drones

MDPI

Article

A Self-Adaptive Trajectory Optimization Algorithm Using Fuzzy Logic for Mobile Edge Computing System Assisted by Unmanned Aerial Vehicle

Brindha Subburaj [1,*], Uma Maheswari Jayachandran [1], Vinothini Arumugham [1] and Miruna Joe Amali Suthanthira Amalraj [2]

1 School of Computer Science and Engineering, Vellore Institute of Technology, Chennai 600127, India
2 Department of Computer Science and Engineering, K.L.N College of Engineering, Sivagangai 630612, India
* Correspondence: brindha.s@vit.ac.in

Abstract: The advancement of the Internet of Things (IoT) and the availability of wide cloud services have led to the horizon of edge computing paradigm which demands for processing the data at the edge of the network. The development of 5G technology has led to the increased usage of IoT-based devices and the generation of a large volume of data followed by increased data traffic, which is difficult to process by the mobile edge computing (MEC) platform. The latest inventions related to unmanned aerial vehicles (UAVs) helps to assist and replace the edge servers used for MEC. In the present work, the objective is to develop self-adaptive trajectory optimization algorithm (STO) which is a multi-objective optimization algorithm used to solve the vital objectives associated with the above scenario of a UAV-assisted MEC system. The objectives identified are minimizing the energy consumed by the MEC and minimizing the process emergency indicator, where the process emergency indicator implies the urgency level of a particular process. Finding the optimal values for these conflicting objectives will help to further efficiently apply UAV for MEC systems. A self-adaptive multi-objective differential evolution-based trajectory optimization algorithm (STO) is proposed, where a pool of trial vector generation strategies is extended. The strategies and the crossover rate associated with a differential evolution (DE) algorithm are self-adapted using fuzzy systems to improve the population diversity. The experimentation is planned to be conducted on hundreds of IoT device instances considered to be fixed on the ground level and to evaluate the performance of the proposed algorithm for a single unmanned aerial vehicle-assisted mobile edge computing system.

Keywords: multi-objective optimization; mobile edge computing; unmanned aerial vehicle; differential evolution algorithm

check for updates

Citation: Subburaj, B.; Jayachandran, U.M.; Arumugham, V.; Suthanthira Amalraj, M.J.A. A Self-Adaptive Trajectory Optimization Algorithm Using Fuzzy Logic for Mobile Edge Computing System Assisted by Unmanned Aerial Vehicle. *Drones* **2023**, *7*, 266. https://doi.org/10.3390/drones7040266

Academic Editor: Carlos Tavares Calafate

Received: 11 February 2023
Revised: 14 March 2023
Accepted: 27 March 2023
Published: 13 April 2023

Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

The recent trend in computing is edge computing, where the computations are performed closer to the source than computing in the cloud [1]. Edge computing thus leverages the cloud computing services to the various devices at the proximity of the network. The advent of 5G communication results in much more utilization of IoT devices leading to the generation of huge volumes of data to be processed leading to data traffic. The edge servers at the mobile edge computing systems are grounded at specific locations, making it infeasible to handle such increasing loads [2]. Moreover, in remote areas where the coverage of wireless network is poor, IoT devices with energy limitation in general fail to transfer data to computation environment [3].

Unmanned aerial vehicles (UAVs) are widely applied in various domains and more advancements are made in this field over the last decade. Recent advances in components, modern sensors, and 5G communication has further contributed to the seamless application of UAV. Two major classifications of UAVs are rotary-wing and fixed-wing UAVs [4]. UAVs are applied in agriculture, smart transportation, rescue system, path planning, etc. Thus, they are used to replace the conventional fixed edge server system which serves the mobile edge computing system. The UAV-assisted mobile edge computing system can overcome the difficulties related to the mobile edge computing system [5]. UAVs thus can be an alternative data collection environment for the Internet of Things devices. Deploying UAVs for MEC system requires several factors to be improved and optimized.

Optimization techniques attempt to find solutions that are feasible for the optimization problem considered. Several application domains do require the associated parameters to be optimized and they are defined as objective function. There can be one or more objectives in an application that needs to be optimized. Several classes of search and optimization techniques are available in the literature and numerous research reports in the field are carried and various algorithms are proposed. Evolutionary algorithms (EAs) are one such widely used search and optimization technique. Genetic algorithm (GA) [6], differential evolution (DE), [7] and particle swarm optimization (PSO) [8] are popular EA-based optimization algorithms.

Recently, the UAV-based MEC system is subject to several research studies to improve the applicability of UAV for MEC systems. The previous vital research contributions in this field are presented here categorizing based on the number of objectives, whether single or multi-objective problems considered. Next, we present related work based on whether the stop position (SP) count of UAV considered as fixed or not. SP is the locations where UAV is hovered at a point for processing the data generated by IoT devices at the location. Finally, we present earlier research contributions related to process emergency indicator.

The common research conducted is related to minimize the energy consumed by the mobile edge computing systems, maximizing the coverage area and few are highlighted. The differential evolution algorithm is extended to minimize the energy factor using a varying population size [9]. In [10], the authors have extended a two-layer optimization method to reduce the energy consumption of the system. In [11], optimization techniques are applied to improve the coverage factor by reducing the transmission power. The techniques to improve the energy efficiency in UAV-based MEC systems are discussed in [12,13]. In these research works, the authors have attempted to solve a single-objective optimization problem associated with MEC systems such as the energy consumption or coverage. The real world optimization problems in MEC systems assisted by UAVs have several conflicting objectives that need to be optimized. A deep Q-network-based strategy is used to solve the multi-objective optimization problem with the objectives: to decrease the latency and energy consumption [14]. A heuristic algorithm is extended by the authors to optimize the objectives, minimizing energy consumption and reducing the task execution delay [15].

Most of the existing research studies attempted to prefix the number of stop positions (SPs) associated along the optimization problem in MEC systems assisted by UAVs, few are listed here. UAV-based data collection systems through deployment optimization [16]. In [17], the authors deployed UAV-based systems for a downlink-based communication application to improve the coverage factor. The above way of attempting UAV deployment for MEC by presetting the stop position with respect to the application may not beneficial. Especially if the number of SPs is not optimal, then the efficiency of UAV-based systems will be reduced. In the present work, the SP is considered as unknown and we try to optimize the stop positions and locations in the proposed algorithm. The applications where UAV-based MEC systems used also includes mission critical tasks, where the time taken by the system to service or process the task is vital. Thus, the important and emergency tasks needed to be prioritized and served as soon as possible [18].

The above research contributions related to the mobile edge computing system assisted by UAVs has motivated to further leverage the efficient applicability of the UAV-based system. The major research contributions are listed below:

- Multi-objective optimization problem for the MEC system assisted with single UAV is represented. The multiple conflicting objectives to be optimized are minimizing the energy consumed by the MEC and minimizing the process emergency indicator.
- To develop a self-adaptive multi-objective differential evolution-based trajectory optimization algorithm (STO) to optimize the multi-objective problem identified in the MEC system assisted by UAVs. STO algorithm is developed based on DE algorithm. The trial vector generation strategies and control parameters associated in DE algorithm are self-adapted using the success index in self-adaptive differential evolution (SaDE) algorithm [19]. Fuzzy Inference system is used to further improve the adaptation characteristics of control parameters in the FAMDE-DC algorithm [20]. The proposed STO algorithm is developed using the above adaptation techniques, along with appropriate encoding strategies [21] suitable to handle the implementation issues associated in the trajectory optimization problem.
- The experimentation is performed by simulating environment with up to hundreds of IoT devices. Results are compared with related research findings and the inferences are discussed.

The paper is structured as follows: The multi-objective optimization problem formulation is detailed in Section 2. The proposed STO algorithm is given in Section 3. The experimental setup and the performance metrics used in the present research study are discussed in Section 4. The results and its analysis, comparison of the results with other related work are given in Section 5. The conclusion of the present work is discussed in Section 6.

2. Multi-Objective Optimization Problem Formulation

Problem formulation is a primary requirement to successfully apply optimization technique to solve any domain specific problem to derive the feasible solution. In the present work, the problem is a multi-objective optimization problem and the objectives are to minimize the energy consumed by the MEC and to minimize the process emergency indicator. The objective functions [11] are presented in this section.

The MEC system considered is assisted with a single UAV and a number of IoT devices given as $T = \{1, 2, .., t\}$. The MEC system is used to complete the process or work associated with the IoT devices. A process i in an IoT device is characterized by three factors $(A_i,\ B_i, C_i)$ where A_i is the number of input information of process i, B_i indicates the computational resources needed to complete one bit involved in process i. C_i is the process emergency indicator and it is prioritized as:

$$C_i \ \in \{1, 2, 3, 4, 5\} \tag{1}$$

It is prioritized such that lesser the C_i value, the most emergency process and is to be computed first.

As mentioned earlier, in the present work the number of stop position is considered to be unknown and we try to optimize it. Thus, the stop position (SP) is given as $S = [1, 2, .., s]$. Next vital parameter is the location of the IoT device and the SP. The coordinates representing IoT device's location is set at $(l_i, m_i, 0)$. Since the device is considered to be at ground, the value of third component representing height is set at 0. The location coordinates of a random stop position j is set as $\left(L_j, M_j, H\right)$, the third component H is the height of the UAV device. The following equation gives the distance between an IoT device and the stop position.

$$d_{ij} = \sqrt{\left(L_j - l_i\right)^2 - \left(M_j - m_j\right)^2 + (H - 0)^2} \tag{2}$$

The variable E_{ij} is used to represent an edge computing link established between an IoT device and UAV's stop position. The device usually establishes link with the nearest UAV's stop position. E_{ij} is subject to the two constraints as given in Equation (3). $E_{ij} = 1$

illustrates a valid link establishment and $E_{ij} = 0$ gives that no such link is established. Further, each device transmits data only at one stop position as given in constraint 2.

$$\text{Constraint 1}: \ E_{ij} = \begin{cases} 1, & if\ j = argmin_{j \in S}\ d_{ij}, \\ 0, & otherwise, \end{cases} \tag{3}$$

and

$$\text{Constraint 2}: \ \sum_{j=1}^{s} E_{ij} = 1, i \in T. \tag{4}$$

The MEC system assisted by UAVs do have bandwidth limitations, thus an UAV at any stop position can serve only up to MI IoT devices and is given in below constraint.

$$\text{Constraint 3}: \ \sum_{i=1}^{t} E_{ij} \leq MI, j \in S. \tag{5}$$

The below constraint is used to confirm that the MEC system assisted by UAVs provides computing services to all the IoT devices.

$$\text{Constraint 4}: \ \sum_{i=1}^{t} \sum_{j=1}^{s} E_{ij} = t \tag{6}$$

The data rate to send data of a process from the IoT device to the UAV at a stop position is given below.

$$q_{ij} = W \log_2 \left(1 + \frac{u_i^t v_{ij}}{\sigma^2} \right) = W \log_2 \left(1 + \frac{u_i^t v_0}{\sigma^2 d_{ij}^2} \right) \tag{7}$$

where the variable W denotes the system bandwidth. The transmission power is represented through u_i^t. The channel power gain is given by v_{ij} and the channel power gain at one unit reference distance is represented as v_0; the white gaussian noise is given by σ^2.

The time taken by an IoT device to transmit A_i number of input information to the UAV at j^{th} stop position is given below.

$$T_{ij}^a = \frac{A_i}{q_{ij}} \tag{8}$$

Energy consumption of a single i^{th} IoT device for data transmission is given by

$$N_{ij}^t = u_i^t T_{ij}^t = \frac{u_i^t A_i}{q_{ij}} \tag{9}$$

The total energy consumption of all the IoT devices is represented in below equation.

$$N_{IoT} =: \sum_{i=1}^{t} \sum_{j=1}^{s} E_{ij} N_{ij}^t \tag{10}$$

The time taken by UAV for completing the process of a i^{th} IoT device is given in below equation.

$$T_i^c = \frac{A_i B_i}{CP} \tag{11}$$

where the variable CP in above equation is the UAV's CPU processing speed.

The UAV will hover at a stop position (SP) j until all the process of the IoT devices at the SP is completed, thus hovering time is represented as T_j^h.

$$T_j^h = max_{i \in T} \left\{ E_{ij} T_{ij}^a + E_{ij} T_j^c \right\} \tag{12}$$

Next, the energy consumption of UAV while the entire hover time is given in (13). Where p^h denotes hovering power.

$$N^h = \sum_{j=1}^{s} p^h T_j^h \tag{13}$$

The next crucial factor is the time taken by the UAV to fly from one stop position j to the other stop position $(j+1)$ after process completion at j. This flight time is given in the below Equation (14) considering that UAV is flying at a constant speed r and at prefixed constant height.

$$T_j^f = \frac{\sqrt{\left(L_{j+1} - L_j\right)^2 + \left(M_{j+1} - M_j\right)^2}}{r} \tag{14}$$

Thus, the energy consumption during the above flight time is calculated using Equation (15). Where the UAV's flight power is given by p^f.

$$N^f = \sum_{j=1}^{s-1} p^f T_j^f \tag{15}$$

The UAV's overall energy consumption is calculated by,

$$N_{UAV} = N^h + N^f \tag{16}$$

and the energy consumption factor of the MEC system assisted by UAVs is

$$N_{MEC} = N_{UAV} + \lambda N_{IoT} \tag{17}$$

where the weight parameter λ is used to ensure that the energy consumption related to IoT devices N_{IoT} is smaller than the UAV's energy consumption metric N_{UAV} and is $\lambda \geq 0$.

As discussed earlier, the two objectives taken for optimization are minimizing the energy consumed by the MEC system N_{MEC} and to minimize the process emergency indicator PE_{Task}. Considering the various process emergency indicator associated with each process and the order of the stop position, the final process emergency indicator PE_{Task} is estimated as below.

$$PE_{Task} = \sum_{i=1}^{t} jC_i, j = argmin_{j \in S}\ d_{ij} \tag{18}$$

In the above equation we estimate sum of products of the process emergency indicator and j. Where j indicates the sequence value of the UAV's stop position; the sequence value is derived using $j = argmin_{j \in S}\ d_{ij}$, representing the nearest distance to an i^{th} device. Lesser the value of PE_{Task}, faster the emergency process is handled.

To summarize, the multi-objective optimization problem in the MEC system assisted with single UAV is given below.

$$\begin{gathered} min_{(L_j, M_j, H), s}\{N_{MEC}, PE_{Task}\} \\ \text{Subject to}: \text{ Constraint 1}: \ E_{ij} \in \{0, 1\}, \\ \text{Constraint 2}: \ \textstyle\sum_{j=1}^{s} E_{ij} = 1, \ \ i \in T, \\ \text{Constraint 3}: \ \textstyle\sum_{i=1}^{t} E_{ij} \leq MI, \ \ j \in S, \\ \text{Constraint 4}: \ \textstyle\sum_{i=1}^{t} \textstyle\sum_{j=1}^{s} E_{ij} = t, \end{gathered} \tag{19}$$

$$\text{Constraint 5}: \ l_{min} \leq L_j \leq l_{max},$$

$$\text{Constraint 6}: \ m_{min} \leq M_j \leq m_{max},$$

where the UAV's stop position's location coordinate set $\{(L_j, M_j) \mid j = 1, 2, .., s\}$ and the number of stop positions s, are to be optimized.

3. Proposed STO Algorithm

The multi-objective optimization problem as given in Equation (19) for the MEC system assisted by UAVs is a challenging environment, since the number of SP and their locations are considered to be unknown. The traditional gradient descent and other related optimization methods are not suitable in handling such problems. Evolution-based optimization algorithms are derivative-free methods and do not require gradient information. Thus, one such evolution-based multi-objective optimization algorithm is extended to solve the discussed problem. The proposed algorithm is named as, self-adaptive trajectory optimization (STO) algorithm. STO is a DE [7]-based algorithm improved by including techniques such as, using a pool of trial vector generation strategies [9] and its adaptation, fuzzy-based control parameter adaptation [10] and an improved encoding scheme [11] to maintain uniformity in length of population individuals. The above inclusions make the proposed STO algorithm robust and suitable to solve the complex multi-objective optimization problem in the MEC system assisted by UAVs.

3.1. Work Flow Model of STO

The process involved in optimization using STO algorithm is illustrated using a flowchart as given in Figure 1. The subsequent section details the steps involved in the process, and the proposed STO algorithm is given in Algorithm 1.

Algorithm 1: STO Algorithm.

1: **Initialization of the dimension of the problem**, here it is the stop position count (s), the **control parameters** NP, CR, F associated with DE algorithm and the other metrics associated in the system. Generation count G is set at 1.
2: **Population initialization** of NP individuals represented as $P-POP$ and its fitness evaluation.
3: While the termination criteria is not met,
Do
4: **Perform non-dominated sorting** on $P-POP$ and select a random non-dominated solution from the first front denoted as $P-Best$.
5: **Apply cutting/padding encoding scheme** on $P-POP$ and obtain $N-POP$ (To ensure all individuals in the population are of same length).
6: Find the **population diversity** and input the diversity error value to the fuzzy system.
7: **Trial vector generation** through mutation, crossover techniques and the resultant population is called as $Q-POP$.
8: **Selection strategy** using a population updating technique to optimize the number of SP and to identify the fittest NP solutions.
9: Increase the generation count.
10: End while.

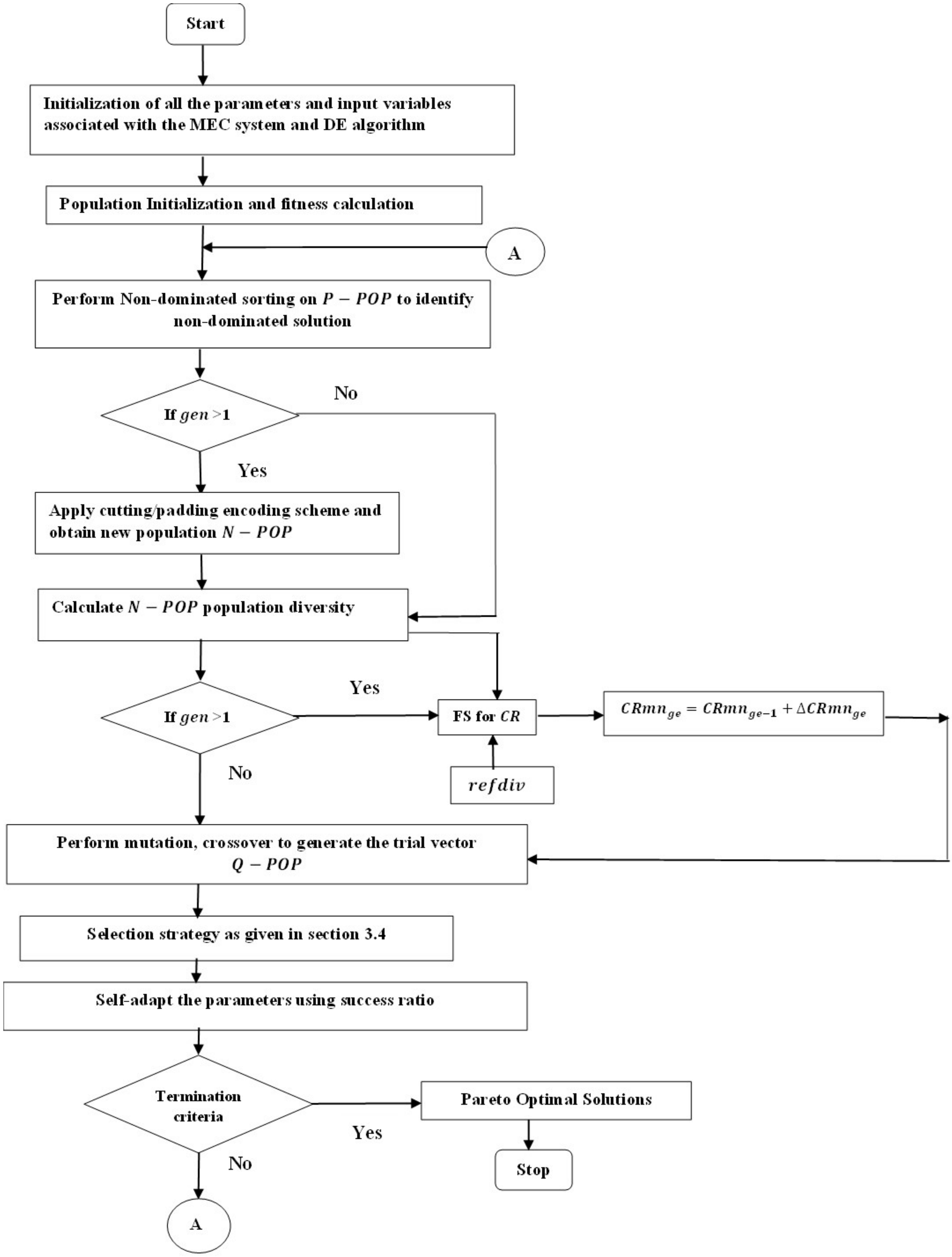


Figure 1. Flowchart of STO algorithm.

3.2. Population Initialization

A population of NP individuals are initialized randomly subject to variable bounds covering the complete search space. EA-based techniques start its search using this random population set and tries to optimize the same within a prespecified limit. Each individual in the population actually represents the number of stop positions (SPs). Since in the present work we consider that the number of SP is unknown, the IoT device count (t) is taken as SP count (s) and each individual in initial population will be of size $2s$, which holds the location coordinates (L, M) of the SP. The height at which UAV hovers while communication with IoT device is fixed and it is used while fitness evaluation.

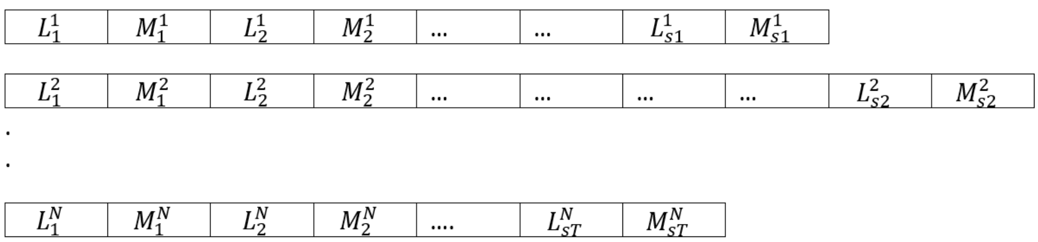
3.3. Trial Vector Generation Using Strategy Adaptation and Cutting/Padding Encoding Scheme

In DE-based algorithm, there are numerous strategies available in literature for trial vector generation. Each strategy is effective in solving one or more optimization problem belonging to different domain effectively. A suitable strategy is identified to solve a particular problem using a trial and error method and, it may even be unsuitable in solving related problem. Thus, in the present work a pool of strategies are included for trial vector generation.

3.3.1. Encoding Scheme Using Cutting/Padding

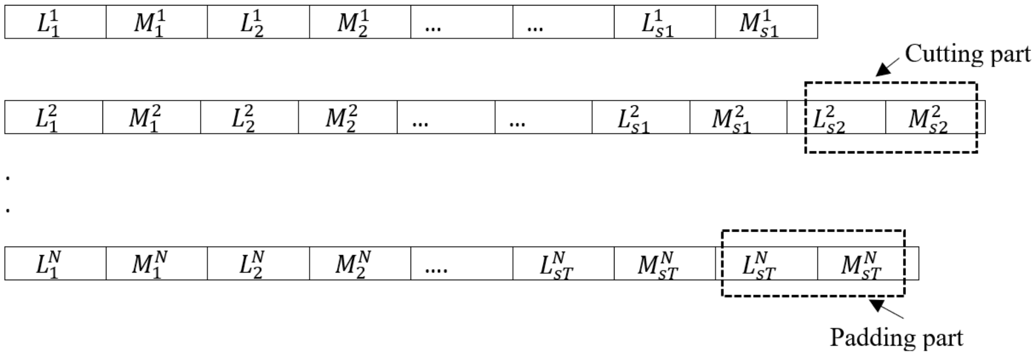
In the MEC system assisted by UAVs, we have attempted to solve multi-objective optimization problem where the number of stop positions (SPs) is considered to be unknown. The length of each individual in the population represents the location coordinates of SP which is the number of decision variables. Since the number of SP, its location coordinates and the order of SP is unknown, the length of each population individual will differ and thus performing mutation and crossover is difficult as they require uniformity in length of all individuals. Thus, the cutting/padding encoding scheme [11] is used to maintain the uniformity.

Figure 2a represents individuals in the population with varying lengths using the regular encoding scheme and Figure 2b exhibits the working of cutting/padding encoding scheme. Where $S_1, S_2, ..S_T$ represents the number of SP of the population individuals, respectively. To perform cutting/padding, non-dominated sorting [22] is performed over the $P-POP$ population, and the population best solution P_{Best} is selected randomly from the first front. In the example shown in Figure 2b assume the first solution is P_{Best} and the number of SP in it is S_1. With this S_1 as the reference, the other remaining solutions are subject to either cutting or padding. If the length of solution exceeds the length of P_{Best}, then the excess part is removed and if the length is less than length of P_{Best}, then to maintain uniformity in length, the last SP in the solution is repeated and added until the length is equal to length of P_{Best} and this is the padding scheme. The result population set $N-POP$ has NP solutions all are of equal length.



(**a**)

Figure 2. *Cont.*



(**b**)

Figure 2. (**a**) Regular encoding scheme; (**b**) Cutting/Padding encoding scheme.

3.3.2. Trial Vector Generation and Its Adaptation

The trial vector/offspring generation in DE-based algorithms are performed using mutation, crossover operations and there are several trial vector generation strategies available in literature. Each of them performs well in optimizing a one or more domain specific problems. Their performance may vary even during various evolution stages. Thus, trying to identify and chose a particular trial vector generation strategy is purely a trial and error mechanism which is time consuming. The chosen strategy and values set for associated control parameters of DE algorithm like the size of population (NP), the crossover rate (CR), and the scaling factor (F) may perform well for that particular problem and may be inefficient in solving other problems.

Thus, strategy pool [9] with various trial vector generation strategies each with varied characteristics are included and are self-adapted based on success index. In present work four different strategies 'DE/rand/1/bin', 'DE/rand/2/bin', 'DE/current-to-rand/1', and 'DE/rand-to-best/2/bin' are all included in the strategy pool, where the solution P_{Best} is the best solution used in strategy 'DE/rand-to-best/2/bin'.

After the initial learning phase (LPE) generations the performance of each of the strategy is assessed and most promising strategies in efficiently solving a particular problem are chosen with higher probability for trial vector generation in upcoming generations. Here, the success index of a strategy is estimated such as, a solution generated by using a strategy entering the next generation after the update technique is a promising solution and the strategy is successful. If a strategy generates much of such promising solutions, its success index will be high and the probability of using such strategy will be more. The probability of choosing a strategy st in generation ge shown below.

$$P_{st,ge} = \frac{Su_{st,ge}}{\sum_{st=1}^{ST} Su_{st,ge}} \tag{20}$$

where

$$Su_{st,ge} = \frac{\sum_{ge=ge-LPE}^{ge-1} nsu_{st,ge}}{\sum_{ge=ge-LPE}^{ge-1} nsu_{st,ge} + \sum_{ge=ge-LPE}^{ge-1} nfa_{st,ge}} + \in; \tag{21}$$

where $st = 1, 2, ..ST$; $ge > LPE$.

The number of promising solutions produced by strategy st chosen for next generation after the updating method is given by $nsu_{st,ge}$ and the count $nfa_{st,ge}$ is the number of solutions generated by strategy st that are not selected in the population selected for subsequent generation; and a constant $\in$ is set at value 0.01 to make sure that none of the strategy has zero success rate. Thus, through the above method, trial vectors are generated.

3.3.3. Fuzzy System-Based Control Parameter Adaptation

The three vital control parameters in DE algorithm are size of population (NP), the crossover rate (CR), and the scaling factor (F). NP depends on the problem. CR value decides in each solution the component count that are to be mutated relies on the complexity of the problem and the convergence is based on the values of F. F value in the present work is generated using a normal distribution $N(0.5, 0.3)$ representing the mean and standard deviation, respectively, and it is ensured that F value is between range [0.4, 1]. The crossover rate CR is adapted based on the population diversity using a Fuzzy Inference System (FIS) [10].

DE algorithm can perform well only if the population are diverse enough, else the evolution will end up in stagnation of the population or premature convergence. In the present work, the population diversity is found using the 'distance-to-average point' metric [23] as:

$$popdiv_{ge}(POP) = \frac{1}{|SLD| * NP} * \sum_{n=1}^{NP} \sqrt{\sum_{g=1}^{PD_n} (z_{ng} - \overline{z}_g)^2} \tag{22}$$

where $|SLD|$ is the search space length measured diagonally using $\sqrt{\sum (z_{max} - z_{min})^2}$ and each search variable z are bound to $z_{min} < z < z_{max}$. POP is the current population of NP solutions. PD_n is the dimensionality of the problem. z_{ng} is the g^{th} value of the n^{th} solution and $\overline{z}_g$ is the g^{th} value of the variable's value $\overline{z}$.

Fuzzy system is used to adapt the cross over rate mean $CRmn$. CR values are generated using the normal distribution $N(CRmn_{st}, std)$, the standard deviation is set at value 0.1, $CRmn_{st}$ is the crossover rate mean for strategy st and is set at value 0.5 for all the strategies during the initial learning phase (LPE). After LPE, the value for $CRmn_{st}$ is calculated as,

$$CRmn_{st} = CRmn_{s,st} + CRmn_{d,st} \tag{23}$$

The variable $CRmn_{s,st}$ is the median of the CR values of strategy st that generated promising solution during the previous (LPE) generations. The second component $CRmn_{d,st}$ is the mean value obtained using fuzzy system considering POP diversity. Through the above adaptations the crossover rate mean values are adapted and crossover rate value CR are generated within the range [0, 1].

The input to the fuzzy system is the diversity difference (Δerr_{ge}) calculated as,

$$\Delta err_{ge} = refdiv - popdiv_{ge} \tag{24}$$

where $popdiv_{ge}$ is the population diversity of the current generation population and $refdiv$ is the referral diversity value which is the actual expected diversity value set by the user.

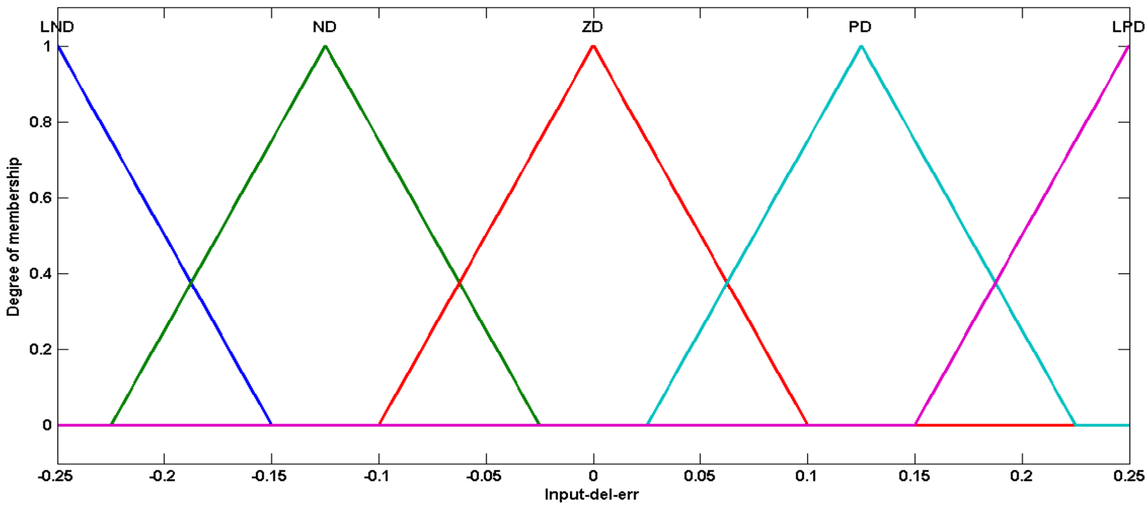
This diversity difference is passed as fuzzy input and the FS maps and gives the changes to be made in the crossover rate mean $(\Delta CRmn_{st}^{ge})$ to improve the population diversity to match with the referral value. Thus, using the fuzzy output the crossover rate mean for generation ge is found as,

$$CRmn_{d,st}^{ge} = CRmn_{st}^{ge-1} + \Delta CRmn_{st}^{ge} \tag{25}$$

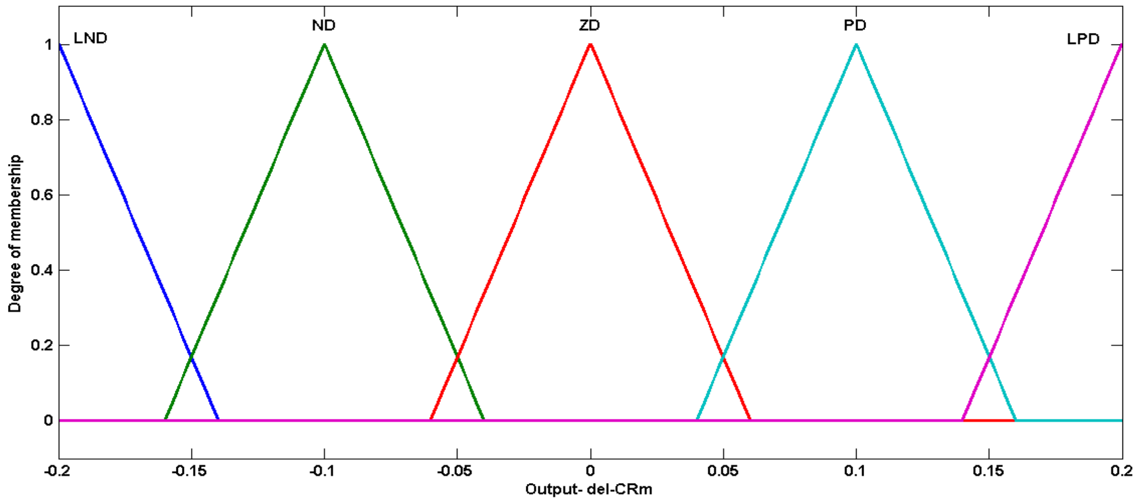
The above $CRmn_{d,st}^{ge}$ value is the second variable used in Equation (23). The crossover rate values adapted using the fuzzy system as above helps to improve the population diversity to enhance the exploration. The fuzzy variables used to define the input and output metrics are zero deviation (ZD), negative deviation (ND), large negative deviation (LND), positive deviation (PD), and the large positive deviation (LPD). The variables are defined as triangular function to establish membership. Centroid technique is utilized for defuzzification. The ruleset is given in Table 1 and the fuzzy input and output variables are given in Figure 3.

Table 1. Ruleset for fuzzy system.

Input—Diversity Difference	ZD	ND	LND	PD	LPD
Output—Changes in crossover rate mean	ZD	ND	LND	PD	LPD



(**a**)



(**b**)

Figure 3. (**a**) Fuzzy input variables and membership range; (**b**) Fuzzy output variables and membership range.

The fuzzy rules are framed such as, if the diversity difference value is within range of zero deviation, it implies that the current population diversity is in the range of reference diversity. Thus, the output value for the variable changes to be performed to crossover rate mean is also set at zero deviation, no change will be conducted. If the diversity difference value is positive or negative then the crossover rate mean value will also be improved accordingly to meet the required reference diversity level. The trial vector $Q - POP$ is generated through the above strategy adaptation, fuzzy-based crossover rate adaptation and cutting/padding encoding scheme.

3.4. Selection Strategy

After generating the trial vector, the population for the next iteration is selected by further optimizing the count of stop positions (SPs), which directly influences the energy consumption value of the system. An updating technique [11] is used to optimize the count of SP. The target vector $P-POP$ and the trial vector $Q-POP$ both are considered for updating and optimizing SP count. For every $Q-POP_i$ individual in $Q-POP$, three tasks are performed such as:

- Random deletion of a SP and this individual is referred as $Q-POP_i^1$.
- The second task is formed by taking the individual $Q-POP_i$ as such and referred as $Q-POP_i^2$.
- The third task is, for the respective i^{th} position in $Q-POP$, the individual $P-POP_i$ is taken and a SP is randomly added to that individual solution and is referred as $Q-POP_i^3$.
- Thus, for every $Q-POP_i$ we have obtained three variants of the individuals $Q-POP_i^1$, $Q-POP_i^2$, $Q-POP_i^3$. Next, the feasibility of the variants are evaluated.
- If more than one variant is feasible, then the individual variant which is non-dominated and having less SP is taken as $Q_{Success}$.
- If only one variant is feasible, it is taken as $Q_{Success}$.
- If none of the three variants are feasible then $P-POP_i$ is not changed.

Next, if the $Q_{Success}$ is not being dominated by $P-POP_i$ then we replace $P-POP_i$ by $Q_{Success}$. Else, $P-POP_i$ is not changed. Through above updating scheme, we select population of NP solutions $P-POP$.

4. Experiments

In this section, the experimental setup, the test problems and the performance metrics used are detailed. The experimentation is conducted in MATLAB. Each algorithm is executed up to 20 runs for all the test problems conducted independently. An example of MEC system assisted by UAVs is given in Figure 4.

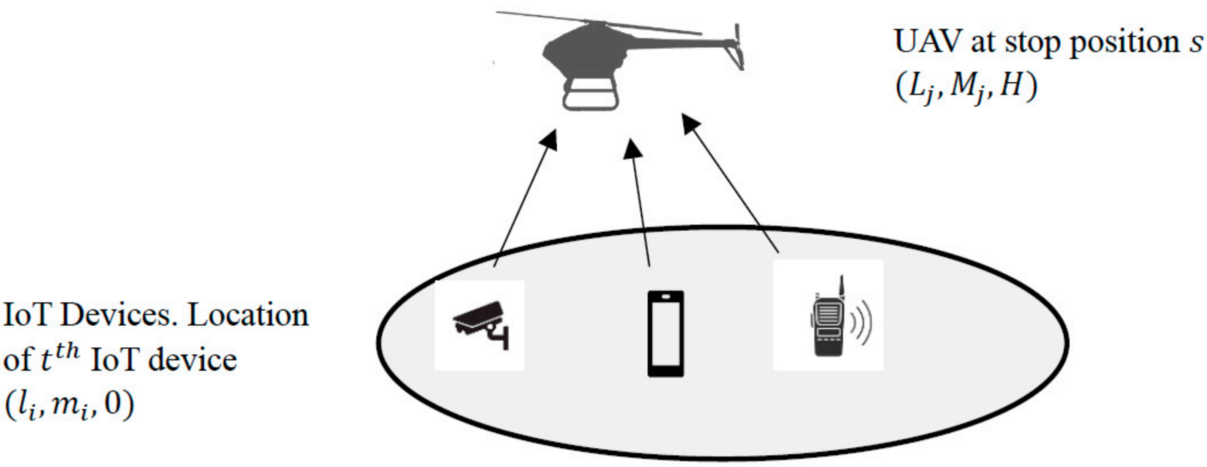


Figure 4. An example of MEC System assisted by UAVs.

4.1. Experimental Setup

The initialization of parameter values involved in the MEC system assisted by UAVs and the parameters related to DE algorithm are all listed in Table 2. The above parameter setting is based on the existing literature listed in Introduction section [11,12,16]. For the other optimization algorithms taken for performance evaluation, the associated parameters are initialized with default values as recommended in related literature.

Table 2. Initial Parameter setup.

Device/MEC System/Algorithm	Parameter	Variable	Initial Value
STO Algorithm (Parameters related to DE)	Crossover rate mean, standard deviation	$(CRmn, std)$	(0.5, 0.1)
	Scaling Factor mean, standard deviation	(Fmn, std)	(0.5, 0.3)
	Learning Period	LPE	50 Generations
	Maximum Number of Function Evaluations	NFE	1000 × t (t = number of IoT devices)
IoT Device	IoT device location coordinates (Devices distributed over a square with 1000 m side length)	(l_i, m_i)	$l_i \in [0,\ 1000]$ m $m_i \in [0,\ 1000]$ m
	Number of input information	A_i	[1, 1000] MB
	Computational resources needed to process one bit	B_i	100 cycles per bit
	Data transmission power	u_i^t	0.1 Watts
	Process emergency indicator	C_i	$C_i \in [1,\ 2,\ 3,\ 4,\ 5]$
MEC System	System bandwidth	W	1 MHZ
	Unit channel power gain	vo	−30 dB
	White Gaussian noise	σ^2	−100 dBm
	Weight parameter	λ	10,000
UAV	The maximum count of IoT devices handled simultaneously	MI	5
	Height-UAV's flying altitude	H	200 m
	UAV's location coordinates	L_j, M_j	$L_j \in [0,\ 1000]$ m $M_j \in [0,\ 1000]$ m
	Constant flying speed of UAV	r	20 m/s
	UAV's CPU processing speed	CP	10 GHZ
	UAV's Battery capacity	B_CAP	28,000 mAh
	UAV's Battery voltage	B_VO	51.8 volts
	UAV's hovering power	p^h	1400 watts
	UAV's flight power	p^f	1600 watts

The UAV'S battery energy B_EN is computed as:

$$B\_EN = \frac{B\_CAP}{1000} \times 3600 \times B\_VO \tag{26}$$

where B_CAP is UAV'S battery capacity, B_VO is UAV'S battery voltage and B_EN value through the above equation is computed as 5,221,440 J.

4.2. Test Problem

The multi-objective optimization problem for the MEC system assisted by UAVs is detailed in Section 2, based on which ten test problems are generated by varying the number IoT devices t, considered in the MEC system. The number of IoT devices in each problem are initialized at [60, 80, 100, 120, 140, 160, 180, 200, 300, 400].

4.3. Performance Measure I—Quality Indicator (Hypervolume)

The efficiency of the proposed STO algorithm in solving the optimization problem related to MEC system is tested using the hypervolume (HV) metric [24,25]. Since it is a real world problem and the true front for the problem is not known, HV is used as quality indicator. HV metric can assess both the convergence and diversity metrics of the attained solutions. Moreover, the problem is a minimization problem where both the objective values are to be minimized. An algorithm is claimed to perform well in solving a particular problem if the value of HV is high. The hypervolume metric is calculated as:

$$\text{HV} = Le\left(\cup_{X \in QN} [f_1(X), re_1] * \ldots [f_m(X), re_m]\right) \tag{27}$$

where QN is the nondominated solution set, Le is the Lebesgue measure. $Re = (re_1,\ re_2, .., re_m)$ are reference points and m is the number of objectives in the problem.

4.4. Performance Measure II—Statistical Test

To further prove the potential of the proposed STO algorithm, the results are analyzed statistically. Friedman test [26] is a non-parametric statistical test which jointly analyzes the results obtained through different algorithms taken for comparison. SPSS—statistical software package is used to perform the Friedman test and the significance level α is set at value 0.05. If the obtained $p - value$ through above test is less than α then the null hypothesis can be rejected and its shows significance of STO algorithm than other algorithms. Else, if $p > \alpha$ then there is no stronger evidence to prove that the proposed algorithm is better than remaining algorithms taken for comparative study.

5. Results and Analysis

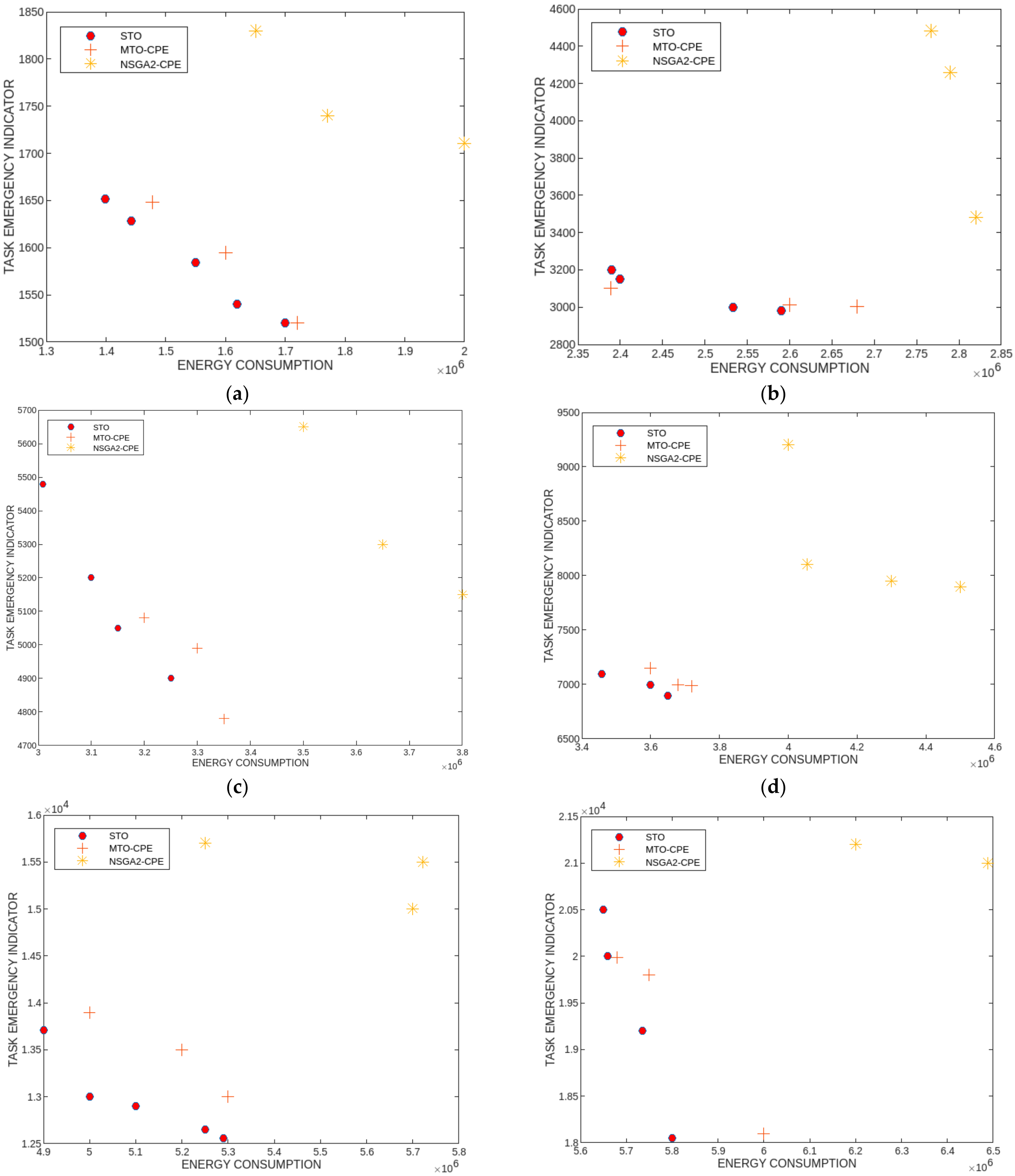
In this section, the performance of the proposed STO algorithm is compared with other competitive algorithms and the results are discussed using the hypervolume performance metric and through statistical tests. The algorithms taken for comparison are NSGA-II [22], MTO-CPE [11], and NSGA-II-CPE. Where, NSGA-II is a classic and representative multi-objective evolutionary algorithm. MTO-CPE is a DE-based algorithm developed using the cutting-padding encoding scheme and the updating mechanism to solve the optimization problem in the MEC system assisted by UAVs. NSGA-II-CPE is a variant of NSGA-II by including the cutting-padding encoding scheme and the updating mechanism to optimize the number of stop positions. All the algorithms are executed 20 independent runs and the mean standard deviation of HV values are given in Table 3. Higher the HV value, better the performance of the algorithm.

Table 3. Results of HV performance metric (mean and standard deviation) attained by STO, NSGA-II, NSGA-II-CPE, and MTO-CPE. The best HV results are in bold.

Test Problem with t Value	STO	NSGA-II	NSGA-II-CPE	MTO-CPE
60	**0.5118 ± 0.00542**	0.165 ± 0.00441	0.473 ± 0.00880	0.510 ± 0.00515
80	**0.478 ± 0.00785**	0.122 ± 0.00386	0.399 ± 0.0109	0.433 ± 0.00845
100	**0.465 ± 0.00410**	0.0876 ± 0.00569	0.372 ± 0.0118	0.412 ± 0.00348
120	**0.412 ± 0.00784**	0.0840 ± 0.00919	0.331 ± 0.0183	0.378 ± 0.00850
140	**0.0345 ± 0.0210**	0.0557 ± 0.00428	0.269 ± 0.00702	0.0300 ± 0.0101
160	**0.328 ± 0.0211**	0.0486 ± 0.00732	0.256 ± 0.0114	0.293 ± 0.0100
180	**0.487 ± 0.00914**	0.0657 ± 0.00172	0.264 ± 0.00509	0.327 ± 0.00825
200	**0.289 ± 0.00841**	0.0431 ± 0.00454	0.245 ± 0.00852	0.283 ± 0.00970
300	**0.278 ± 0.0246**	0.0337 ± 0.00315	0.176 ± 0.00547	0.243 ± 0.0148
400	**0.187 ± 0.00467**	0.0239 ± 0.00318	0.149 ± 0.00191	0.166 ± 0.00501

From the HV results it is evident that the STO algorithm is efficient in solving all the test problems than the other algorithms taken for comparison. Another observation from the above results, as the count of IoT device increases the HV value decreases for all the algorithms. The fuzzy system-based self-adaptation of control parameters by considering the population diversity makes the STO algorithm robust in handling such complex real world optimization problem. The strategy pool of trial vector generation strategies and its adaptation, crossover rate adaptation using fuzzy system makes the DE algorithm suitable to solve the real world complex optimization problem without extensive fine tuning. Moreover, the other related problems can also be solved using the algorithm with less modifications related to the problem.

Figure 5 illustrates the Pareto fronts obtained using the STO and the other algorithms using the cutting/padding encoding scheme. It is evident that the solutions obtained using the STO dominate the solutions attained using other algorithms.



(**a**) (**b**)

(**c**) (**d**)

(**e**) (**f**)

Figure 5. *Cont.*

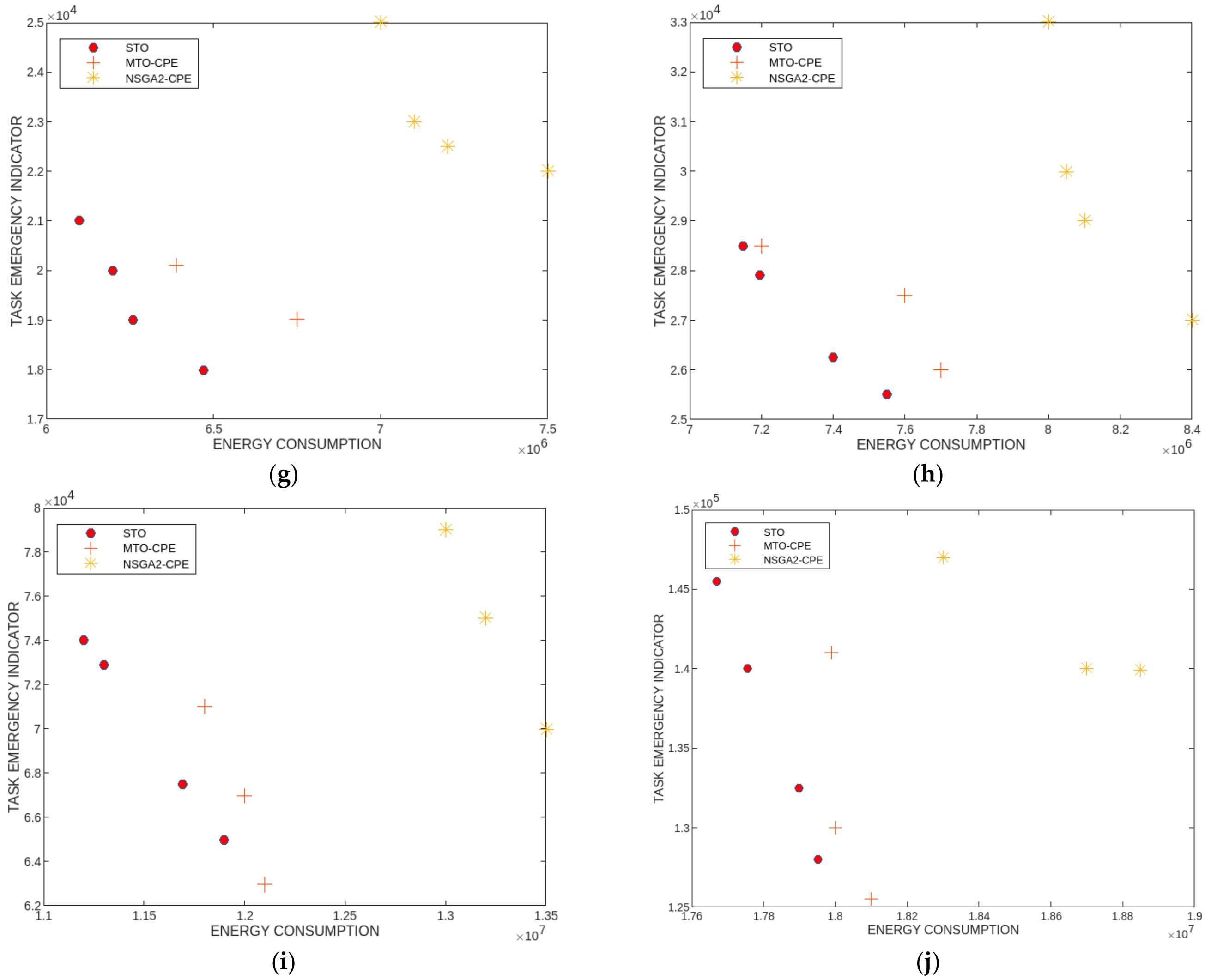


Figure 5. Pareto Fronts obtained using STO and other algorithms for varying IoT device count (**a**) t = 60; (**b**) t = 80; (**c**) t = 100; (**d**) t = 120; (**e**) t = 140; (**f**) t = 160; (**g**) t = 180; (**h**) t = 200; (**i**) t = 300; (**j**) t = 400.

Figure 6 gives the trial vector generation strategy adaptation for the test problem with number of IoT device t value as 300, over the various stages of evolution, where the strategies are self-adapted according to the success index. The strategies that are successful in generating promising solutions entering the next generation are chosen with higher probability in upcoming generation for trial vector generation. All the strategies except DE/current-to-rand/1 are chosen with higher probability. The reason is DE/current-to-rand/1 strategy doesn't involve any crossover operator. This shows the effectiveness of the proposed algorithm where the strategies and the crossover rate are self-adapted according to the population diversity through fuzzy system and attain promising solutions. Thus, the success index of the other strategies are higher.

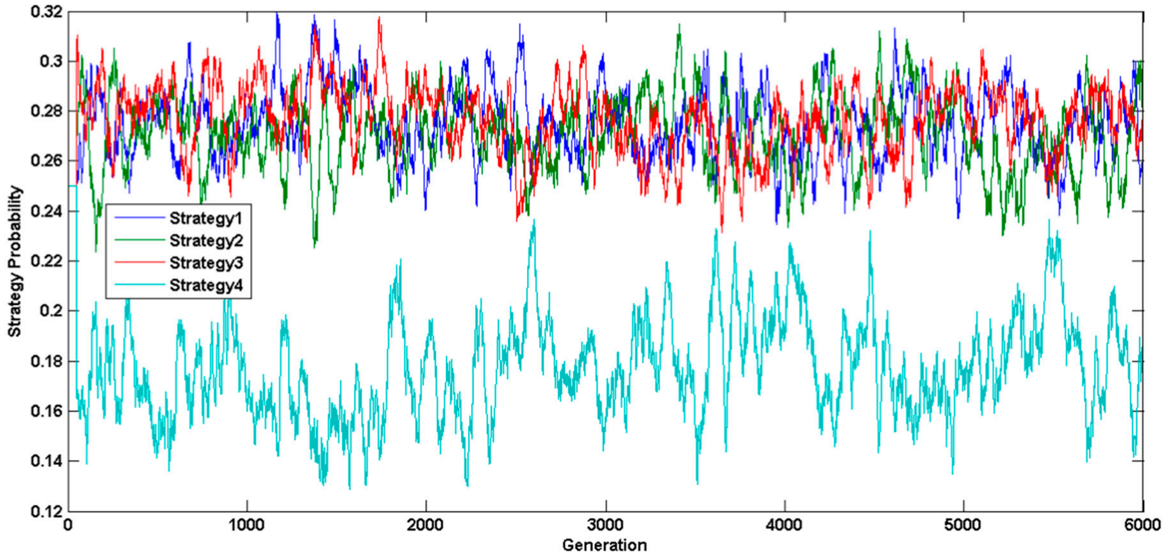


Figure 6. Trial vector generation strategies and its adaptation.

Figure 7 represents the adaptation of the CR values over the generations for the test problem with number of IoT device t value as 300. The crossover rate values for the strategies are gradually adapted according to the environment which helps to escape from the population stagnation and premature convergence. This self-adaptation of CR values by improving the population diversity is one among the vital reasons for better optimization results.

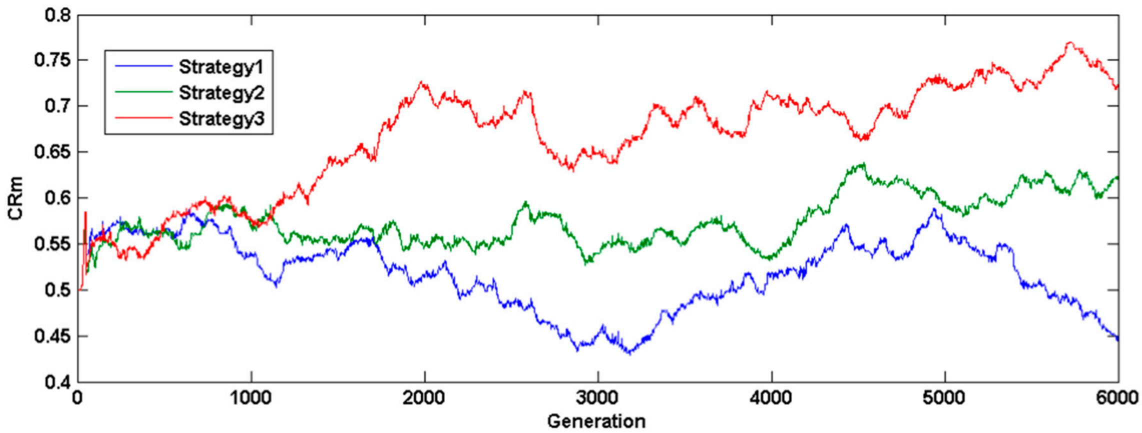


Figure 7. Crossover rate adaptation for the strategies.

To further strengthen the discussion on the present algorithm, the Friedman statistical test is conducted. The significance level α value is set at 0.050. The test results are shown in Figure 8. STO algorithm is ranked first among other algorithms in the test. The obtained ρ value is 0.001, which is less than 0.05. Thus, the null hypothesis can be rejected which statistically proves the significance of the present work.

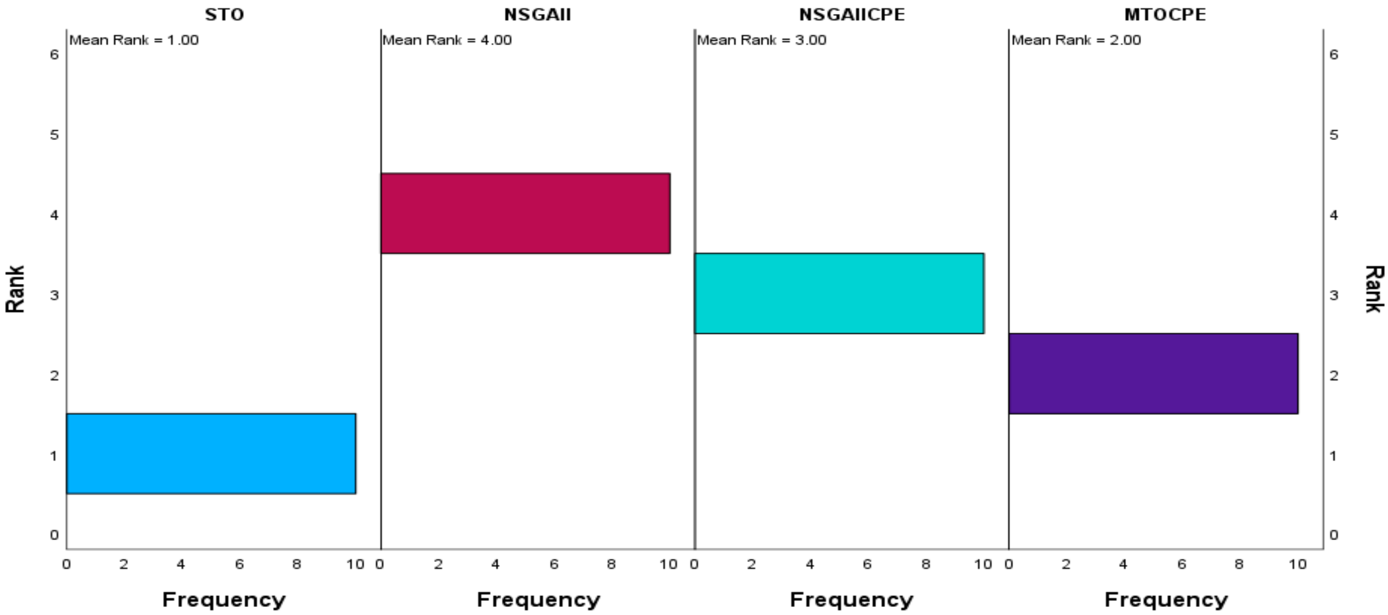


Figure 8. Friedman test results.

6. Conclusions

In the present research work, the optimization problem related to mobile edge computing system assisted with a single unmanned aerial vehicle is considered. The multi-objective optimization problem is detailed where the objectives are minimizing the energy consumed by the MEC and minimizing the process emergency indicator. To solve the above complex real world problem, a novel algorithm based on DE algorithm is proposed and is named as the self-adaptive trajectory optimization (STO) algorithm. The number of stop positions is considered as unknown, which makes the problem environment challenging. STO algorithm is developed by including a pool of trial vector generation strategies and their self-adaptation using success index. Further, fuzzy system is used to self-adapt the crossover rate associated in DE algorithm by improving the population diversity. STO also uses the cutting/padding encoding scheme to ensure all the individuals in the population are of uniform length. The updating method further aids in optimizing the stop position count. STO algorithm developed using above techniques make it robust in handling the complex optimization problem presented.

The test problems are taken by varying the number of IoT devices. The efficiency of the proposed STO algorithm is proved by comparing the results with other representative evolutionary algorithms. The hypervolume metric value computed through 20 independent runs is presented and the STO algorithm outperforms the results attained by the other algorithms considered. To further strengthen the proposed research work, a statistical test was conducted and the STO algorithm is ranked first in the test. The above results are promising to further leverage the research in this domain by considering multiple UAVs and a larger number of IoT devices resembling the challenging MEC environment.

Author Contributions: Conceptualization, B.S.; methodology, B.S., U.M.J. and V.A.; software, B.S. and M.J.A.S.A.; validation, B.S. and U.M.J.; formal analysis, B.S. and V.A.; writing—original draft preparation, B.S.; writing—review and editing, B.S.; supervision, B.S. and M.J.A.S.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: No publicly available data were used or generated.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Mao, Y.; You, C.; Zhang, J.; Huang, K.; Letaief, K.B. A survey on mobile edge computing: The communication perspective. *IEEE Commun. Surv. Tutor.* **2017**, *19*, 2322–2358. [CrossRef]
2. Abbas, N.; Zhang, Y.; Taherkordi, A.; Skeie, T. Mobile edge computing: A survey. *IEEE Internet Things J.* **2017**, *5*, 450–465. [CrossRef]
3. Kawamoto, Y.; Nishiyama, H.; Fadlullah, Z.M.; Kato, N. Effective data collection via satellite-routed sensor system (SRSS) to realize global-scaled Internet of Things. *IEEE Sens. J.* **2013**, *13*, 3645–3654. [CrossRef]
4. Ahmed, F.; Mohanta, J.C.; Keshari, A.; Yadav, P.S. Recent Advances in Unmanned Aerial Vehicles: A Review. *Arab. J. Sci. Eng.* **2022**, *47*, 7963–7984. [CrossRef]
5. Zeng, Y.; Zhang, R.; Lim, T.J. Wireless communications with unmanned aerial vehicles: Opportunities and challenges. *IEEE Commun. Mag.* **2016**, *54*, 36–42. [CrossRef]
6. Michalewicz, Z. *Genetic Algorithms + Data Structures = Evolutionary Programs*; Springer: Berlin, Germany, 1996.
7. Storn, R.; Price, K. Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces. *J. Glob. Optim.* **1997**, *11*, 341–359. [CrossRef]
8. Kennedy, J.; Eberhart, R. Particle swarm optimization. In Proceedings of the International Conference on Neural Networks, Perth, Australia, 27 November–1 December 1995; pp. 1942–1948.
9. Huang, P.; Wang, Y.; Wang, K.; Yang, K. Differential evolution with a variable population size for deployment optimization in a UAV-assisted IoT data collection system. *IEEE Trans. Emerg. Top. Comput. Intell.* **2019**, *4*, 324–335. [CrossRef]
10. Wang, Y.; Ru, Z.; Wang, K.; Huang, P. Joint deployment and task scheduling optimization for large-scale mobile users in multi-UAV-enabled mobile edge computing. *IEEE Trans. Cybern.* **2019**, *50*, 3984–3997. [CrossRef]
11. Alzenad, M.; El-Keyi, A.; Lagum, F.; Yanikomeroglu, H. 3D placement of an unmanned aerial vehicle base station (UAV-BS) for energy efficient maximal coverage. *IEEE Wirel. Commun. Lett.* **2017**, *6*, 434–437. [CrossRef]
12. Li, L.; Wen, X.; Lu, Z.; Pan, Q.; Jing, W.; Hu, Z. Energy-efficient UAV-enabled MEC system: Bits allocation optimization and trajectory design. *Sensors* **2019**, *19*, 4521. [CrossRef]
13. Huang, P.; Wang, Y.; Wang, K. Energy-efficient trajectory planning for a multi-UAV-assisted mobile edge computing system. *Front. Inf. Technol. Electron. Eng.* **2020**, *21*, 1713–1725. [CrossRef]
14. Lai, S.; Zhao, R.; Tang, S.; Xia, J.; Zhou, F.; Fan, L. Intelligent secure mobile edge computing for beyond 5G wireless networks. *Phys. Commun.* **2021**, *45*, 101283. [CrossRef]
15. Cheng, Y.; Liao, Y.; Zhai, X. Energy-efficient resource allocation for UAV empowered mobile edge computing system. In Proceedings of the 2020 IEEE/ACM 13th International Conference on Utility and Cloud Computing, UCC, Leicester, UK, 7–10 December 2022; pp. 408–413.
16. Mozaffari, M.; Saad, W.; Bennis, M.; Debbah, M. Mobile Unmanned Aerial Vehicles (UAVs) for Energy-Efficient Internet of Things Communications. *IEEE Trans. Wirel. Commun.* **2017**, *16*, 7574–7589. [CrossRef]
17. Du, W.; Ying, W.; Yang, P.; Cao, X.; Yan, G.; Tang, K.; Wu, D. Network-based heterogeneous particle swarm optimization and its application in UAV communication coverage. *IEEE Trans. Emerg. Top. Comput. Intell.* **2019**, *4*, 312–323. [CrossRef]
18. Dai, H.; Zeng, X.; Yu, Z.; Wang, T. A scheduling algorithm for autonomous driving tasks on mobile edge computing servers. *J. Syst. Archit.* **2019**, *94*, 14–23. [CrossRef]
19. Qin, A.K.; Huang, V.L.; Suganthan, P.N. Differential evolution algorithm with strategy adaptation for global numerical optimization. *IEEE Trans. Evol. Comput.* **2009**, *13*, 398–417. [CrossRef]
20. Brindha, S.; Amali, S.M.J. A robust and adaptive fuzzy logic based differential evolution algorithm using population diversity tuning for multi-objective optimization. *Eng. Appl. Artif. Intell.* **2021**, *102*, 104240.
21. Jianqing, L.; Linqiang, P. Multiobjective trajectory optimization with a cutting and padding encoding strategy for single-UAV-assisted mobile edge computing system. *Swarm Evol. Comput.* **2022**, *75*, 101163.
22. Deb, K.; Pratap, A.; Agarwal, S.; Meyarivan, T. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.* **2002**, *6*, 182–197. [CrossRef]
23. Ursem, R.K. Diversity-guided evolutionary algorithms. In Proceedings of the Seventh International Conference on Parallel Problem Solving from Nature VII (PPSN-2002), Granada, Spain, 7–11 September 2022.
24. Zitzler, E.; Thiele, L. Multiobjective optimization using evolutionary algorithms—A comparative case study. In Proceedings of the International Conference on Parallel Problem Solving from Nature, Amsterdam, The Netherlands, 27–30 September 1998; Springer: Berlin/Heidelberg, Germany, 1998.
25. Ishibuchi, K.; Imada, R.; Setoguchi, Y.; Nojima, Y. How to Specify a Reference Point in Hypervolume Calculation for Fair Performance Comparison. *Evol. Comput.* **2018**, *26*, 411–440. [CrossRef]
26. Derrac, J.; García, S.; Molina, D.; Herrera, F. A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm Evol. Comput.* **2011**, *1*, 3–18. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.