

Algorithm 2 The pseudocode of greedy mechanism

for $j = 1: m$
  Select an HP randomly from C_j
  Add this HP to g_j as the first HP to be visited by the jth UAV
  for $i = 1: k_j$
    Compute the Euclidian distance d_{ij} between the last added HP to g_j and all the other HPs in C_j
    Find the nearest HP and add it to g_j
  end for
end for
Output: Return g_j

Encoding mechanisms

The encoding mechanisms significantly affect the performance of the optimization techniques when tackling a specific problem. The trajectory planning problem has different encoding mechanisms in the literature to encode a set of HPs, which could minimize the energy consumed by UAVs in the MEC system. For example, some works encode the solutions to this problem using a population of individuals, where each individual is responsible for the HPs that construct a possible trajectory for UAVs. However, assume that there are k HPs, to encode the solutions that could estimate those HPs, each solution is comprised of $2k$ dimensions. Hence, under this encoding mechanism, it is hard to solve this problem due to the high dimensionality, especially with increasing the number of HPs. Also, this mechanism involves an additional array of the same length of the HPs to determine whether the corresponding HP is considered in the trajectory or not. In brief, this mechanism is ineffective because its performance significantly deteriorates with increasing the number of HPs. Therefore, in [2], an alternative encoding mechanism was proposed to represent the solutions to this problem more effectively. This mechanism is called the variable population size (VIPS)-based encoding mechanism, which is discussed in detail in the following section.

VIPS mechanism [3]

To better encode the solutions to the trajectory planning problem, this mechanism was devised to avoid the high dimensionality problem in addition to removing the need for auxiliary variables. This mechanism makes each individual responsible for the coordinate (x_i, y_i) of a HP, and the entire population represents all possible HPs for the UAVs. According to that, each solution is comprised of only two dimensions, as depicted in Fig. 3. Each solution is responsible for the location of HPs; however, their number is not taken into consideration even now. To handle that, in [3], the authors presented a novel idea based on using removing, insertion, and replacing operators to create three new populations, namely P_1, P_2, and P_3. P_1 includes all the individuals in the current population P, except for only one individual that is added to it from the newly-generated population Q. This means that the first population is increased one by one in the hope of achieving the near-optimal number of HPs. The second population, P_2, is first assigned to all individuals in P, and then an individual from it is randomly selected and replaced with the current individual from the newly generated population. This means that the number of HPs is kept unchanged while their locations are optimized. The third population copies all the

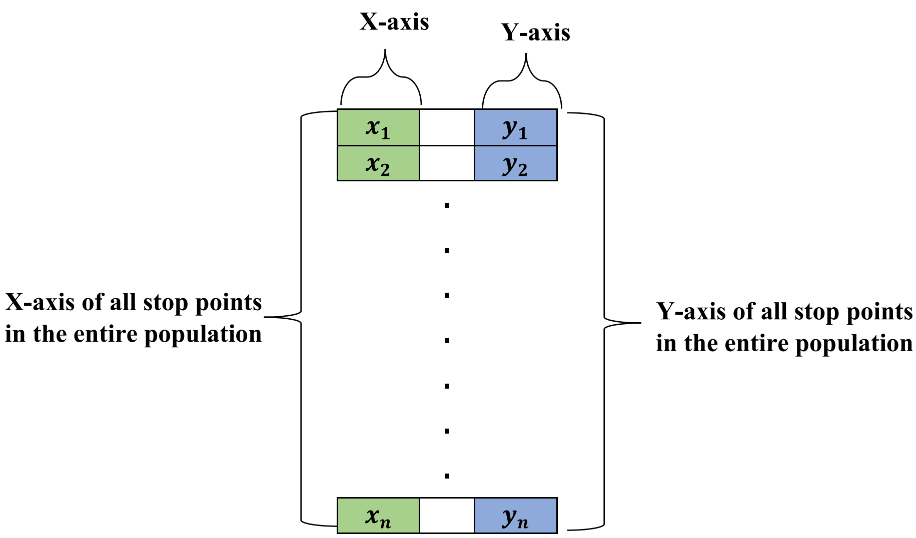

Fig. 3 Solution representation for TPP under the VIPS-based encoding mechanism

individuals from P, and then an individual is randomly selected and removed. This means that the number of stop points is decreased. Based on that, under those three operators, the number of HPs is increased, kept unchanged, or decreased; hence, the near-optimal number for HPs might be achieved. For simplification, those three populations are further discussed in brief in the following list:

- First, $P_1 = P$, then the *ith* solution in $\overrightarrow{x^t}$ is added to it. All individuals in P_1 are the same as those of P, except for only one solution.
- First, $P_2 = P$, then a solution from it is randomly chosen and replaced with the *ith* solution from $\overrightarrow{x^t}$.
- First, $P_3 = P$, then an individual is randomly selected and removed.

Cyclic selection-based encoding mechanism

However, the VIPS-based encoding scheme still needs strong improvement due to creating and evaluating three newly generated populations in each loop. This might require several function evaluations to achieve the required outcomes. Therefore, in this study, we present a novel idea for selecting only one population to be evaluated in each loop in the hope that this population is the best among the others. This idea is based on proposing a new factor, known as the cyclic factor ↕, which starts with a high probability for a tradeoff between the insertion and replacing operators and gradually decreases over time to maximize the probability of the removing operator and minimize the probability for the other two operators. That factor is reset a number of times within the whole optimization process to cover the three operators more accurately. For simplification, our idea is herein based on proposing a novel selection mechanism, namely cyclic selection mechanism (CSM), to accurately tradeoff between insertion, removing, and replacing operators for generating a new population based on a cyclic factor for covering all possible probabilities for each operator within the whole optimization process. In addition, under this mechanism, the operator that could achieve a better solution in the current generation is considered for generating a new population in the next generation in the hope of accelerating the convergence speed. The mathematical model for the cyclic factor is defined as:

$$\ell = \left(1 - \left(\frac{\boldsymbol{t}\%\frac{\boldsymbol{T_{max}}}{\boldsymbol{T}}}{\frac{\boldsymbol{T_{max}}}{\boldsymbol{T}}}\right)\right) \tag{23}$$

where $\boldsymbol{t}$ stands for the current function evaluation, $\boldsymbol{T_{max}}$ represents the maximum function evaluation, $\boldsymbol{T}$ stands for the number of cycles (Estimated in the experiments section), and **%** stands for the remainder operator. The pseudocode of CSM is described in Algorithm 3.

Algorithm 3 The pseudocode of CSM

```
Input: P, Q, T
b=0, a Boolean variable to determine if the last operator could achieve
    better findings.
Io = −1, including the selected operator
for i = 1 to n
    if Io ≠ 0%% Using the last operator to generate a new pop
        if Io == 1
            P̂: Generating the first population P1
        else if Io == 2
            P̂: Generating the second population P2
        else
            P̂: Generating the third population P3
        end if
    else
        r1, r2, and r3: Three numbers selected at random in (0, 1)
        if r1 < ℓ
            if r2 < r3
                P̂: Generating the first population P1
                Io = 1, %% Insertion operator
            else
                P̂: Generating the second population P2
                Io = 2, %% Replacing operator
            end if
        else
            P̂: Generating the third population P3
            Io = 3, %% Removing operator
        end if
    end
    Applying KMC to associate the HPs in P̂ into m UAVs
    Applying the greedy method to generate each UAV's trajectory
    f3 =Evaluating the trajectories of various UAVs using F1 and F2
    if f3 dominates F(P)
        P = P̂
        Add P̂ to A whether it is non-dominated.
    else
        Io = 0, %% to use another operator to generate the next P̂
    endif
end for
Return P and A
```

Adaptation of some metaheuristics with VIPS mechanism for MTPP in the UMEC system

In this study, nine metaheuristic algorithms, including GTO [23], GBO [24], TLBO [25], NOA [26], SMA [27], SWO [28], DE [2], RUN [29], and INFO [30], are adapted using the Pareto optimality theory to present multiobjective trajectory planning algorithms (MTPA), namely MTPA-GTO, MTPA-GBO, MTPA-SWO, MTPA-DE, MTPA-TLBO, MTPA-RUN, MTPA-INFO, MTPA-SMA, and MTPA-NOA, for tackling this problem. In more detail, those algorithms first create a two-dimensional matrix of $n \times 2$, where n represents the population size, and 2 represents the dimension size or coordinate of each HP. This matrix is randomly initialized within the lower bound of 0 and upper bound of 1000 for both X_j and Y_j, and evaluated using $F1$ and $F2$. Then, it is added to the archive (A) to represent the non-dominated solution obtained so far since there is no other solutions to be compared with it. Afterwards, the optimization process of those algorithms is independently executed for generating new population (Q), which is employed with the current population (P) to generate three populations (P_1, P_2, P_3) according to the insertion, deletion, and removing operators. The HPs in each population from those are associated with m UAVs using the KMC algorithm and the best trajectory of the HPs for each UAV is constructed by the greedy method discussed above. Afterwards, the trajectories estimated from those populations for various UAVs in UMEC system populations are evaluated using the objectives $F1$ and $F2$, and compared with each other to identify their dominance. A non-dominated population from those three populations is set to P for using in the next generation to search for better solutions. In addition, the nondominated populations from those three populations are compared to those in A, and added to A those that are non-dominated. At the beginning of the optimization process, the studied algorithms will extensively explore the regions around the current population to avoid stagnation into local minima. Then, gradually, with increasing the current function evaluation, they will convert the exploration into exploitation for exploiting the regions around a solution selected randomly from A for accelerating the convergence speed. Those process is continued until the maximum number of function evaluations is satisfied. A general pseudocode of a MTPA based on one of the investigated metaheuristic algorithms for solving MTPP is presented in Algorithm 4.

Algorithm 4: MTPA

```
Create a matrix, namely P, of n × 2
Initialize randomly each individual x⃗i in P
Applying KMC algorithm to associate the HPs in x⃗i into m UAVs
Applying the greedy method to generate the trajectory for each UAV
Evaluating the trajectories of various UAVs using F1 and F2
Add P to the archive A as the non-dominated solution so far
t = 1; %% Function evaluation counter
while (t < t_max)
  Generating Q using one of the studied algorithm under the current population P
  for i = 1 to n
       P1 = [P; Qi]%% Generating the first population
       Applying KMC to associate the HPs in P1 into m UAVs
       Applying the greedy method to generate each UAV's trajectory
        f1 =Evaluating the trajectories of various UAVs using F1 and F2
 %% Generating the second population
       P2 = P
       I ← The index of a random individual from P
       P2,I = Qi %% Applying the replacing operator
       Applying KMC to associate the HPs in P2 into m UAVs
       Applying the greedy method to generate each UAV's trajectory
        f2 =Evaluating the trajectories of various UAVs using F1 and F2
 %% Generating the third population
       P3 = P
       I ← The index of a random individual from P
       Removing P3,I %% Applying the removing operator
       Applying KMC to associate the HPs in P3 into m UAVs
       Applying the greedy method to generate each UAV's trajectory
        f3 =Evaluating the trajectories of various UAVs using F1 and F2
        [P, A]=Calling Algorithm 5 to get the nondominated from those three population
         t = t + 3
  end for
  Removing dominated solutions from A
  %% Tradeoff between exploration and exploitation
   if r1 < (t / T_max)
       I ← The index of a random population from A
       P = A_I
   end
end while
Return A
```

Algorithm 5: GetNondominated [f_1, f_2, f_3]

```
Input: [f1, f2, f3], A
if f1 dominates f2
    if f1 dominates f3
        P = P1
        Add P1 to A if it is not dominated
    else if f3 dominates f1
        P = P3
        Add P3 to A if it is not dominated
    else
        if r1 < r2
            P = P1
        else
            P = P3
        end if
        Add P3, P1, or both to A if they are not dominated
    end if
else if f2 dominates f1
    if f2 dominates f3
        P = P2
        Add P2 to A if it is not dominated by any existing solution
    else if f3 dominates f2
        P = P3
        Add P3 to A if it is not dominated by any existing solution
    else
        if r1 < r2
            P = P2
        else
            P = P3
        end if
        Add P3, P2, or both to A if they are not dominated
    end if
else if f1 dominates f3
    if r1 < r2
        P = P1
    else
        P = P2
    end if
    Add P2, P1, or both to A if they are not dominated
else if f3 dominates f1
    if f3 dominates f2
        P = P3
        Add P3 to A if it is not dominated
    else if f2 dominates f3
        P = P2
        Add P2 to A if it is not dominated
    else
        if r1 < r2
            P = P2
        else
            P = P3
        end if
        Add P3, P2, or both to A if they are not dominated
    end if
else if f2 dominates f3
    if r1 < r2
        P = P2
    else
        P = P1
    end if
    Add P2, P1, or both to A if they are not dominated
else if f3 dominates f2
    if r1 < r2
        P = P1
    else
        P = P3
    end if
    Add P1, P3, or both to A if they are not dominated
else %% Three populations are nondominated
    Add P1, P2, P3, or three to A if they are not dominated
end if
Return P and A
```

Adaptation of some metaheuristics with CSM mechanism for MTPP in the UMEC system

In the previous section, we clarified how to adapt the metaheuristic algorithms using the Pareto optimality and VIPS-based encoding mechanism for tackling MTPP. In this section, we will clarify how to adapt those algorithms under the Pareto optimality and CSM for tackling the same problem. Those algorithms with CSM are named MTPA-GTO, MTPA-VGBO, MTPA-VSWO, MTPA-VDE, MTPA-VTLBO, MTPA-VRUN, MTPA-VINFO, MTPA-VSMA, and MTPA-VNOA. Likewise, these algorithms generate a $n \times 2$ matrix, where n represents the population size and 2 represents the dimension size or coordinate of each HP. This matrix is initialized

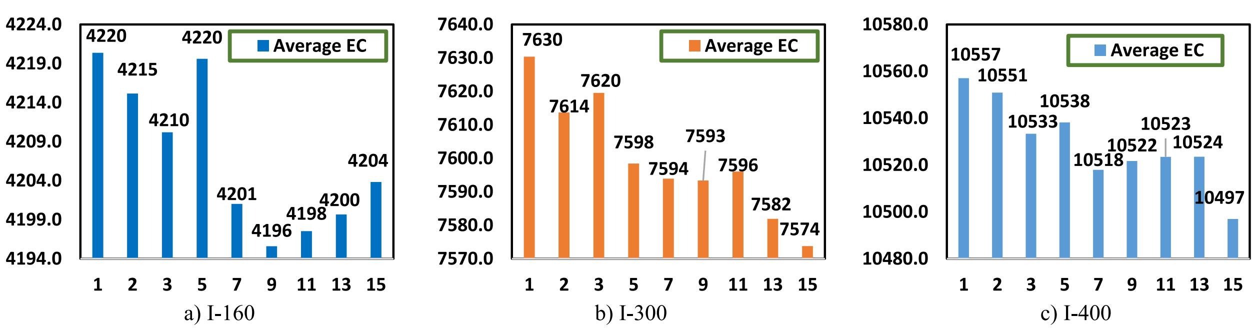

Fig. 4 Tuning the parameter *T* over three instances

Table 1 Comparison between MTPA-VDE, MTPA-DE, TPA-DE over SI

	MTPA-VDE				MTPA-DE					TPA-DE				
	BEC	**AEC**	**SD**	**FR**	**BEC**	**AEC**	**SD**	**PV**	**FR**	**BEC**	**AEC**	**SD**	**PV**	**FR**
I-80	**2.046E+06**	**2.084E+06**	**1.E+04**	**1.36**	2.047E+06	2.109E+06	3.E+04	**2.E-03**	2.20	2.071E+06	2.116E+06	3.E+04	**6.E-05**	2.44
I-100	**2.652E+06**	**2.690E+06**	**2.E+04**	**1.52**	2.653E+06	2.713E+06	4.E+04	**2.E-02**	2.12	2.651E+06	2.733E+06	4.E+04	**4.E-05**	2.36
I-120	2.766E+06	**2.817E+06**	**3.E+04**	**1.12**	**2.757E+06**	2.894E+06	6.E+04	**7.E-07**	2.32	2.831E+06	2.908E+06	7.E+04	**2.E-08**	2.56
I-140	**3.597E+06**	**3.686E+06**	**3.E+04**	**1.08**	3.667E+06	3.757E+06	4.E+04	**3.E-07**	2.24	3.699E+06	3.802E+06	1.E+05	**3.E-08**	2.68
I-160	**4.140E+06**	**4.229E+06**	**4.E+04**	**1.20**	4.235E+06	4.308E+06	4.E+04	**3.E-07**	2.40	4.245E+06	4.329E+06	1.E+05	**6.E-07**	2.40
I-180	**4.720E+06**	**4.801E+06**	**4.E+04**	**1.08**	4.802E+06	4.887E+06	5.E+04	**3.E-07**	2.16	4.849E+06	4.920E+06	4.E+04	**3.E-09**	2.76
I-200	**5.256E+06**	**5.352E+06**	**4.E+04**	**1.16**	5.345E+06	5.441E+06	6.E+04	**1.E-06**	2.28	5.362E+06	5.476E+06	8.E+04	**2.E-08**	2.56

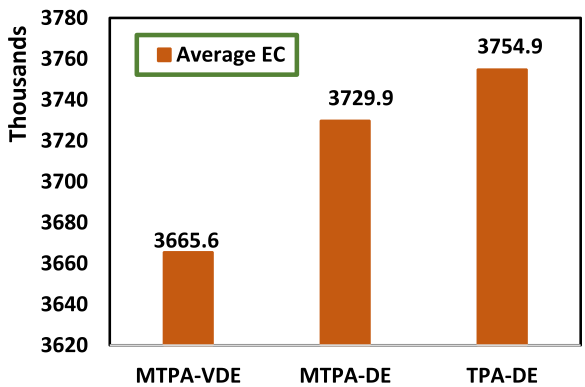

Fig. 5 Average EC value under SI between MTPA-VDE, MTPA-DE, and TPA-DE

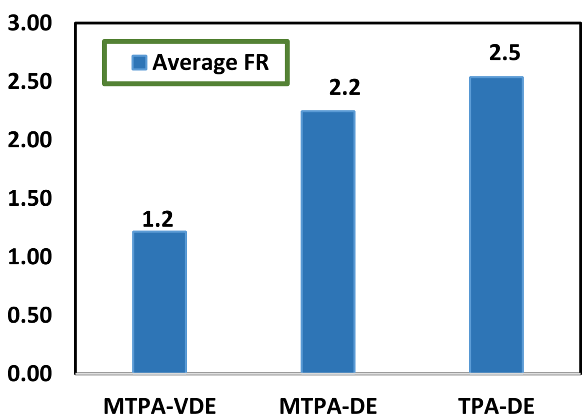

Fig. 6 Average FR under SI between MTPA-VDE, MTPA-DE, and TPA-DE

at random within the lower and upper bounds of each dimension and evaluated using $F1$ and $F2$. Then, it is added to the archive (A) to represent the non-dominated solution obtained thus far, as there are no other solutions against which it can be compared. The optimization process of these algorithms is then independently executed to generate a new population (Q), which is collaborated with the current population (P) under the insertion, removing, and replacing operators to generate a new population with variable population size to optimize the location and number of HPs for the UAVs in UMEC systems, as described in Algorithm 3. The nondominated populations estimated by CSM are added to A whether they are not dominated by any solution inside it. This procedure is repeated until the utmost number of function evaluations has been attained. Algorithm 6 presents a general pseudocode of an MTPA with CSM (namely MTPA-V) for solving MTPP.

Algorithm 6: MTPA-V

```
Create a matrix, namely P, of n × 2
Initialize randomly each individual x⃗_i in P
Applying KMC algorithm to associate the HPs in x⃗_i into m UAVs
Applying the greedy method to generate the trajectory for each UAV
Evaluating the trajectories of various UAVs using F1 and F2
Add P to the archive A as the non-dominated solution so far
t = 1; %% Function evaluation counter
while (t < t_max)
    Generating Q using one of the studied algorithms under the current population P
    [A, P]= Calling Algorithm 1 to execute CSM to combine Q with P under three considered operators
    t = t + n
    Removing dominated solutions from A
end while
Return A
```

Experimental settings

In this study, nine instances, which include various numbers of IoT devices of I-80, I-100, I-120, I-140, I-160, I-180, I-200, I-300, and I-400, are employed to investigate the performance of the proposed algorithms. Those devices are randomly scattered in a squared area of $1000m$, and four UAVs are flying at a velocity of $20m/s$ and a height of $200m$ to collect the data from those devices. The other parameters used in the MEC system assisted by multi-UAV are as follows:

Table 2 Comparison between MTPA-VDE, MTPA-DE, and TPA-DE over the E_{UAV} objective

	MTPA-VDE				MTPA-DE					TPA-DE				
	BEC	AEC	SD	FR	BEC	AEC	SD	PV	FR	BEC	AEC	SD	PV	FR
I-80	1.359E+02	1.360E+02	2.E-01	1.80	**1.358E+02**	**1.359E+02**	**3.E-01**	4.E-01	**1.40**	1.359E+02	1.364E+02	4.E-01	1.E-05	2.80
I-100	1.762E+02	1.765E+02	3.E-01	1.88	**1.762E+02**	**1.763E+02**	**1.E-01**	**5.E-04**	**1.28**	1.763E+02	1.771E+02	8.E-01	7.E-04	2.84
I-120	1.808E+02	1.810E+02	4.E-01	1.88	**1.808E+02**	**1.810E+02**	**7.E-01**	**3.E-02**	**1.60**	1.808E+02	1.814E+02	7.E-01	2.E-02	2.52
I-140	2.395E+02	2.397E+02	2.E-01	2.08	**2.395E+02**	**2.396E+02**	**1.E-01**	**2.E-03**	**1.40**	2.395E+02	2.403E+02	9.E-01	2.E-02	2.52
I-160	2.753E+02	2.755E+02	2.E-01	1.68	**2.753E+02**	**2.754E+02**	**8.E-02**	3.E-01	**1.52**	2.753E+02	2.761E+02	6.E-01	3.E-05	2.80
I-180	3.159E+02	3.161E+02	2.E-01	2.00	**3.159E+02**	**3.160E+02**	**7.E-02**	**6.E-03**	**1.32**	3.159E+02	3.169E+02	7.E-01	2.E-04	2.68
I-200	3.513E+02	3.515E+02	3.E-01	1.6	**3.513E+02**	**3.515E+02**	**4.E-01**	8.E-01	**1.56**	3.514E+02	3.523E+02	8.E-01	5.E-06	2.84

- M_i is randomly set between 1 and 10^3 MB
- P_i is assigned a value of $0.1W$
- p^u and p^F are assigned a value of $1000W$
- δ is assigned a number of 5
- B is set a value of $1MHz$
- G_0 is assigned a value of $-30dB$
- β^2 is assigned a value of $-174dBm$.
- $c_{ijl} = 10GHz$
- $S_i = 100$ cycles/bit
- D_i is scattered at random between 1 and 10^3 MB

Nine metaheuristic algorithms are chosen to investigate their performance for finding the best trajectories for four UAVs that could gather data from IoT devices in less EC; those algorithms include GTO [23], GBO [24], TLBO [25], NOA [26], SMA [27], SWO [28], DE [2], RUN [29], and INFO [30]. Those algorithms are adapted using the Pareto optimality theory, as discussed in detail in the section of the proposed algorithms, to present multiobjective variants for trajectory planning; those variants are termed MTPA-GTO, MTPA-GBO, MTPA-SWO, MTPA-DE, MTPA-TLBO, MTPA-RUN, MTPA-INFO, MTPA-SMA, and MTPA-NOA. All those algorithms are implemented in MATLAB R2019A within the same environments. The maximum number of function evaluations for all those algorithms is limited to 10,000 to ensure a fair comparison. All those algorithms are first executed 25 independent times on each instance and the obtained solutions are analyzed in terms of the summation indicator (SI) of two optimized objectives, the E_{UAV} objective, and the E_{iot} objective over several performance metrics, including best EC (BEC), average EC (AEC), p-value (PV) of the WRS test, Friedman mean rank (FR), and standard deviation (SD). The SI value of those objectives are computed in this study as follows [2]:

$$SI(\overrightarrow{x}_j^t) = E_{UAV} + \epsilon E_{iot} \tag{24}$$

where ϵ is set to 10,000 as discussed in [2], and $\overrightarrow{x}_j^t$ is the newly-generated deployment by an optimization algorithm.

Results and discussions

This section first compares the performance of the proposed MTPA-DE and MTPA-VDE to the recently published TPA to highlight the effectiveness of the Pareto optimality theory in optimizing all objectives simultaneously and reveal the performance of the proposed cyclic selection mechanism. Afterwards, the Pareto optimality theory and cyclic selection mechanism are integrated with eight additional metaheuristic algorithms to further observe their effectiveness in presenting a strong trajectory planning algorithm with a higher ability to minimize total EC.

Parameter adjustment

The proposed cyclic selection mechanism has an effective parameter that has to be accurately estimated to maximize its performance with the proposed algorithms. This parameter determines the number of times the removing, insertion, and replacing operators are applied. Several experiments have been conducted using various values for this parameter when applying MTPA-VGBO for solving three instances such as I-80, I-160, and I-300. The results of these experiments are analyzed using the FR metric and depicted in Fig. 4. Inspecting this figure shows that the performance of MTPA-VGBO is maximized when setting T to 15 for the instances I-300 and I-400. So, this value is considered in all experiments conducted later in this study.

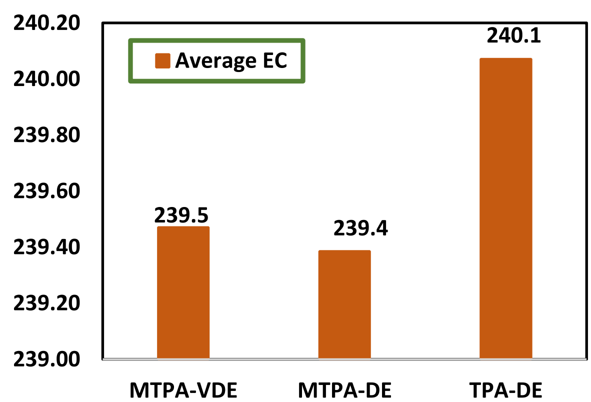

Fig. 7 Average value under the E_{UAV} objective for MTPA-VDE, MTPA-DE, and TPA-DE

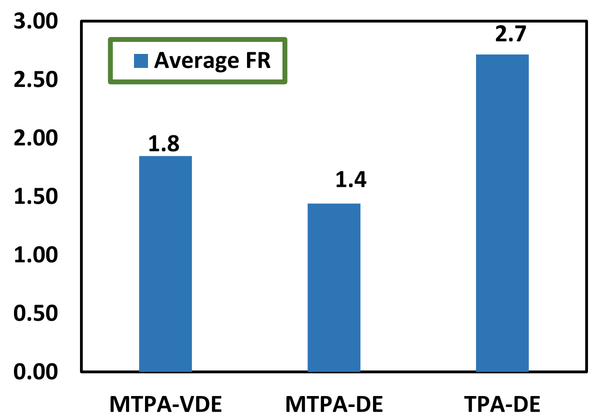

Fig. 8 Average FR under the E_{UAV} objective for MTPA-VDE, MTPA-DE, and TPA-DE

Table 3 Comparison between MTPA-VDE, MTPA-DE, and TPA-DE over the E_{iot} objective

	MTPA-VDE				MTPA-DE					TPA-DE				
	BEC	**AEC**	**SD**	**FR**	**BEC**	**AEC**	**SD**	**PV**	**FR**	**BEC**	**AEC**	**SD**	**PV**	**FR**
I-80	**6.871E+05**	**7.245E+05**	**2.E+04**	**1.28**	7.128E+05	7.570E+05	3.E+04	**9.E-05**	2.32	6.849E+05	7.573E+05	3.E+04	**1.E-04**	2.40
I-100	**8.886E+05**	**9.245E+05**	**2.E+04**	**1.52**	8.909E+05	9.500E+05	4.E+04	**1.E-02**	2.12	8.844E+05	9.785E+05	5.E+04	**2.E-05**	2.36
I-120	**9.472E+05**	**1.007E+06**	**3.E+04**	**1.12**	9.494E+05	1.084E+06	6.E+04	**5.E-07**	2.28	1.024E+06	1.111E+06	7.E+04	**1.E-08**	2.60
I-140	**1.201E+06**	**1.289E+06**	**3.E+04**	**1.08**	1.271E+06	1.361E+06	4.E+04	**2.E-07**	2.16	1.320E+06	1.427E+06	1.E+05	**3.E-09**	2.76
I-160	**1.386E+06**	**1.474E+06**	**4.E+04**	**1.2**	1.481E+06	1.553E+06	4.E+04	**2.E-07**	2.32	1.501E+06	1.594E+06	1.E+05	**4.E-08**	2.48
I-180	**1.589E+06**	**1.655E+06**	**3.E+04**	**1.12**	1.642E+06	1.727E+06	5.E+04	**2.E-06**	2.08	1.694E+06	1.781E+06	5.E+04	2.E-09	2.80
I-200	**1.771E+06**	**1.843E+06**	**3.E+04**	**1.16**	1.832E+06	1.927E+06	6.E+04	**1.E-06**	2.32	1.840E+06	1.983E+06	9.E+04	2.E-08	2.52

Comparison between MTPA-VDE, MTPA-DE, and TPA-DE

In this section, the proposed MTPA-VDE and MTPA-DE are compared to the recently published TPA-DE in terms of the SI, E_{UAV} objective, and E_{iot} objective to observe their effectiveness in finding the near-optimal deployment of HPs for multiple UAVs. Those algorithms have been executed 25 independent times on each instance, and the values of SI within those runs are analyzed in terms of several performance metrics and presented in Table 1. This table shows that MTPA-VDE is the best for all the performance metrics, except for BEC over I-120. In addition, this WRS test is applied to observe the difference between MTPA-VDE and the other algorithms. The resultant of this test, represented in the PV, is presented in the same table, which shows that the outcomes of MTPA-VDE are significantly different from the other algorithms for all instances. To summarize the outcomes of various algorithms over the SI, Figs. 5 and 6 are presented to compute the average of AEC and FR values over all instances. Those figures show that MTPA-VDE could rank first for the average AEC with a value of 3665.6 and the average FR with a value of 1.2, MTPA-DE ranks second with an AEC value of 3729 and FR value of 2.2, and TPA-DE is the worst algorithm.

The summation indicator sums the values of two optimized objectives without taking into consideration the performance of the algorithms for each objective independently. Therefore, Table 2 is presented to analyze the outcomes for the E_{UAV} objective obtained by each algorithm within 25 independent times. From this table, the proposed MTPA-DE could be the best for all performance indicators, followed by MTPA-VDE as the second high-performing algorithm, while TPA-DE is the worst. The average of both AEC and FR values presented in Table 2 for all instances is computed and reported in Figs. 7 and 8, respectively. Those figures show that MTPA-DE is the best, and MTPA-VDE is the second best. From those experiments, it is obvious that the performance of MTPA-DE is significantly better than that of TPA-DE and slightly competitive with the performance of MTPA-VDE. This reveals the significance of the Pareto optimality theory for optimizing all objectives simultaneously without giving higher preference to some objectives over others.

The outcomes of the E_{iot} objective obtained by each algorithm within 25 independent times are analyzed in Table 3. From this table, the proposed MTPA-VDE could be the best for all performance metrics, followed by MTPA-DE as the second high-performing algorithm, while TPA-DE is the worst. The average AEC and FR values shown in this table for all instances are calculated and displayed in Figs. 9 and 10, respectively. These figures confirm that MTPA-VDE is significantly superior to the other algorithms. Those experiments reveal the effectiveness of the Pareto optimality theory for simultaneously optimizing both E_{iot} and E_{UAV} objectives. To further show the effectiveness of MTPA-VDE and MTPA-VDE over TPA-DE, the convergence curves for each of them over all instances are depicted in Fig. 11. This figure reveals the high convergence speed of the proposed algorithms over TPA-DE. However, the proposed MTPA-VDE could be significantly superior to the MTPA-DE. According to the previous experiments, MTPA-VDE could achieve lower energy consumption than the other algorithms, so its flight trajectories might be shorter. To reveal that, Fig. 12 is presented to report the trajectories of each UAV over I-180 and I-200. This figure shows that MTPA-VDE could achieve shorter

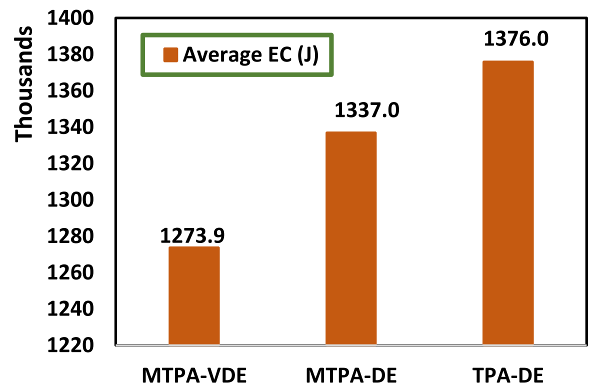

Fig. 9 Average value under the E_{iot} objective for MTPA-VDE, MTPA-DE, and TPA-DE

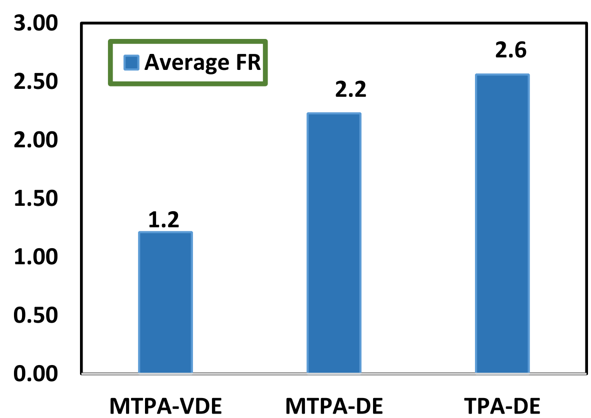

Fig. 10 Average FR under the E_{iot} objective for MTPA-VDE, MTPA-DE, and TPA-DE

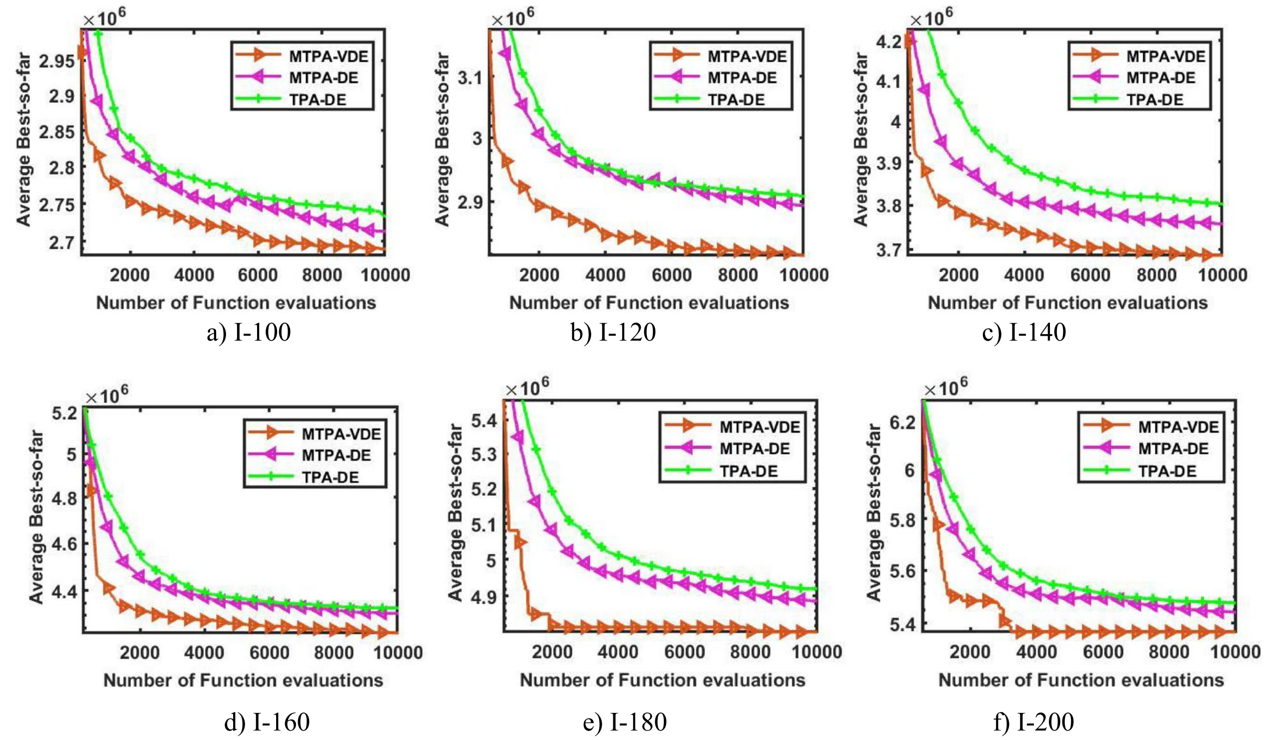

Fig. 11 Convergence curve under SI obtained by MTPA-VDE, MTPA-DE, and TPA-DE on some instances

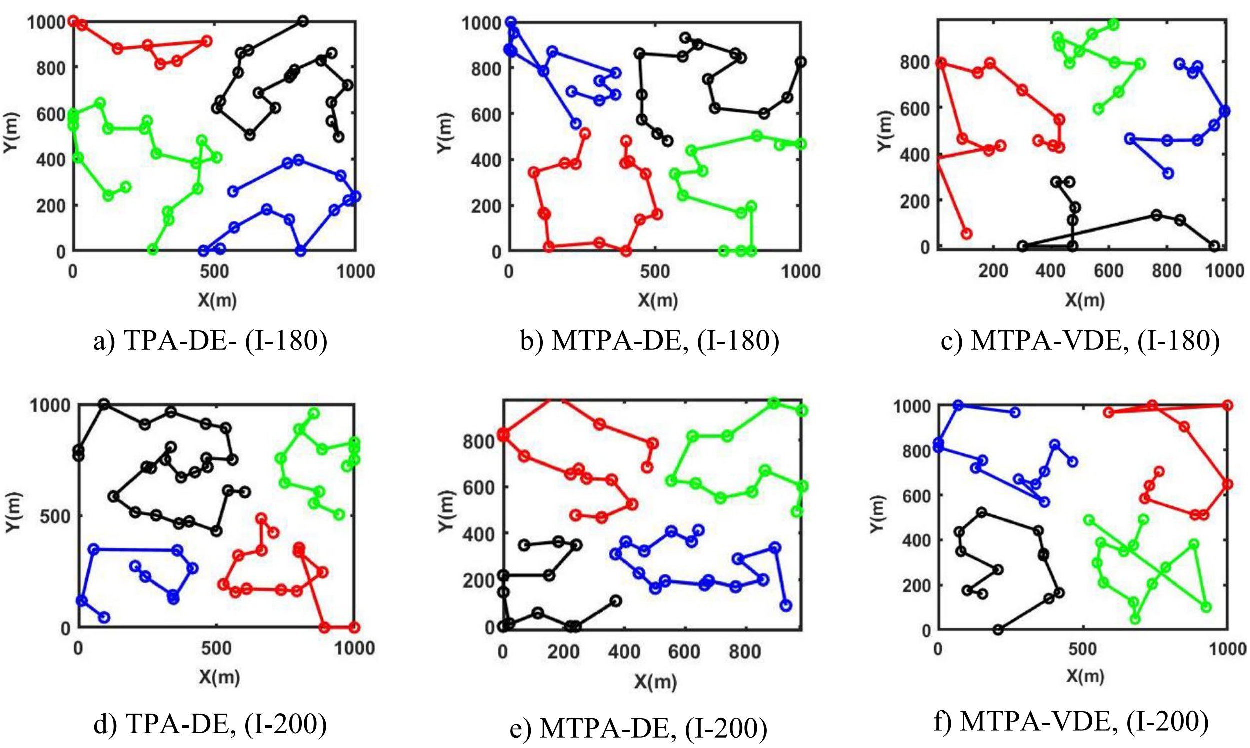

Fig. 12 Flight trajectories of various UAVs obtained by MTPA-VDE, MTPA-DE, and TPA-DE in some instances

Table 4 Comparison among nine multobjective metaheuristic algorithms over SI

		I-80				**I-100**					**I-120**				
	BEC	**AEC**	**SD**	**FR**	**PV**	**BEC**	**AEC**	**SD**	**PV**	**FR**	**BEC**	**AEC**	**SD**	**PV**	**FR**
MTPA-TLBO	2.05E+06	2.10E+06	3.E+04	4.2		2.64E+06	**2.70E+06**	4.E+04	3.8		**2.76E+06**	2.85E+06	3.E+04	4.6	
MTPA-GBO	2.04E+06	**2.09E+06**	**3.E+04**	**3.5**	3.E-01	**2.62E+06**	**2.70E+06**	**4.E+04**	**3.4**	6.E-01	2.79E+06	**2.84E+06**	**4.E+04**	**3.3**	4.E-02
MTPA-DE	2.07E+06	2.12E+06	3.E+04	6.2	2.E-02	2.67E+06	2.73E+06	3.E+04	5.9	1.E-02	2.78E+06	2.87E+06	5.E+04	5.5	5.E-01
MTPA-INFO	2.06E+06	2.10E+06	2.E+04	4.9	6.E-01	2.64E+06	**2.70E+06**	3.E+04	4.1	6.E-01	2.78E+06	2.85E+06	4.E+04	4.3	9.E-01
MTPA-NOA	2.05E+06	2.12E+06	4.E+04	5.6	6.E-02	2.68E+06	2.74E+06	3.E+04	6.8	3.E-04	2.81E+06	2.88E+06	4.E+04	5.8	5.E-02
MTPA-RUN	2.06E+06	2.11E+06	3.E+04	5.0	4.E-01	2.66E+06	2.72E+06	4.E+04	4.8	9.E-02	2.78E+06	2.87E+06	5.E+04	5.2	5.E-01
MTPA-SMA	2.08E+06	2.11E+06	2.E+04	5.6	7.E-02	2.68E+06	2.73E+06	2.E+04	5.7	2.E-03	2.78E+06	2.88E+06	5.E+04	5.4	2.E-01
MTPA-SWO	**2.03E+06**	**2.09E+06**	**4.E+04**	**3.5**	1.E-01	2.65E+06	2.71E+06	4.E+04	4.3	6.E-01	2.80E+06	2.86E+06	4.E+04	4.7	7.E-01
MTPA-GTO	2.04E+06	2.12E+06	4.E+04	6.3	1.E-02	2.68E+06	2.74E+06	3.E+04	6.2	7.E-04	2.79E+06	2.89E+06	4.E+04	6.2	1.E-02
		I-140				I-160					I-180				
MTPA-TLBO	3.66E+06	**3.72E+06**	**4.E+04**	**3.3**		4.22E+06	4.28E+06	3.E+04	4.2		4.78E+06	**4.84E+06**	**4.E+04**	**3.0**	
MTPA-GBO	3.64E+06	3.73E+06	3.E+04	3.8	3.E-01	4.19E+06	**4.27E+06**	**4.E+04**	**3.8**	4.E-01	4.78E+06	4.86E+06	5.E+04	3.8	2.E-01
MTPA-DE	3.69E+06	3.77E+06	4.E+04	6.0	5.E-04	4.20E+06	4.29E+06	5.E+04	4.6	5.E-01	4.79E+06	4.91E+06	8.E+04	5.6	5.E-04
MTPA-INFO	3.64E+06	3.74E+06	5.E+04	4.5	3.E-02	4.21E+06	4.28E+06	**4.E+04**	4.3	5.E-01	**4.75E+06**	4.87E+06	5.E+04	4.4	4.E-02
MTPA-NOA	3.65E+06	3.77E+06	6.E+04	5.4	6.E-03	4.19E+06	4.34E+06	7.E+04	6.8	8.E-04	4.83E+06	4.94E+06	8.E+04	6.9	2.E-06
MTPA-RUN	**3.63E+06**	3.75E+06	5.E+04	4.7	4.E-02	4.22E+06	4.29E+06	**4.E+04**	4.4	7.E-01	4.81E+06	4.88E+06	**4.E+04**	5.2	2.E-03
MTPA-SMA	3.70E+06	3.77E+06	5.E+04	6.3	1.E-04	4.23E+06	4.30E+06	**4.E+04**	5.6	8.E-02	4.81E+06	4.90E+06	6.E+04	5.7	4.E-04
MTPA-SWO	3.65E+06	3.74E+06	4.E+04	4.6	6.E-02	**4.18E+06**	4.29E+06	6.E+04	4.9	8.E-01	4.77E+06	4.87E+06	**4.E+04**	4.2	1.E-02
MTPA-GTO	3.69E+06	3.78E+06	5.E+04	6.4	3.E-05	4.25E+06	4.33E+06	5.E+04	6.4	5.E-04	4.82E+06	4.91E+06	6.E+04	6.2	4.E-05
		I-200				I-300					I-400				
MTPA-TLBO	5.33E+06	**5.40E+06**	**5.E+04**	**3.1**		**7.56E+06**	7.74E+06	9.E+04	4.0		**1.05E+07**	**1.07E+07**	**8.E+04**	**2.7**	
MTPA-GBO	**5.29E+06**	**5.40E+06**	5.E+04	3.2	7.E-01	7.61E+06	7.74E+06	1.E+05	**3.7**	6.E-01	**1.05E+07**	**1.07E+07**	1.E+05	3.7	3.E-01
MTPA-DE	5.37E+06	5.46E+06	5.E+04	5.6	4.E-04	7.62E+06	7.79E+06	1.E+05	5.4	5.E-02	1.06E+07	1.08E+07	1.E+05	5.2	4.E-04
MTPA-INFO	5.34E+06	5.45E+06	7.E+04	5.2	4.E-03	7.61E+06	7.77E+06	1.E+05	4.9	2.E-01	1.06E+07	1.07E+07	1.E+05	4.9	9.E-04
MTPA-NOA	5.39E+06	5.49E+06	6.E+04	6.7	4.E-06	7.69E+06	7.85E+06	1.E+05	6.9	6.E-04	1.06E+07	1.09E+07	2.E+05	7.8	4.E-08
MTPA-RUN	5.35E+06	5.45E+06	6.E+04	5.2	3.E-03	7.67E+06	7.77E+06	**7.E+04**	5.3	4.E-02	1.06E+07	1.07E+07	9.E+04	4.2	2.E-02
MTPA-SMA	5.35E+06	5.45E+06	5.E+04	5.1	2.E-03	7.63E+06	7.77E+06	**7.E+04**	5.2	9.E-02	1.06E+07	1.08E+07	1.E+05	5.9	1.E-04
MTPA-SWO	5.31E+06	5.42E+06	6.E+04	3.9	2.E-01	7.61E+06	**7.73E+06**	9.E+04	3.9	6.E-01	**1.05E+07**	1.07E+07	1.E+05	4.0	9.E-02
MTPA-GTO	5.34E+06	5.49E+06	8.E+04	7.0	2.E-05	7.61E+06	7.79E+06	8.E+04	5.6	1.E-02	1.06E+07	1.08E+07	1.E+05	6.6	3.E-06

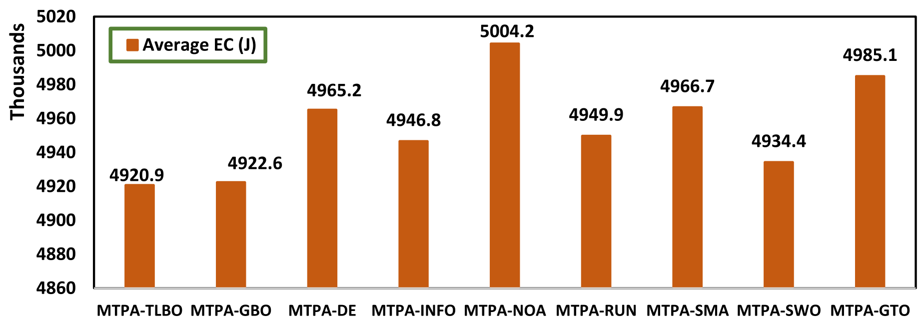

Fig. 13 Average EC under SI for nine multi-objective metaheuristic algorithms

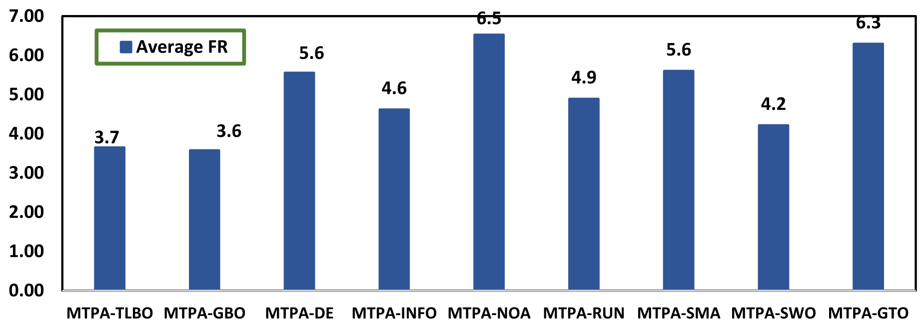

Fig. 14 Average FR under SI for nine multi-objective metaheuristic algorithms

flight trajectories for some UAVs and competitive flight trajectories for others.

Comparison among nine multiobjective algorithms with VIPS-based encoding scheme

In this study, eight additional metaheuristic algorithms are integrated with the Pareto optimality theory and VIPS-based encoding scheme to observe their performance for planning the UAV trajectories more efficiently. Those algorithms are executed 25 independent times and the SI values under two optimized objectives are computed according to Eq. (24). Those values are analyzed in terms of various performance metrics and reported in Table 4. This table shows that MTPA-GBO could achieve better FR values for five instances, and MTPA-TLBO is the best for the other instances. To summarize the findings in this table, Figs. 13 and 14 are presented to display the average of AEC and FR values for all instances, respectively. Inspecting those figures shows that MTPA-TLBO is the best-performing algorithm in terms of average AEC with a value of 4920.9, and MTPA-GBO is better than all in terms of average FR with a value of 3.6. From that, it is observed that both MTPA-GBO and MTPA-TLBO have competitive performance for updating the deployment of HPs for UAVs with the purpose of minimizing the total energy consumption.

To further observe the performance of those algorithms, the convergence curve obtained by each algorithm for each instance is computed and presented in Fig. 15. This figure shows that MTPA-GBO could converge faster than the others for I-100, I-120, I-160, and I-200; MTPA-TLBO could achieve better convergence speed for I-140, I-180, and I-400; and MTPA-SWO is better for the remaining instances.

Table 5 presents an analysis of the outcomes obtained by these algorithms for the first objective separately. This table demonstrates that MTPA-GBO could attain superior FR values in eight instances, while both MTPA-TLBO and MTPA-GBO are competitive in the remaining instances. The average AEC and FR values for all instances are depicted in Figs. 16 and 17, respectively, to provide a summary of the findings in Table 5. According to this figure, MTPA-GBO is the best-performing algorithm in terms of average

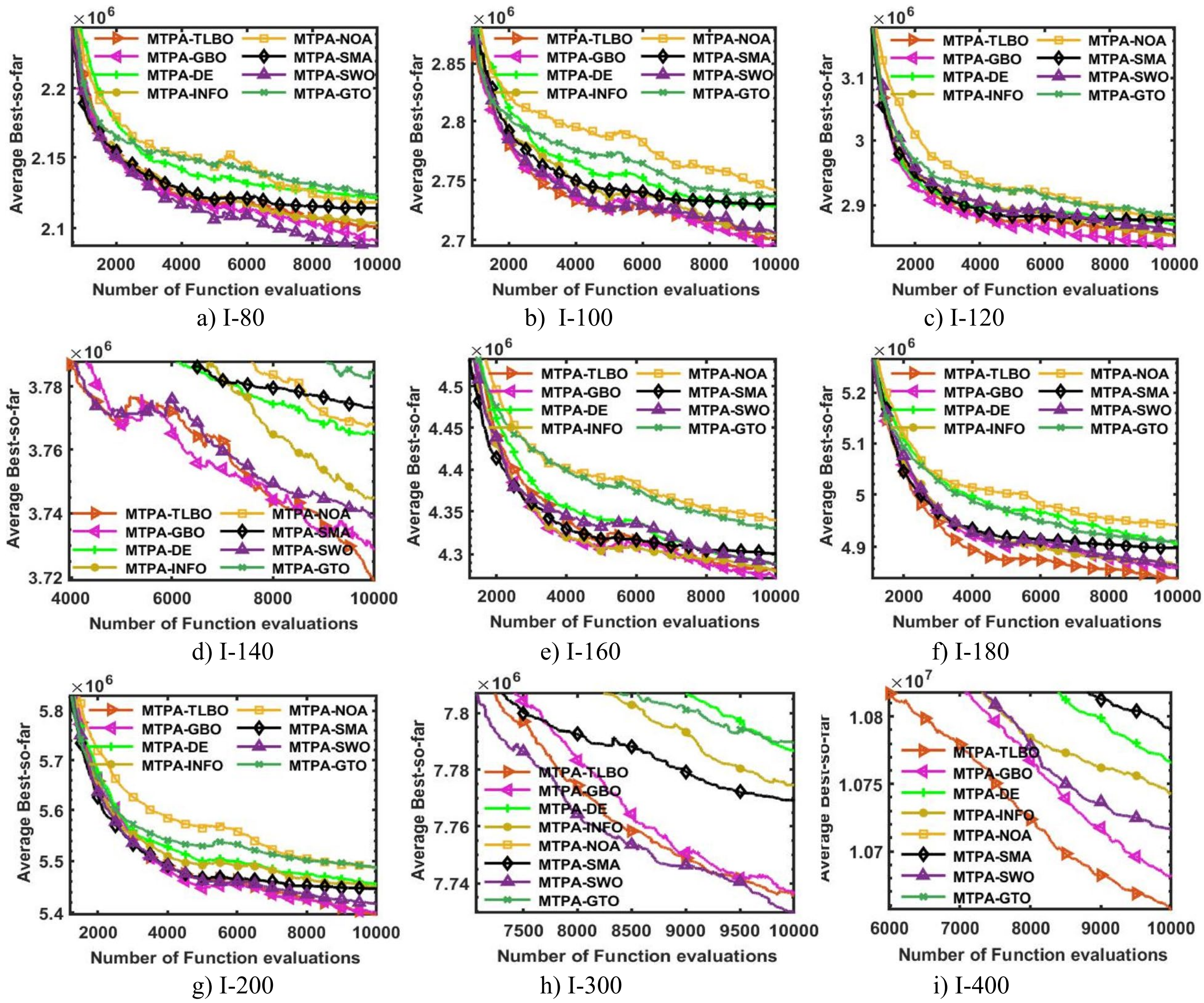

Fig. 15 Convergence curve under SI obtained by various multiobjective metaheuristic algorithms on all instances

AEC and FR, MTPA-TLBO is the second-best algorithm, and MTPA-SMA is classified as the worst-performing algorithm.

The analysis of the outcomes obtained by various algorithms for the E_{iot} objective is presented in Table 6. The presented table illustrates that MTPA-GBO exhibits higher FR values in six instances, MTPA-TLBO is the best for two other instances, and MTPA-SWO achieves outstanding outcomes for only one instance. Figures 18 and 19 present the mean values of AEC and FR, respectively, for all instances. These figures serve as a concise representation of the results stated in Table 6. Based on the data presented in the figure, it can be observed that the MTPA-GBO algorithm exhibits the highest performance in terms of FR, while the MTPA-TLBO algorithm exhibits the highest performance in terms of average AEC.

Comparison among some metaheuristic algorithms with CMS

The best five algorithms under Pareto optimality are integrated with the cyclic selection mechanism to further improve their performance for the trajectory planning of UAVs. These algorithms are executed 25 times independently, and the SI values for two optimized objectives are computed based on Eq. (24). Table 7 presents the results of an analysis of these values in terms of various performance metrics. This table

Table 5 Comparison of nine multiobjective metaheuristic algorithms over E_{UAV}

		I-80				**I-100**					**I-120**				
	BEC	**AEC**	**SD**	**FR**	**PV**	**BEC**	**AEC**	**SD**	**PV**	**FR**	**BEC**	**AEC**	**SD**	**PV**	**FR**
MTPA-TLBO	**135.85**	135.92	1.E-01	4.8		**176.21**	176.28	2.E-01	4.8		**180.78**	181.07	5.E-01	5.2	
MTPA-GBO	**135.85**	**135.86**	**3.E-02**	**3.3**	**3.E-02**	**176.21**	**176.25**	**9.E-02**	**3.4**	1.E-01	**180.78**	180.90	**4.E-01**	**3.3**	2.E-02
MTPA-DE	**135.85**	135.93	2.E-01	5.9	2.E-01	**176.21**	176.27	7.E-02	4.8	4.E-01	**180.78**	181.01	6.E-01	3.8	2.E-02
MTPA-INFO	**135.85**	135.89	1.E-01	4.0	7.E-01	**176.21**	176.28	2.E-01	4.4	7.E-01	**180.78**	**180.84**	9.E-02	4.5	8.E-02
MTPA-NOA	**135.85**	135.91	1.E-01	5.7	6.E-01	**176.21**	176.34	2.E-01	6.4	1.E-02	**180.78**	180.98	4.E-01	5.0	5.E-01
MTPA-RUN	**135.85**	135.99	2.E-01	6.6	2.E-02	**176.21**	176.37	5.E-01	5.3	3.E-01	**180.78**	181.17	7.E-01	5.4	8.E-01
MTPA-SMA	**135.85**	136.02	3.E-01	6.4	5.E-02	**176.21**	176.35	1.E-01	6.5	1.E-02	**180.78**	181.10	6.E-01	6.6	9.E-02
MTPA-SWO	**135.85**	135.91	1.E-01	4.1	4.E-01	**176.21**	176.24	5.E-02	4.3	4.E-01	**180.78**	181.29	9.E-01	5.6	9.E-01
MTPA-GTO	**135.85**	135.87	2.E-02	4.4	7.E-01	**176.21**	176.30	1.E-01	5.0	5.E-01	**180.78**	180.95	3.E-01	5.4	7.E-01
		I-140				I-160					I-180				
MTPA-TLBO	**239.48**	239.54	7.E-02	4.1		**275.31**	275.45	2.E-01	5.0		315.95	**315.98**	4.E-02	3.7	
MTPA-GBO	**239.48**	**239.52**	**6.E-02**	**3.5**	2.E-01	**275.31**	275.46	4.E-01	**3.8**	2.E-01	**315.94**	315.99	7.E-02	**3.2**	4.E-01
MTPA-DE	239.49	239.64	3.E-01	6.1	1.E-02	**275.31**	275.43	1.E-01	5.8	4.E-01	**315.94**	316.08	1.E-01	6.0	1.E-03
MTPA-INFO	**239.48**	239.54	8.E-02	4.4	8.E-01	**275.31**	**275.37**	8.E-02	4.0	2.E-01	**315.94**	316.05	1.E-01	5.0	1.E-02
MTPA-NOA	239.49	239.53	5.E-02	4.5	7.E-01	**275.31**	275.46	3.E-01	4.7	8.E-01	**315.94**	316.13	2.E-01	6.1	3.E-04
MTPA-RUN	**239.48**	239.73	6.E-01	6.2	3.E-02	**275.31**	275.47	2.E-01	5.9	3.E-01	**315.94**	316.05	1.E-01	5.6	7.E-03
MTPA-SMA	**239.48**	239.63	1.E-01	6.0	4.E-02	**275.31**	275.40	2.E-01	4.6	4.E-01	**315.94**	316.12	3.E-01	5.4	1.E-02
MTPA-SWO	**239.48**	239.55	7.E-02	4.9	5.E-01	**275.31**	275.40	1.E-01	4.8	5.E-01	**315.94**	316.01	5.E-02	4.6	7.E-02
MTPA-GTO	**239.48**	239.58	1.E-01	5.4	1.E-01	275.32	275.49	2.E-01	6.5	9.E-02	**315.94**	316.05	1.E-01	5.5	2.E-02
		I-200				I-300					I-400				
MTPA-TLBO	**351.30**	**351.36**	9.E-02	3.8		498.26	498.35	9.E-02	4.8		**696.33**	**696.41**	9.E-02	**4.1**	
MTPA-GBO	**351.30**	**351.36**	**2.E-01**	**2.7**	1.E-02	**498.25**	**498.31**	**8.E-02**	**3.2**	**3.E-02**	**696.33**	696.43	2.E-01	**4.1**	8.E-01
MTPA-DE	**351.30**	351.42	2.E-01	5.1	3.E-01	**498.25**	498.37	1.E-01	4.5	9.E-01	**696.33**	696.47	2.E-01	4.6	4.E-01
MTPA-INFO	**351.30**	351.56	5.E-01	5.6	1.E-01	**498.25**	498.40	1.E-01	5.2	3.E-01	**696.33**	696.48	2.E-01	4.9	8.E-02
MTPA-NOA	**351.30**	351.40	8.E-02	5.2	5.E-02	498.26	498.39	2.E-01	4.9	6.E-01	**696.33**	696.51	2.E-01	5.4	3.E-02
MTPA-RUN	351.32	351.43	9.E-02	6.1	7.E-05	**498.25**	498.43	2.E-01	5.4	4.E-01	**696.33**	696.46	1.E-01	4.9	6.E-01
MTPA-SMA	351.31	351.67	6.E-01	7.1	4.E-05	498.27	498.65	6.E-01	7.2	4.E-04	**696.33**	696.64	4.E-01	6.1	3.E-03
MTPA-SWO	**351.30**	351.43	3.E-01	3.8	4.E-01	**498.25**	498.36	1.E-01	4.5	9.E-01	**696.33**	696.46	2.E-01	4.4	8.E-01
MTPA-GTO	**351.30**	351.47	2.E-01	5.7	2.E-02	**498.25**	498.45	3.E-01	5.3	3.E-01	**696.33**	696.59	2.E-01	6.4	6.E-04

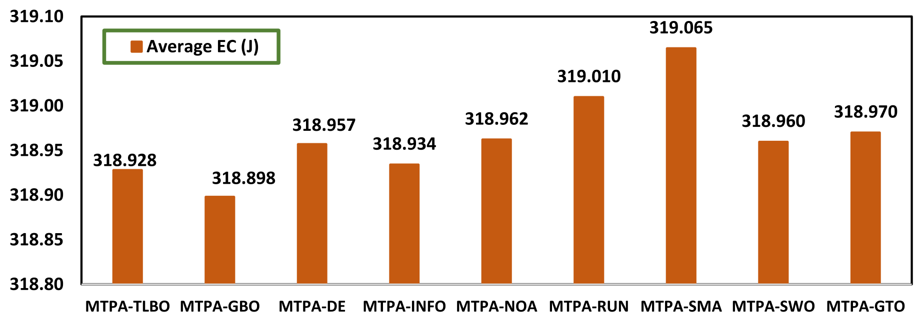

Fig. 16 Average EC value under the E_{UAV} objective for nine multi-objective metaheuristic algorithms

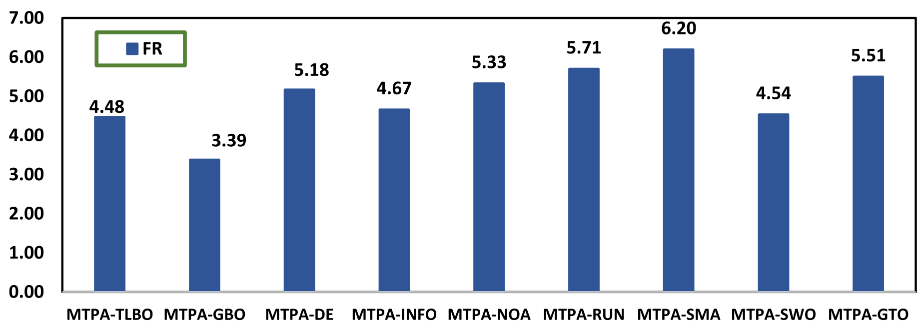

Fig. 17 Average FR value under the E_{UAV} objective for nine multi-objective metaheuristic algorithms

demonstrates that MTPA-VGBO could obtain higher FR values in five instances, MTPA-VINFO is superior in two instances, and both MTPA-VTLBO and MTPA-VSWO are superior in only one instance. The average AEC and FR values for all instances are displayed in Figs. 20 and 21, respectively, to summarize the data presented in this table. MTPA-VGBO has the highest average AEC with a value of 4843.9 and the highest average FR with a value of 2.51. To further examine the efficacy of these algorithms, Fig. 22 is presented to depict the convergence curves for each algorithm on each instance. This figure demonstrates that MTPA-VDE has the lowest convergence speed; however, the convergence speeds of the other algorithms are somewhat competitive.

To reveal the efficacy of these algorithms for each objective separately, Tables 8 and 9 are presented to report the outcomes obtained by each algorithm for each objective in various instances. Starting with the E_{UAV} objective, Table 8 presents the outcomes obtained by each algorithm for this objective. This table demonstrates that MTPA-VTLBO could achieve superior FR values in two instances; MTPA-VGBO, MTPA-VDE, and MTPA-VINFO are competitive in only one instance; both MTPA-VDE and MTPA-VSWO are superior in two instances; and both MTPA-VINFO and MTPA-VTLBO are superior in only one instance. In addition, the average AEC and FR values for all instances are depicted in Figs. 23 and 24, respectively, to provide a summary of the findings in Table 8. According to this data, MTPA-VTLBO is the best-performing algorithm in terms of average AEC, MTPA-VSWO is the second-best algorithm, and MTPA-VINFO is the worst-performing algorithm. For the E_{iot} objective, Table 9 reports the outcomes achieved by each algorithm for this objective. This table illustrates that MTPA-VGBO could achieve superior FR values in six instances, MTPA-VGBO and MTPA-VTLBO are competitive in only one instance, and MTPA-VINFO is superior in only one instance. In addition, to provide a summary of the findings in Table 9, Figs. 25 and 26 are presented to report the average AEC and FR values for all instances, respectively. According to this data, MTPA-VGBO is the most effective algorithm in terms

Table 6 Comparison of nine multiobjective metaheuristic algorithms over E_{iot}

		I-80				I-100					I-120				
	BEC	AEC	SD	FR	PV	BEC	AEC	SD	PV	FR	BEC	AEC	SD	PV	FR
MTPA-TLBO	6.94E+05	7.41E+05	3.E+04	4.3		8.81E+05	9.39E+05	4.E+04	3.8		9.49E+05	1.04E+06	4.E+04	4.4	
MTPA-GBO	6.85E+05	7.32E+05	3.E+04	3.6	3.E-01	**8.54E+05**	**9.33E+05**	**4.E+04**	**3.3**	6.E-01	9.86E+05	**1.03E+06**	**3.E+04**	**3.4**	**7.E-02**
MTPA-DE	7.06E+05	7.62E+05	3.E+04	6.2	1.E-02	9.08E+05	9.65E+05	3.E+04	5.9	1.E-02	9.73E+05	1.06E+06	5.E+04	5.4	4.E-01
MTPA-INFO	7.00E+05	7.45E+05	2.E+04	4.9	6.E-01	8.79E+05	9.42E+05	3.E+04	4.1	6.E-01	**9.69E+05**	1.04E+06	4.E+04	4.4	9.E-01
MTPA-NOA	6.84E+05	7.60E+05	4.E+04	5.7	6.E-02	9.16E+05	9.78E+05	3.E+04	6.6	**4.E-04**	9.97E+05	1.07E+06	4.E+04	5.7	5.E-02
MTPA-RUN	6.97E+05	7.47E+05	3.E+04	4.8	4.E-01	8.95E+05	9.54E+05	3.E+04	4.7	1.E-01	9.55E+05	1.06E+06	5.E+04	5.2	4.E-01
MTPA-SMA	7.21E+05	7.54E+05	2.E+04	5.4	1.E-01	9.12E+05	9.67E+05	2.E+04	5.6	3.E-03	9.71E+05	1.07E+06	5.E+04	5.5	1.E-01
MTPA-SWO	**6.71E+05**	**7.29E+05**	**4.E+04**	**3.4**	1.E-01	8.85E+05	9.44E+05	3.E+04	4.3	6.E-01	9.87E+05	1.05E+06	4.E+04	4.4	9.E-01
MTPA-GTO	6.77E+05	7.66E+05	4.E+04	6.3	1.E-02	9.22E+05	9.73E+05	3.E+04	6.1	**7.E-04**	9.85E+05	1.08E+06	4.E+04	6.2	1.E-02
		I-140				I-160					I-180				
MTPA-TLBO	1.26E+06	1.32E+06	4.E+04	3.4	0.E+00	1.46E+06	1.53E+06	3.E+04	4.2	0.E+00	1.62E+06	1.68E+06	4.E+04	3.0	0.E+00
MTPA-GBO	1.25E+06	**1.33E+06**	**3.E+04**	**3.8**	3.E-01	1.44E+06	**1.52E+06**	**4.E+04**	**3.7**	4.E-01	1.62E+06	**1.70E+06**	**5.E+04**	**3.8**	2.E-01
MTPA-DE	1.30E+06	1.37E+06	4.E+04	6.0	5.E-04	1.45E+06	1.53E+06	5.E+04	4.6	5.E-01	1.63E+06	1.75E+06	8.E+04	5.6	**6.E-04**
MTPA-INFO	1.25E+06	1.35E+06	5.E+04	4.5	2.E-02	1.46E+06	1.53E+06	**4.E+04**	4.3	6.E-01	**1.59E+06**	1.71E+06	5.E+04	4.4	5.E-02
MTPA-NOA	1.26E+06	1.37E+06	6.E+04	5.4	6.E-03	1.44E+06	1.58E+06	7.E+04	6.8	9.E-04	1.67E+06	1.78E+06	8.E+04	6.9	**1.E-06**
MTPA-RUN	**1.24E+06**	1.35E+06	5.E+04	4.7	4.E-02	1.46E+06	1.53E+06	**4.E+04**	4.4	7.E-01	1.65E+06	1.72E+06	4.E+04	5.1	3.E-03
MTPA-SMA	1.30E+06	1.38E+06	5.E+04	6.2	2.E-04	1.48E+06	1.55E+06	**4.E+04**	5.6	1.E-01	1.65E+06	1.74E+06	6.E+04	5.6	**6.E-04**
MTPA-SWO	1.26E+06	1.34E+06	4.E+04	4.6	6.E-02	**1.42E+06**	1.53E+06	6.E+04	4.9	8.E-01	1.61E+06	1.71E+06	4.E+04	4.2	2.E-02
MTPA-GTO	1.29E+06	1.39E+06	5.E+04	6.3	3.E-05	1.50E+06	1.58E+06	5.E+04	6.4	4.E-04	1.66E+06	1.75E+06	6.E+04	6.2	4.E-05
		I-200				I-300					I-400				
MTPA-TLBO	1.81E+06	**1.88E+06**	**5.E+04**	**3.1**		**2.57E+06**	**2.75E+06**	9.E+04	4.0		3.58E+06	**3.69E+06**	**8.E+04**	**2.7**	
MTPA-GBO	**1.77E+06**	1.89E+06	**5.E+04**	3.2	7.E-01	2.62E+06	**2.75E+06**	1.E+05	**3.7**	7.E-01	**3.50E+06**	3.72E+06	1.E+05	3.7	3.E-01
MTPA-DE	1.86E+06	1.94E+06	5.E+04	5.6	5.E-04	2.64E+06	2.80E+06	1.E+05	5.5	5.E-02	3.64E+06	3.80E+06	1.E+05	5.3	**5.E-04**
MTPA-INFO	1.82E+06	1.94E+06	7.E+04	5.2	5.E-03	2.63E+06	2.79E+06	1.E+05	4.9	2.E-01	3.68E+06	3.78E+06	1.E+05	4.9	**1.E-03**
MTPA-NOA	1.88E+06	1.97E+06	6.E+04	6.7	5.E-06	2.70E+06	2.87E+06	1.E+05	7.0	**6.E-04**	3.67E+06	3.94E+06	2.E+05	7.8	**4.E-08**
MTPA-RUN	1.84E+06	1.93E+06	6.E+04	5.3	4.E-03	2.68E+06	2.79E+06	**6.E+04**	5.3	4.E-02	3.63E+06	3.75E+06	9.E+04	4.2	**2.E-02**
MTPA-SMA	1.84E+06	1.93E+06	5.E+04	5.1	4.E-03	2.64E+06	2.78E+06	7.E+04	5.0	1.E-01	3.62E+06	3.83E+06	1.E+05	5.8	**1.E-04**
MTPA-SWO	1.79E+06	1.90E+06	6.E+04	3.9	2.E-01	2.62E+06	2.75E+06	9.E+04	3.9	6.E-01	3.57E+06	3.75E+06	1.E+05	4.0	9.E-02
MTPA-GTO	1.83E+06	1.97E+06	8.E+04	6.9	2.E-05	2.63E+06	2.80E+06	8.E+04	5.6	1.E-02	3.65E+06	3.85E+06	1.E+05	6.6	**3.E-06**

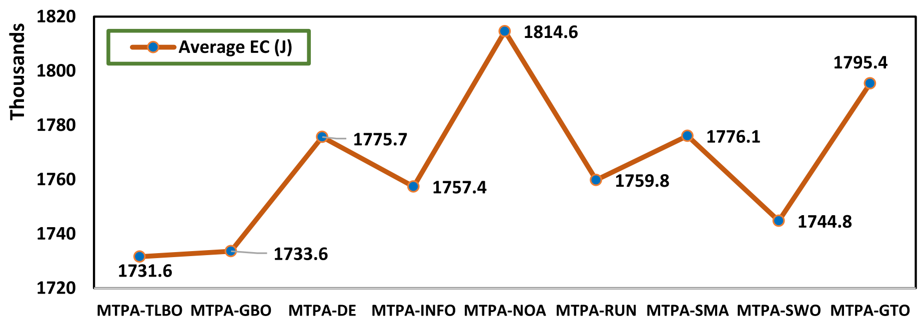

Fig. 18 Average EC value under the E_{iot} objective for nine multi-objective metaheuristic algorithms

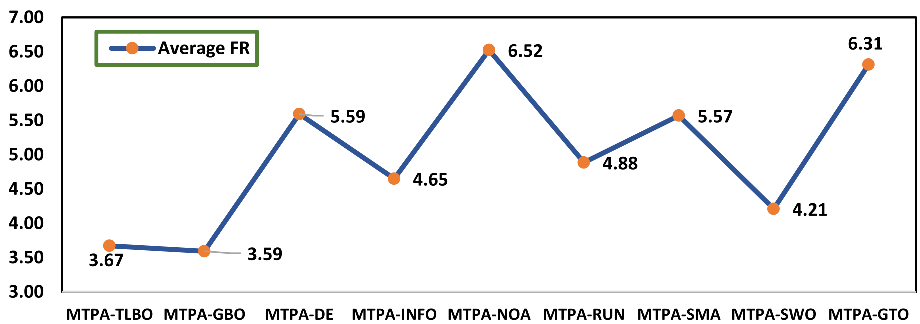

Fig. 19 Average FR value under the E_{iot} objective for nine multi-objective metaheuristic algorithms

Table 7 Comparison among five multiobjective metaheuristic algorithms with CSM over SI

		I-80				I-100					I-120				
	BEC	AEC	SD	FR	PV	BEC	AEC	SD	PV	FR	BEC	AEC	SD	PV	FR
MTPA-VTLBO	2.04E+06	2.08E+06	2.E+04	3.4		2.63E+06	2.67E+06	2.E+04	2.7		2.74E+06	2.80E+06	3.E+04	2.7	
MTPA-VGBO	2.01E+06	**2.06E+06**	**2.E+04**	**1.9**	**3.E-03**	**2.61E+06**	2.67E+06	**2.E+04**	2.8	7.E-01	**2.73E+06**	**2.80E+06**	**2.E+04**	**2.7**	5.E-01
MTPA-VDE	2.05E+06	2.09E+06	2.E+04	3.8	2.E-01	2.65E+06	2.69E+06	2.E+04	4.1	1.E-04	2.75E+06	2.82E+06	3.E+04	3.8	8.E-02
MTPA-VINFO	2.02E+06	2.07E+06	2.E+04	3.2	6.E-01	2.62E+06	**2.66E+06**	2.E+04	**2.6**	1.E+00	2.74E+06	2.79E+06	3.E+04	2.8	5.E-01
MTPA-VSWO	**2.00E+06**	2.07E+06	2.E+04	2.7	1.E-01	2.62E+06	**2.66E+06**	3.E+04	2.8	8.E-01	2.75E+06	2.80E+06	3.E+04	3.0	9.E-01
		I-140				I-160					I-180				
MTPA-VTLBO	3.59E+06	3.66E+06	3.E+04	2.8		4.14E+06	4.21E+06	3.E+04	2.6		4.73E+06	4.79E+06	3.E+04	2.8	
MTPA-VGBO	**3.58E+06**	**3.65E+06**	**3.E+04**	**1.9**	**2.E-02**	4.14E+06	**4.20E+06**	**2.E+04**	**2.5**	5.E-01	**4.68E+06**	**4.79E+06**	4.E+04	3.0	8.E-01
MTPA-VDE	3.64E+06	3.69E+06	4.E+04	3.9	2.E-02	4.16E+06	4.24E+06	4.E+04	4.0	**8.E-03**	4.76E+06	4.82E+06	3.E+04	4.2	**3.E-03**
MTPA-VINFO	3.60E+06	3.68E+06	4.E+04	3.4	1.E-01	**4.13E+06**	4.21E+06	3.E+04	3.0	5.E-01	4.73E+06	4.78E+06	3.E+04	2.4	2.E-01
MTPA-VSWO	3.62E+06	3.67E+06	3.E+04	3.0	8.E-01	4.17E+06	4.21E+06	2.E+04	2.9	8.E-01	4.74E+06	4.79E+06	3.E+04	**2.6**	8.E-01
		I-200				I-300					I-400				
MTPA-VTLBO	5.27E+06	5.33E+06	3.E+04	3.2		**7.51E+06**	7.60E+06	**4.E+04**	2.9		1.05E+07	**1.05E+07**	**4.E+04**	**2.4**	
MTPA-VGBO	5.27E+06	**5.32E+06**	**2.E+04**	**2.4**	8.E-02	**7.51E+06**	**7.59E+06**	**4.E+04**	2.6	5.E-01	**1.04E+07**	1.05E+07	6.E+04	2.8	7.E-01
MTPA-VDE	5.27E+06	5.36E+06	4.E+04	3.7	**2.E-02**	7.54E+06	7.64E+06	**4.E+04**	3.8	**8.E-03**	1.05E+07	1.06E+07	7.E+04	4.2	**4.E-04**
MTPA-VINFO	**5.26E+06**	5.33E+06	4.E+04	2.9	4.E-01	**7.51E+06**	**7.59E+06**	**4.E+04**	**2.5**	6.E-01	1.05E+07	1.05E+07	4.E+04	2.8	3.E-01
MTPA-VSWO	**5.26E+06**	5.32E+06	3.E+04	2.8	2.E-01	7.54E+06	7.60E+06	**4.E+04**	3.2	7.E-01	**1.04E+07**	1.05E+07	6.E+04	2.8	7.E-01

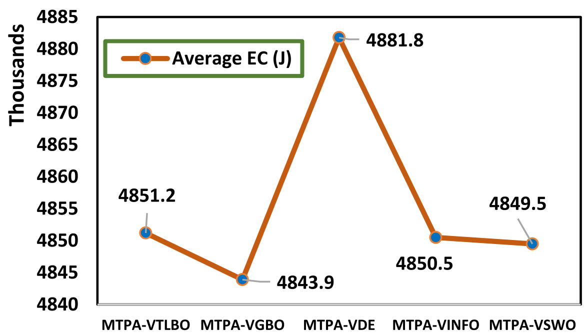

Fig. 20 Average EC value under *SI* for algorithms with CSM

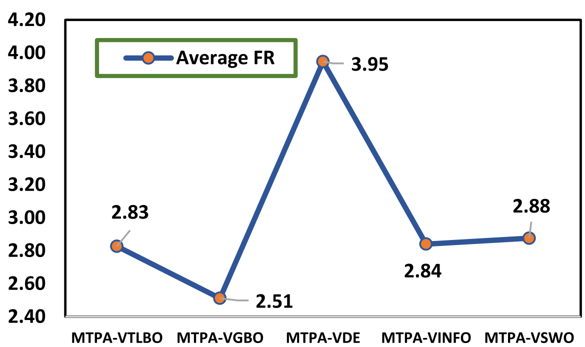

Fig. 21 Average FR value under *SI* for algorithms with CSM

of average AEC, MTPA-VSWO is the second highest-performing algorithm, and MTPA-VDE is the worst-performing algorithm.

Comparison among some algorithms under VIPS and CSM

Finally, in this section, we will compare the outcomes obtained by the top-ranked multiobjective metaheuristic algorithms with both VIPS and CSM to highlight their effectiveness in minimizing the total EC of UAVs. Figure 27 presents the average AEC value for SI, the E_{iot} objective, and the E_{UAV} objective obtained by each algorithm in all instances. This figure shows that the performance of the algorithms with CSM could be better than that with the VIPS mechanism for SI and F2 objectives; however, MTPA-VGBO could be better than all the others. On the contrary, for the F1 objective, the multiobjective algorithms with the VIPS mechanism could be slightly better than the algorithms with CSM. Among them, MTPA-INFO could achieve the lowest average value for this objective. From that, it is clear that the cyclic selection mechanism has a significant effect on the performance of the optimization algorithms, where it could accelerate their convergence speed for achieving outstanding outcomes for the F2 objective in fewer function evaluations and somewhat competitive outcomes for the F1 objective. In addition, it is worth mentioning that the Pareto optimality theory could simultaneously optimize both F1 and F2 objectives because it is not reliant on relating the objectives together using a weighted variable, and hence the solution that could improve at least one objective without deteriorating any of the others is considered a non-dominated solution. On the contrary, the weighted objective function, which relates the two objectives together using a weighted variable, does not take into consideration the deterioration of some objectives, but it relies on the objective values of the estimated solutions. For example, assume that the TPA-DE algorithm could find a new solution with a better value for the F2 objective but a worse value for the F1 objective; however, the F2 objective has the highest effect on the objective function, thereby making the overall objective value for this solution smaller than the best-so-far objective value. Hence, this solution is considered the new best-so-far solution, even though it negatively affects the F1 objective.

Conclusion and future work

Due to their limited storage and computing capacity, IoT devices are incapable of executing resource-intensive tasks. Therefore, the MEC technology has been utilized recently to provide computing and storage capabilities to these devices so that they can execute these tasks with low latency and low EC. However, because the MEC network's edge servers are located in a fixed location, they cannot be adjusted to meet the needs of end users. In order to overcome this limitation, UAVs have recently been loaded with edge servers to assist the MEC systems in presenting better computing services to end users. However, the trajectories of the UAVs must be meticulously planned in order to reduce the amount of energy consumed by IoT devices during data transmission and by UAVs during hovering time and mobility between halting points (HPs). The trajectory planning problem is a difficult optimization problem due to the multiple factors that should be taken into consideration when solving it; these factors include the location and number of HPs, the relationship between HPs and UAVs, and the sequence of HPs for each UAV. This problem is a multiobjective optimization problem because it requires optimizing the EC

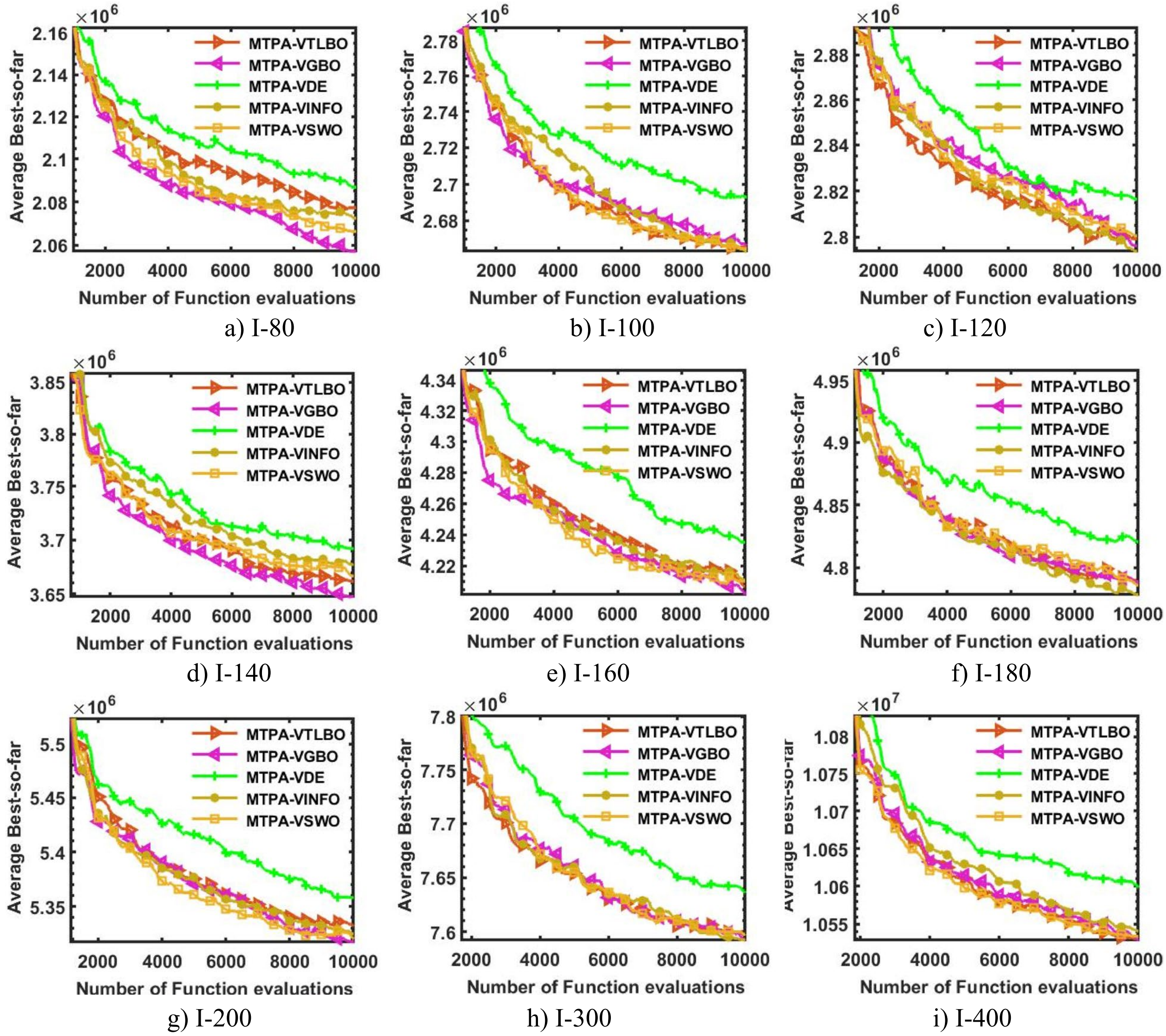

Fig. 22 Convergence curve under SI obtained by five top-ranked multiobjective metaheuristic algorithms with CSM on all instances

of UAVs and IoT devices simultaneously. To optimize both objectives simultaneously, this study presents several multiobjective trajectory planning algorithms (MTPA) for solving this problem. Those algorithms are based on several metaheuristic algorithms with variable population size and the Pareto optimality theory. In addition, a novel mechanism, namely the cyclic selection mechanism, to manage the population size for optimizing the number of HPs for each UAV more accurately is proposed in this study. Moreover, the HPs estimated by each MTPA are associated with the multiple UAVs using the k-means clustering algorithm, and then the low-complexity greedy mechanism is used to generate the order of HPs assigned to each UAV for generating its trajectory. Several experiments are conducted to determine the efficacy of Pareto optimality and CSM with the investigated metaheuristic algorithms for solving multiple instances involving 80 to 400 IoT devices. Experiment results indicate that treating this problem as a multiobjective can result in a higher total EC than converting it to a single objective. In addition, the CSM has the potential to substantially

Table 8 Comparison of five multiobjective metaheuristic algorithms with CSM over E_{UAV}

		I-80				**I-100**					**I-120**				
	BEC	**AEC**	**SD**	**FR**	**PV**	**BEC**	**AEC**	**SD**	**FR**	**PV**	**BEC**	**AEC**	**SD**	**FR**	**PV**
MTPA-VTLBO	**135.85**	135.98	2.E-01	3.5		176.21	176.38	2.E-01	3.0		180.78	181.06	3.E-01	3.1	
MTPA-VGBO	**135.85**	135.94	2.E-01	**2.8**	2.E-01	**176.21**	**176.33**	**2.E-01**	**2.4**	**1.E-02**	180.79	181.24	5.E-01	3.6	2.E-01
MTPA-VDE	**135.85**	135.98	2.E-01	**2.8**	7.E-01	176.21	176.45	3.E-01	3.2	9.E-01	**180.78**	**180.88**	**1.E-01**	**1.9**	**8.E-03**
MTPA-VINFO	**135.85**	**135.92**	**8.E-02**	**2.8**	4.E-01	176.21	176.48	3.E-01	3.0	6.E-01	180.78	181.14	4.E-01	3.4	7.E-01
MTPA-VSWO	**135.85**	135.97	2.E-01	3.1	6.E-01	176.22	176.43	2.E-01	3.2	6.E-01	180.78	181.13	4.E-01	3.0	9.E-01
		I-140				I-160					I-180				
MTPA-VTLBO	239.49	**239.68**	1.E-01	3.0		**275.31**	275.56	3.E-01	2.8		315.95	316.15	2.E-01	3.0	
MTPA-VGBO	**239.48**	239.73	3.E-01	3.0	8.E-01	275.32	275.53	**2.E-01**	3.4	4.E-01	**315.95**	**316.12**	2.E-01	3.0	1.E+00
MTPA-VDE	239.50	239.80	3.E-01	3.2	4.E-01	275.32	**275.52**	3.E-01	**2.3**	8.E-01	315.95	316.19	2.E-01	3.4	3.E-01
MTPA-VINFO	239.49	**239.68**	**2.E-01**	**2.8**	6.E-01	275.32	275.55	2.E-01	3.3	4.E-01	315.96	316.18	3.E-01	3.2	7.E-01
MTPA-VSWO	**239.48**	239.73	3.E-01	3.0	9.E-01	275.32	275.56	3.E-01	3.2	6.E-01	315.95	**316.12**	3.E-01	**2.4**	4.E-01
		I-200				I-300					I-400				
MTPA-VTLBO	**351.30**	351.55	3.E-01	**2.6**		498.27	**498.60**	3.E-01	**2.9**		696.34	696.66	3.E-01	3.0	
MTPA-VGBO	351.31	351.58	3.E-01	2.9	6.E-01	498.26	**498.60**	4.E-01	2.8	6.E-01	696.37	696.81	5.E-01	3.2	3.E-01
MTPA-VDE	351.32	351.64	3.E-01	3.1	2.E-01	498.27	498.63	3.E-01	3.2	4.E-01	**696.33**	696.85	5.E-01	3.4	2.E-01
MTPA-VINFO	351.34	351.76	4.E-01	3.5	**6.E-03**	**498.25**	498.70	4.E-01	3.2	4.E-01	696.34	696.81	6.E-01	2.8	8.E-01
MTPA-VSWO	351.31	**351.54**	**2.E-01**	2.9	3.E-01	**498.25**	498.66	5.E-01	2.8	6.E-01	**696.33**	**696.61**	3.E-01	**2.6**	6.E-01

Table 9 Comparison of five multiobjective metaheuristic algorithms with CSM over E_{iot}

		I-80				I-100					I-120				
	BEC	AEC	SD	FR	PV	BEC	AEC	SD	PV	FR	BEC	AEC	SD	FR	PV
MTPA-VTLBO	6.85E+05	7.18E+05	**2.E+04**	3.3		8.69E+05	9.02E+05	**2.E+04**	**2.8**		9.30E+05	9.89E+05	3.E+04	2.8	
MTPA-VGBO	6.51E+05	**6.98E+05**	**2.E+04**	**1.9**	**5.E-03**	8.51E+05	9.03E+05	**2.E+04**	**2.8**	7.E-01	**9.21E+05**	**9.84E+05**	**2.E+04**	**2.5**	4.E-01
MTPA-VDE	6.87E+05	7.27E+05	**2.E+04**	3.8	2.E-01	8.86E+05	9.29E+05	**2.E+04**	4.1	**2.E-04**	9.44E+05	1.01E+06	3.E+04	3.8	4.E-02
MTPA-VINFO	6.63E+05	7.13E+05	**2.E+04**	3.2	6.E-01	8.57E+05	8.99E+05	**2.E+04**	2.5	7.E-01	9.33E+05	9.82E+05	2.E+04	2.8	4.E-01
MTPA-VSWO	**6.46E+05**	7.07E+05	**2.E+04**	2.8	1.E-01	**8.48E+05**	**9.00E+05**	3.E+04	2.8	9.E-01	9.27E+05	9.89E+05	3.E+04	3.1	9.E-01
		I-140				I-160					I-180				
MTPA-VTLBO	1.19E+06	1.26E+06	3.E+04	2.8		**1.38E+06**	**1.45E+06**	3.E+04	2.6		1.57E+06	**1.63E+06**	**3.E+04**	2.8	
MTPA-VGBO	**1.18E+06**	**1.25E+06**	**2.E+04**	**1.9**	**3.E-02**	1.39E+06	**1.45E+06**	**2.E+04**	**2.5**	5.E-01	**1.52E+06**	**1.63E+06**	4.E+04	3.0	7.E-01
MTPA-VDE	1.24E+06	1.29E+06	4.E+04	3.8	**2.E-02**	1.40E+06	1.48E+06	4.E+04	3.9	**8.E-03**	1.60E+06	1.66E+06	3.E+04	4.1	**3.E-03**
MTPA-VINFO	1.20E+06	1.28E+06	4.E+04	3.4	1.E-01	1.38E+06	1.46E+06	3.E+04	3.0	5.E-01	1.57E+06	1.62E+06	3.E+04	**2.4**	2.E-01
MTPA-VSWO	1.22E+06	1.27E+06	3.E+04	3.1	9.E-01	1.41E+06	1.45E+06	2.E+04	2.9	7.E-01	1.58E+06	1.63E+06	3.E+04	2.7	8.E-01
		I-200				I-300					I-400				
MTPA-VTLBO	1.76E+06	1.82E+06	3.E+04	3.2		**2.52E+06**	2.61E+06	4.E+04	2.8		3.50E+06	3.56E+06	**4.E+04**	**2.6**	
MTPA-VGBO	1.76E+06	**1.80E+06**	**2.E+04**	**2.4**	6.E-02	**2.52E+06**	**2.61E+06**	**4.E+04**	**2.5**	5.E-01	**3.44E+06**	**3.56E+06**	6.E+04	2.7	8.E-01
MTPA-VDE	1.75E+06	1.84E+06	4.E+04	3.7	**3.E-02**	2.56E+06	2.65E+06	4.E+04	3.8	**1.E-02**	3.52E+06	3.63E+06	7.E+04	4.2	**5.E-04**
MTPA-VINFO	1.75E+06	1.81E+06	4.E+04	2.8	3.E-01	2.53E+06	2.61E+06	4.E+04	2.6	5.E-01	3.51E+06	3.57E+06	**4.E+04**	2.8	5.E-01
MTPA-VSWO	**1.74E+06**	**1.80E+06**	3.E+04	2.9	2.E-01	2.55E+06	2.61E+06	4.E+04	3.2	7.E-01	3.46E+06	3.57E+06	6.E+04	2.8	8.E-01

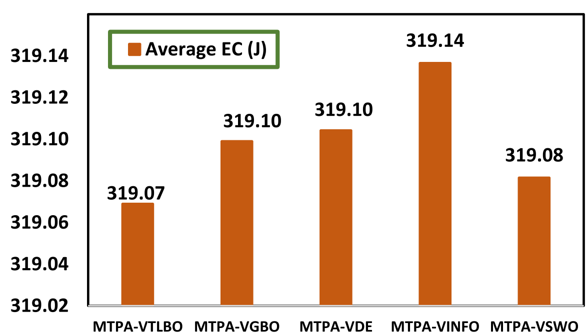

Fig. 23 Average EC value under E_{UAV} objective for algorithms with CSM

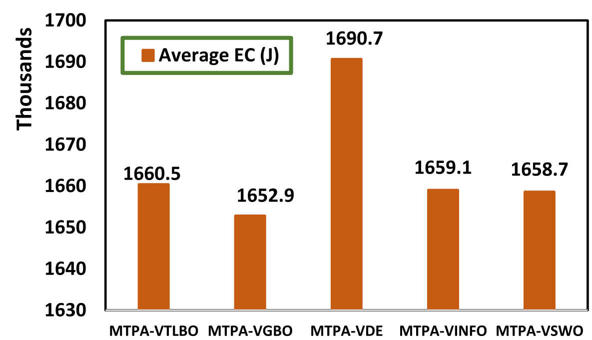

Fig. 25 Average EC value under E_{iot} objective for algorithms with CSM

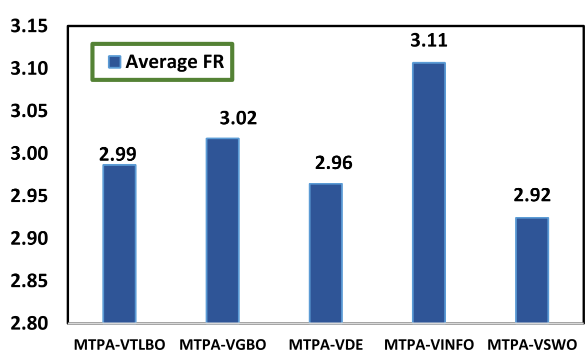

Fig. 24 Average FR value under E_{UAV} objective for algorithms with CSM

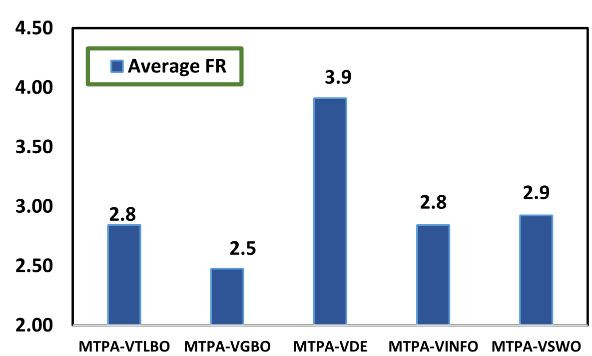

Fig. 26 Average FR value under E_{iot} objective for algorithms with CSM

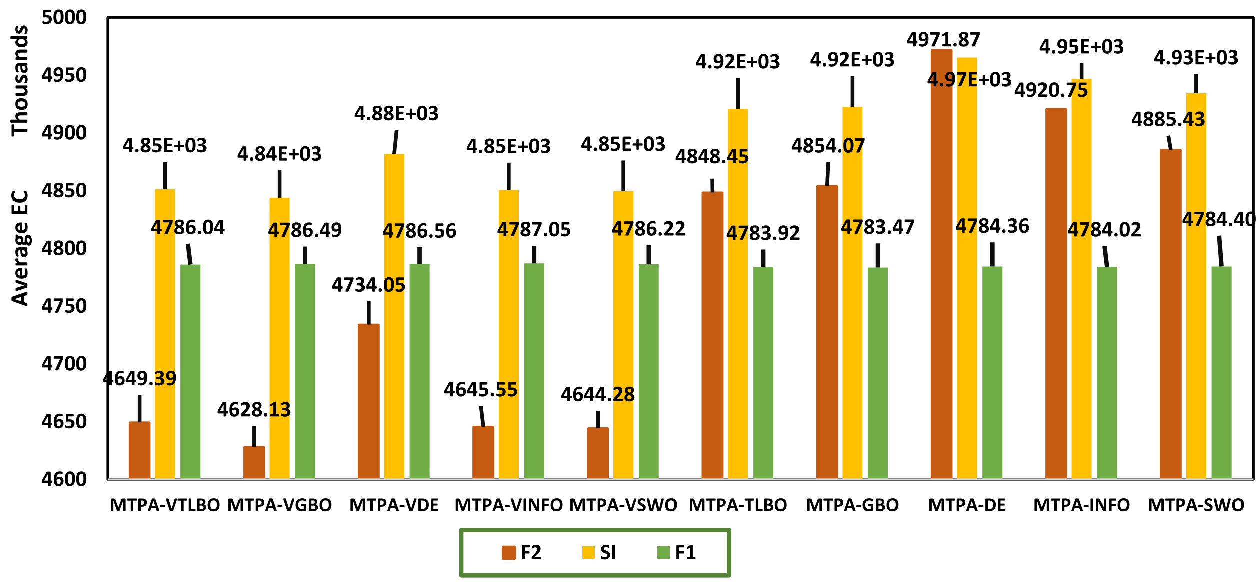

Fig. 27 Average EC value under SI, F1, and F2 for top-ranked algorithms with CSM and VIPS mechanism

enhance the performance of optimization algorithms. Among all examined algorithms, MTPA-VTLBO and MTPA-VGBO were capable of producing the best results.

Our future work will involve employing the proposed CSM with some recently published metaheuristic algorithms, such as the mantis search algorithm, for tackling resource allocation and mining decisions in MEC-supported blockchain networks. In addition, the greedy mechanism used for constructing the order of HPs for each UAV will be replaced with a metaheuristic algorithm in the hope of achieving better trajectories.

Nomenclature

m The number of UAVs

n The number of IoT devices

t_{max} The maximum function evaluation

E_{UAV} Energy consumption of UAV

E_{iot} Energy consumption of all IoT devices

t The current function evaluation

Abbreviations

HPs	Halting points
EC	Energy consumption
MEC	Mobile edge computing
UMEC	Multi-UAV-assisted MEC systems
WRS	Wilcoxon rank-sum
FR	Friedman means rank
UAV	Unmanned aerial vehicle
IoT	Internet of Things

Authors' contributions

MA; RM and IMH Investigation, Methodology, Resources, Supervision, Visualization, Writing original draft, Writing-—review and editing, Software, Conceptualization, Methodology, Writing-review and editing. KMS and IAH Investigation, Methodology, Validation, Writing—original draft, Writing-review and editing. A F Resources , Investigation, Validation, Writing-review and editing.

Funding

Open access funding provided by Norwegian University of Science and Technology This research is supported by the Researchers Supporting Project number (RSP2024R389), King Saud University, Riyadh, Saudi Arabia.

Availability of data and materials

No datasets were generated or analysed during the current study.

Declarations

Ethics approval and consent to participate

None of the authors experimented with human subjects or animals during this research.

Competing interests

The authors declare no competing interests.

Author details

[1]Zagazig University, Zagazig, Sharqiyah 44519, Egypt. [2]Computer Science Department, College of Computing and Informatics, University of Sharjah, Sharjah, United Arab Emirates. [3]School of IT and Systems, Faculty of Science and Technology, University of Canberra, Canberra, Australia. [4]Statistics & Operations Research Department, College of Sciences, King Saud University, Riyadh 11451, Saudi Arabia. [5]Department of ICT and Natural Sciences, Norwegian University of Science and Technology (NTNU), Ålesund, Norway.

Received: 24 November 2023 Accepted: 7 January 2024
Published online: 07 February 2024

References

1. E. Ahmed and M. H. Rehmani, "Mobile edge computing: opportunities, solutions, and challenges," vol. 70, ed: Elsevier, 2017, pp. 59–63.
2. Huang P-Q, Wang Y, Wang K-Z (2020) Energy-efficient trajectory planning for a multi-UAV-assisted mobile edge computing system. Front Inform Technol Elect Eng 21(12):1713–1725

3. Huang P-Q, Wang Y, Wang K, Yang K (2019) Differential evolution with a variable population size for deployment optimization in a UAV-assisted IoT data collection system. IEEE Transact Emerg Topics Computational Intelligence 4(3):324–335
4. Zhang L, Zhao Z, Wu Q, Zhao H, Xu H, Wu X (2018) Energy-aware dynamic resource allocation in UAV assisted mobile edge computing over social internet of vehicles. IEEE Access 6:56700–56715
5. Asim M, Mashwani WK, Shah H, Belhaouari SB (2022) An evolutionary trajectory planning algorithm for multi-UAV-assisted MEC system. Soft Comput 26(16):7479–7492
6. Li W, Guo Y, Li N, Yuan H, Liu C (2023) Resource allocation and trajectory optimization in OTFS-Based UAV-assisted mobile edge computing. Electronics 12(10):2212
7. Wang L, Wang K, Pan C, Xu W, Aslam N, Hanzo L (2020) Multi-agent deep reinforcement learning-based trajectory planning for multi-UAV assisted mobile edge computing. IEEE Transact Cognitive Commun Net 7(1):73–84
8. Savkin AV, Huang C, Ni W (2022) Joint multi-UAV path planning and LoS communication for mobile-edge computing in IoT networks with RISs. IEEE Internet Things J 10(3):2720–2727
9. Asim M, Mashwani WK, Belhaouari SB, Hassan S (2021) A novel genetic trajectory planning algorithm with variable population size for multi-UAV-assisted mobile edge computing system. IEEE Access 9:125569–125579
10. Zhang J et al (2019) Computation-efficient offloading and trajectory scheduling for multi-UAV assisted mobile edge computing. IEEE Trans Veh Technol 69(2):2114–2125
11. Sun C, Ni W, Wang X (2021) Joint computation offloading and trajectory planning for UAV-assisted edge computing. IEEE Trans Wireless Commun 20(8):5343–5358
12. H. Wu, M. Wu, W. Peng, S. Chen, and Z. Feng, "ITS: Improved Tabu Search Algorithm for Path Planning in UAV-Assisted Edge Computing Systems," pp. 340–349: IEEE.
13. M. J. Sobouti, A. H. Mohajerzadeh, S. A. H. Seno, and H. Yanikomeroglu, "Managing Sets of Flying Base Stations Using Energy Efficient 3D Trajectory Planning in Cellular Networks," IEEE Sensors Journal, 2023.
14. Lin J, Pan L (2022) Multiobjective trajectory optimization with a cutting and padding encoding strategy for single-UAV-assisted mobile edge computing system. Swarm Evol Comput 75:101163
15. Chen E, Chen J, Mohamed AW, Wang B, Wang Z, Chen Y (2020) Swarm intelligence application to UAV aided IoT data acquisition deployment optimization. IEEE Access 8:175660–175668
16. Zhang Y, Huang C, Huang H (2023) Backtracking search algorithm with dynamic population for energy consumption problem of a UAV-assisted IoT data collection system. Eng Appl Artif Intell 123:106331
17. Y. Zhang, Y. Huang, C. Huang, H. Huang, and A.-T. Nguyen, "Joint Optimization of Deployment and Flight Planning of Multi-UAVs for Long-distance Data Collection from Large-scale IoT Devices," IEEE Internet of Things Journal, 2023.
18. Abu-Baker A, Shakhatreh H, Sawalmeh A, Alenezi AH (2023) Efficient data collection in UAV-assisted cluster-based wireless sensor networks for 3D environment: optimization study. J Sensors 12:2023
19. Y. Miao, K. Hwang, D. Wu, Y. Hao, and M. Chen, "Drone swarm path planning for mobile edge computing in industrial internet of things," IEEE Transactions on Industrial Informatics, 2022.
20. Lu W et al (2022) Secure transmission for multi-UAV-assisted mobile edge computing based on reinforcement learning. IEEE Transact Net Sci Eng 10(3):1270–1282
21. Liu Q, Shi L, Sun L, Li J, Ding M, Shu F (2020) Path planning for UAV-mounted mobile edge computing with deep reinforcement learning. IEEE Trans Veh Technol 69(5):5723–5728
22. F. Song et al., "Evolutionary multi-objective reinforcement learning based trajectory control and task offloading in UAV-assisted mobile edge computing," IEEE Transactions on Mobile Computing, 2022.
23. Abdollahzadeh B, SoleimanianGharehchopogh F, Mirjalili S (2021) Artificial gorilla troops optimizer: a new nature-inspired metaheuristic algorithm for global optimization problems. Intern J Intelligent Syst 36(10):5887–5958
24. Ahmadianfar I, Bozorg-Haddad O, Chu X (2020) Gradient-based optimizer: a new metaheuristic optimization algorithm. Inf Sci 540:131–159
25. Rao RV, Savsani VJ, Vakharia DP (2011) Teaching–learning-based optimization: a novel method for constrained mechanical design optimization problems. Comput Aided Des 43(3):303–315
26. Abdel-Basset M, Mohamed R, Jameel M, Abouhawwash M (2023) Nutcracker optimizer: a novel nature-inspired metaheuristic algorithm for global optimization and engineering design problems. Knowl-Based Syst 262:110248
27. Li S, Chen H, Wang M, Heidari AA, Mirjalili S (2020) Slime mould algorithm: a new method for stochastic optimization. Futur Gener Comput Syst 111:300–323
28. Abdel-Basset M, Mohamed R, Jameel M, Abouhawwash M (2023) Spider wasp optimizer: a novel meta-heuristic optimization algorithm. Artificial Intelligence Rev 2023(54):1–64
29. Ahmadianfar I, Heidari AA, Gandomi AH, Chu X, Chen H (2021) RUN beyond the metaphor: an efficient optimization algorithm based on Runge Kutta method. Expert Syst Appl 181:115079
30. Ahmadianfar I, Heidari AA, Noshadian S, Chen H, Gandomi AH (2022) INFO: an efficient optimization algorithm based on weighted mean of vectors. Expert Syst Appl 195:116516
31. Abdel-Basset M, Mohamed R, K-B SJ, Mirjalili S (2021) "A novel whale optimization algorithm integrated with for multi-objective optimization problems. Knowledge-Based Syst 212:106619

Publisher's Note