

## Article

# EETO-GA: Energy Efficient Trajectory Optimization of UAV-IoT Collaborative System Using Genetic Algorithm

M M Hafizur Rahman <sup>1,\*</sup> , Mohammed Al-Naeem <sup>1</sup> , Anuradha Banerjee <sup>2</sup> and Abu Sufian <sup>3</sup> 

<sup>1</sup> Department of Computer Networks & Communications, CCSIT, King Faisal University, Al Ahsa 31982, Saudi Arabia

<sup>2</sup> Department of Computer Applications, Kalyani Government Engineering College, Maulana Abul Kalam Azad University of Technology, Kalyani 741235, India

<sup>3</sup> Department of Computer Science, University of Gour Banga, Malda 732103, India

\* Correspondence: mrahman@kfu.edu.sa

**Abstract:** Unmanned aerial vehicle (UAVs) is capable of adding significant potential to the internet of thing (IoT) devices and hence smart UAV-IoT collaborative system has attracted the attention of many researchers. This system has to be energy efficient for its nature and functionalities. Optimized trajectory planning is a significant area of research for any automatic movable device. In this article, we propose a technique, called EETO-GA for energy-efficient trajectory optimization of UAV-IoT using a genetic algorithm (GA). This technique prescribes each device of: (i) the next timestamp of arrival on the present cluster of IoT devices, at which the task queue of its header contains the highest possible number of tasks, and (ii) the minimum amount of energy that requires to complete all the tasks present in task queue of the IoT device. This technique uses a GA for optimization where the fitness function is designed by optimizing objectives: (i) the total number of tasks that can be completed, (ii) minimization of consumed energy, and (iii) the number of devices that could be served. A GA is applied here to accommodate a large number of IoT devices. A binary method of encoding is applied and methods like cross-over and mutation are used to arrive at the optimal solution. Through a simulation study, the proposed technique shows significant improvement in terms of UAV energy saved (UAVE), energy saving in IoT devices (IoTDEC), the average delay in execution of the task (ADET), and the percentage of tasks that could be completed (PTSK). Proposed EETO-GA improved average UAVE: 43%, IoTDEC: 56%, PTSK: 7.5%, and ADET: 38% over the state of the art.

**Keywords:** ARMA model; clustered model; delay aware; energy efficient; internet of thing; path planning; trajectory optimization; unmanned aerial vehicles



**Citation:** Rahman, M.M.H.; Al-Naeem, M.; Banerjee, A.; Sufian, A. EETO-GA: Energy Efficient Trajectory Optimization of UAV-IoT Collaborative System Using Genetic Algorithm. *Appl. Sci.* **2023**, *13*, 2535. <https://doi.org/10.3390/app13042535>

Academic Editor: Christos Bouras

Received: 26 December 2022

Revised: 8 February 2023

Accepted: 11 February 2023

Published: 16 February 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

There have been numerous applications of unmanned aerial vehicles (UAVs) in the last decade in the fields of wireless ad hoc and sensor networks, vehicular networks, and the internet of thing (IoT) devices [1–4]. The applications encapsulated various fields including civilian purposes, military, disaster management, surveillance, source localization, task execution, products delivery, agriculture, etc. [5–9]. The reason is that UAVs are fast, lightweight, and cheap. If their trajectories are planned properly then a lot of tasks can be completed through energy-efficient UAVs and less energy consumption means a lot of money is saved [10–14]. This is entirely necessary from the perspective of the scarcity of energy resources and the sustainability of mother earth [15–19].

The target in this context is energy optimization in UAV-controlled IoT-monitored big agricultural fields, surveillance areas, etc., through energy efficient trajectory planning and properly suggesting the IoT devices about the timestamp of submitting the next task to the clusterhead (CH). Although many UAV trajectory deciding algorithms exist in the literature as discussed in Section 2, to the best of the authors' knowledge, the issue of arrival

of the UAV over a CH when its task queue is approximately full, has been inadequately addressed in UAV-IoT systems. But it is extremely important for efficient task execution and resource utilization. It is always better and energy-saving for the UAV to come to a CH and execute the set of tasks in the completely full task queue than coming to the same CH twice and executing the set of tasks in its half-filled task queue. This saves propulsion energy in the UAVs. Energy in non-CH ordinary devices is saved by the CH IoT device as it prescribes the next timestamp of submitting the next task by this particular ordinary IoT device so that the chance of this task being dropped, becomes much less. The job of the CH is to collect the tasks of the individual IoT devices and under it and arrange them in a suitable sequence. This is extremely important because each retransmission of a task from the source IoT device to CH doubles the consumption of energy by ordinary members. The scheduling process has been made energy efficient because tasks are executed in order of priority and starving tasks, especially those that have short lengths are assigned a higher priority. This is an optimization problem where the optimal trajectory to be followed by UAV changes dynamically with a number of tasks that are generated at each individual IoT device, their length, the starvation status, predecessor, etc., until the history is developed, we rely on the initial solution which is the trajectories suggested by the Dijkstra's shortest path algorithm [20]. Iterative improvements take place on existing solutions based on the history that gradually develops in CH IoT devices. This matches with the nature of genetic algorithm (GA) [21,22] and so we have used it in the proposed technique.

In this paper, we propose a GA-based trajectory optimization strategy that will optimize the objectives including the number of distinct IoT devices served, the number of tasks that are successfully completed, and the amount of energy consumed which includes propulsion energy as well as the energy required to execute the task submitted by several different IoT devices. Energy is saved not only in the UAVs but also in the IoT devices by avoiding the resubmission of tasks as much as possible. Here we have considered the constraints like time of use, tariff, etc., because the system has been implemented here using a simulator.

### 1.1. Highlights of this Article

The main highlights of our article are as follows:

- The IoT devices in our system are divided into multiple clusters with each cluster having a CH. A UAV flies over these CHs in such a trajectory that optimizes the number of tasks executed, the number of devices served, along with consumption of energy. Here we have avoided the round-robin mode of task execution because tasks are generally scheduled according to priority.
- Each CH maintains two tables, one corresponding to IoT devices under itself and the other corresponding to other CHs. Information about IoT devices is required to compute the priority of the current task submitted by one particular IoT device whereas cluster information is required to compute the credibility of the CH to be visited next. This trajectory planning of UAV is completely dynamic and with every traversal of CH, training data is refined with the most recent experiences (that is, made updated with the most recent records), so that the next CH to visit can be computed efficiently. Dynamic planning of energy-conserving UAV trajectory is an important contribution of this article.
- Energy is preserved not only during trajectory planning of UAVs but also during task propagation from a non-CH IoT device to its CH. When submitting the next task, this information is supplied to an IoT device by the CH so that its chance of being dropped becomes lower. This preserves the energy that would have been required otherwise for resubmitting the task. This is accomplished using the ARMA model [23] and decision tree [24].
- Use of GA enables incorporating a large number of clusters of IoT devices providing a scalability feature of the system.
- Simulation results demonstrate that our proposed scheme EETO-GA outperforms other state-of-the-art UAV trajectory optimization algorithms in terms of performance

Metrics like UAV energy saved (UAVE), energy saving in IoT devices (IoTDEC), the average delay in execution of the task (ADET), and the percentage of tasks that could be completed (PTSK). Proposed EETO-GA improved average UAVE: 43%, IoTDEC: 56%, PTSK: 7.5%, and ADET: 38% over the state of the art.

### 1.2. Organization of the Article and Description of Used Symbols

The rest of the paper is organized as: Section 2 illustrates literature on this topic. Section 3 illustrates the proposed model (method) including computation of task priority, predicting the time of submission of the next task, and GA-assigned trajectory planning. Simulation experiments, results, and discussion appear in Section 4 whereas Section 5 concludes the paper with leftover a few immediate future scopes.

The paper presents many symbols and abbreviations that are tried to explain the location where they are used. But Table 1 presents a brief description of each symbol and abbreviations together.

**Table 1.** Symbols, Abbreviation, and their Descriptions.

Symbol/Abbreviation	Brief Description
$X_{cen}$	X co-ordinate centroid of the cluster C.
$Y_{cen}$	Y co-ordinate centroid of the cluster.
$num(c)$	Number of devices in cluster C.
$dev(i)$	$i$ th device of IoT devices.
$h_i$	Denotes x latitude point of dev( $i$ ) location.
$k_i$	Denotes y longitude point of dev( $i$ ) location.
$cls\_clg(i)$	This function denotes Eligibility of the $i$ th device dev( $i$ ) as head of its cluster.
$(R - E_i)$	Residual energy of the $i$ th device.
$\tau(j)$	Denotes new task.
$Pr(\tau(j), dev(i))$	Possible priority of a new task $\tau(j)$ submitted by the device dev( $i$ ) of cluster C.
IG	Information Gain.
Comp_priority	Computation priority.
Entropy(Parent)	The entropy of the parent node.
Entropy(children)	The average of entropies of all its children.
$p_i$	Probability of the random selection of an element in class $i$ .
Max_pr	The maximum possible priority in the the system.
Min_pr( $f'$ )	The minimum priority of a task present in the task queue at current time $t$ .
multi_regress	Denotes multiple regression function.
$f'$	This function denotes increase the priority value.
tm_gen( $j$ )	Timestamp of its generation based on current task $\tau(j)$ .
tm_exec( $j$ )	Timestamp of completing execution of the task $\tau(j)$ .
len_tsk( $j$ )	Length of the task where priority of the current task $\tau(j)$ .
$ST(i, t')$	A set of task submission sessions of dev( $i$ ) till current time $t'$ .
Prt $y$	Currently computed priority.
PTSK	Percentage of task that could be completed.
IoTE	Average energy saving in IoT devices.
UAVE	UAV energy saved.
ADET	Average delay in execution of the task.
UAV	Unmanned aerial vehicles.
EETOGA	Energy-Efficient Trajectory Optimization using Genetic Algorithm.
CEC	Comprehensive energy consumption.

**Table 1.** Cont.

Symbol/Abbreviation	Brief Description
$ST \sim starve(i, t')$	The subset of $ST(i, t')$ , which consists of only the nonstarving tasks.
PEM	The minimization of propulsion energy.
TDMA	Time Division Multiple Access.
Prednext	Predict the next suitable value.
$tm\_gen\_next(i, t')$	Generation of the next task of device I at current time $t'$ .
$cls(u)$	The CH closest to the UAV.
$lat(u)$	Denotes the geographical position of the CHs in terms of latitude.
$long(u)$	Denotes the geographical position of the CHs in terms of longitude.
GATE	Genetic algorithm applied for task execution.

## 2. Related Work

These days a huge amount of research works are being carried out in the domain of UAVs and drones in connection with wireless sensor networks, vehicular networks as well as IoT devices because of the possibilities of huge applications expected to enhance the performance of those systems [2–4,25]. When the number of such networks, nodes, or IoT devices increases up to a great extent, these nodes or devices are generally clustered and UAVs have to hover on these CHs. Cluster members submit their tasks to their own CHs which are received by the UAVs that execute those tasks by hovering over the CHs. This hovering costs a huge amount of energy, especially if the CHs are traversed in a zigzag manner [10,26]. Therefore, for energy-efficient operation of the system, the trajectories of the UAVs should be planned properly [4,16]. Additionally, in multi-UAV systems, there may be the intersection of trajectories of different UAVs, leaving a nonzero probability of collision. This collision must be avoided at any cost for which proper trajectory planning is extremely necessary [27–29].

A deep reinforcement learning best trajectory design is presented in [16] where energy efficiency is ensured through minimizing residual energy, safe distance between multiple UAVs, and minimizing the application of age information in the current context. This technique is especially applicable for a multi-UAV system where the authors have given special attention to reducing dependencies of testing data on the training data.

The comprehensive energy consumption (CEC) model is studied in [18,19] where energy consumed by a UAV consists of two components: propulsion energy and communication energy. Communication energy is due to the transmission of tasks from network nodes to UAV and returning back the results from UAV to nodes. Propulsion energy is because of the hovering of UAVs on nodes. However, a lot of propulsion energy is wasted if the task queue is not full (or approximately full) when the UAV arrives over one particular network node for task execution of a collection of nodes under it. The minimization of propulsion energy (PEM) of fixed-wing UAVs is considered by authors in [19] where the communication energy part of UAV's energy consumption is ignored.

These days, gradually, applications of UAV-assisted systems have started to become processing intensive; additionally, the number of real-time tasks has increased for which delay has become an important performance metric along with energy, as proposed in [30]. In order to finish all the tasks within a specific time duration, hovering decisions of UAVs have to be based on recent information; therefore, age-of-information needed to be considered these days as an important criterion [31]. The recency of information is considered by authors of [4] where trajectory planning (AoI-IP) is based on dynamic programming. However, the communication energy part is ignored in this paper as well.

Various learning algorithms have been presented in the literature some of which are limited to only trajectory optimization of single or multiple UAVs [32–34] whereas some others consider trajectory optimization along with scheduling [10,35]. In this article, IoT devices are not clustered, but rather all these devices apply Time Division Multiple Access

(TDMA) to get served on the UAV. However, interference is a big problem in this case, and the time quantum will be small if a huge number of IoT devices are grouped together. Hence, TDMA-based task execution will not be efficient in this scenario.

### 3. Proposed Method EETO-GA

#### 3.1. Model Architecture

In the proposed EETO-GA, the set of IoT devices is divided into certain clusters and each cluster has a CH as shown in Figure 1. The IoT devices send their tasks to their CH where they get stored in a task queue in descending order of priority. The task with the highest priority gets placed in the front end of the queue whereas the one with the least priority is placed in the rear end. Whenever the queue is full, no further tasks are accepted and inserted in the task queue until the UAV arrives on the CH and executes all those.

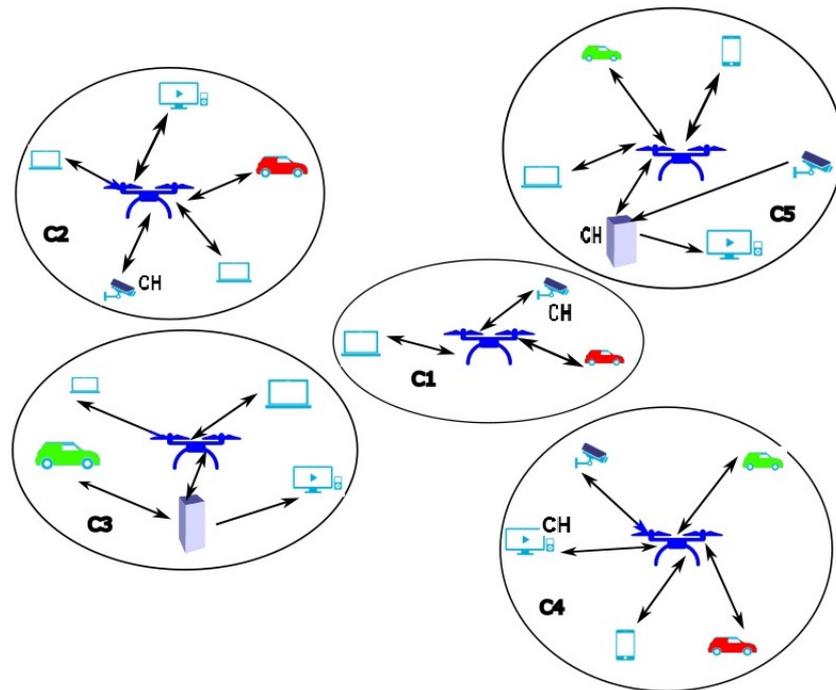


Figure 1. Model Architecture of an UAV-IoT Collaborative System.

#### 3.2. Selection of Clusterhead (CH)

Suppose in cluster  $C$ , IoT devices are denoted as  $dev(1), dev(2), \dots, dev(num(C))$  where  $num(C)$  is the number of devices in cluster  $C$ . Assuming that  $(h_i, k_i)$  be location of  $dev(i)$ , where  $i \leq i \leq num(c)$  then centroid of the cluster will be denoted as  $(X_{cen}, Y_{cen})$  as formulated in (1) and (2).

$$X_{cen} = \left\{ \sum_{i=1}^{num(c)} h_i \right\} / num(c) \tag{1}$$

$$Y_{cen} = \left\{ \sum_{i=1}^{num(c)} k_i \right\} / num(c) \tag{2}$$

Eligibility of a device  $dev(i)$  as a CH is governed by the function  $cls\_elg(i)$  as in (3), where  $(R - E_i)$  is residual energy of the  $i$ th device.

$$cls\_elg(i) = \frac{R - E_i}{\sqrt{\{(h_i - X_{cen})^2 + (k_i - Y_{cen})^2\}}} \tag{3}$$

The above mathematical expression is based on the fact that if a node has high residual energy and geographical closeness to the centroid of the cluster then its eligibility increases to become a CH.

### 3.3. Assigning Priority to Tasks

Each CH maintains a history of task propagation from its member IoT devices to it. The attributes of this information are as follows:

- i.  $tsk\_id$  — Unique identification number of the task.
- ii.  $tsk\_gen\_id$  — Unique identification number of the cluster member that has generated the task.
- iii.  $tm\_gen$  — Timestamp of generation of the task.
- iv.  $tm\_exec$  — Timestamp of completing execution of the task. If the task is not executed so far this field is set to  $-1$ . That is, for ongoing tasks,  $tm\_exec$  is set to  $-1$ .
- v.  $pr\_tsk$  — Priority assigned by the task.
- vi.  $len\_tsk$  — Length of the task.
- vii.  $starv\_tsk$  — Starvation status of the task. An IoT device is termed as starving provided it was inserted at least three times in the task queue and subsequently got retrieved from it to give place to higher priority tasks.

At least thirty tasks are required in the history of computing priority of subsequent tasks and participation of all IoT devices, and cluster members, within the history, is considered mandatory in this article to arrive at a close-to-accurate task priority computation priority of a new task  $\tau(j)$  submitted by the device  $dev(i)$  of cluster  $C$ , is formulated in (4).

$$pr(\tau(j), dev(i)) = \begin{cases} Min(comp\_priority(\tau(j), dev(i)), Max\_pr), & \text{if } \emptyset(j) \text{ is predicted not to starve} \\ Min(f'(comp\_priority(\tau(j), dev(i))), Max\_pr), & \text{otherwise} \end{cases} \tag{4}$$

where,

$$comp\_priority(\tau(j), dev(i)) = multi\_regress(tm\_gen(j), len\_tsk(j), \left( \sum_{\forall \tau(k) \in \Gamma_j(i)} (tm\_exec(k) - tm\_gen(k)) / |\Gamma_j(i)|, |S_j(i)| \right))$$

$$f'(prty) = prty \times \frac{Max\_pr - Min\_pr(t')}{prty - Min\_pr(t')} \tag{5}$$

$Max\_pr$  is the maximum possible priority in the system. Through the decision tree, it is predicted whether the new task  $\tau(j)$  generated by the IoT device  $dev(i)$  is expected to starve or not. It is predicted not to starve, then it is assigned the priority which is the minimum of its normally computed priority using  $comp\_priority$  function and  $Max\_pr$ .  $comp\_priority$  applies multiple regression function  $mult\_regress$  where priority of the current task  $\tau(j)$  is computed based upon timestamp of its generation ( $tm\_gen(j)$ ), length of the task ( $len\_tsk(j)$ ), waiting time of predecessors of  $\tau(j)$  who served. By predecessors of  $\tau(j)$  we mean a set of all tasks  $\tau(k)$  generated by the same device  $dev(i)$ , s.t.,  $tm\_exec(k) \neq -1$ , i.e., the task  $\tau(k)$  has already been completed before the current time  $t$ . For a task  $\tau(k) \in \Gamma(i)$ ,  $(tm\_exec(k) - tm\_gen(k))$  is the sum of its waiting time in the task queue and processing time by the UAV. Therefore,

$$\left\{ \sum_{\forall \tau(k) \in \Gamma_j(i)} (tm\_exec(k) - tm\_gen(k)) \right\} / |\Gamma_j(i)|$$

is the average time required for completion of all predecessors of  $\tau(i)$ . If this computed priority of  $\text{dev}(i)$  is higher than  $\text{Max\_pr}$ , then it is set to  $\text{Max\_pr}$  otherwise the computed priority requires. On the other hand, if the decision tree predicts starvation of the task, then its priority is ignored a bit from computed priority within the state of  $\text{Max\_pr}$  which is the maximum possible priority in the system and,  $\text{Min\_pr}(f')$  which is the minimum priority of a task present in the task queue at current time  $t$ . Please note that irrespective of the currently computed priority  $\text{prty}$ ,  $\text{Max\_pr} - \text{Min\_pr}(f')$  is always greater than  $\text{prty}$  or equal to  $(\text{prty} - \text{Min\_pr}(f'))$  and therefore multiplying  $\text{prty}$  with the fraction  $(\text{Max\_pr} - \text{Min\_pr}(f')) / (\text{prty} - \text{Min\_pr}(f'))$  using  $f'$  function will increase the priority value. However, this increased value cannot cross  $\text{Max\_pr}$  because we are taking the minimum of the output priority generated by  $f'$  and  $\text{Max\_pr}$ .

### Implementation Details of the Decision Tree

A decision tree is a powerful supervised learning algorithm that we have used in our research article for predicting the starvation status of the task. Available data is sliced as 80% and 20%, where 80% is training data and 20% is testing data. "Information Gain (IG)" in (6) is applied to select that particular attribute that will be placed at the top or as a root node, computation of information gain is based on the concept of entropy or "measure of uncertainty" as shown in (7). Certain accuracy, precision, and recall are used to calculate the efficiency of the trained classifier as shown in (8)–(10). The parameters of the decision tree appear in Table 2.

$$IG = Entropy(\text{parent}) - Entropy(\text{children}) \tag{6}$$

Here IG specifies information gain, and as the name specifies, Entropy(parent) and Entropy(children) denote the entropy of the parent node and the average of entropies of all its children. The formula for entropy appears in (7).

$$Entropy = - \sum_{i=1}^N p_i \log_2 p_i \tag{7}$$

where  $p_i$  is the probability of the random selection of an element in class  $i$ .

**Table 2.** Parameters of Decision Tree and Its Values.

Name of the Parameter	Value
max_depth (maximum number of levels permitted)	8 (Selection based on all attributes is permitted)
min_samples_leap (minimum number if samples stable in a leap node)	1 (least number of sample is 1)
min_sample_split (minimum number of samples a node must have before splitting)	2 (this is the minimum requirement in line with least number of samples per leaf is 1)

Like any other prediction, for the prediction of task starvation also, there are provisions for true positive, true negative, false positive, and false negative. Based on this accuracy, precision and recall are calculated below using (8)–(10).

$$accuracy = \frac{truepositive + truenegative}{truepositive + truenegative + falsepositive + falsenegative} \tag{8}$$

$$precision = \frac{truepositive}{truepositive + falsepositive} \tag{9}$$

$$recall = \frac{truepositive}{truepositive + falsenegative} \tag{10}$$

The values of these metrics should be high enough (at least 80%) in order to increase the efficiency of the prediction.

### 3.4. Prediction Next Timestamp of Task Submission

From the history of task propagation, a CH can predict the timestamp of the next intended submission for a device dev(i). Suppose, a set of task submission sessions of dev(i) till current time t' is denoted as ST(i, t'). From this set, some sessions starved where as others did not starve. Let  $ST \sim starve(i, t')$  be the subset of ST(i, t'), which consists of only the nonstarving tasks. For each task  $\tau'' \in ST \sim starve(i)$  let  $tm\_gen(\tau'')$  be the time of generation of  $\tau''$ , we apply the ARMA model to predict the next suitable value of generation of the next task in  $tm\_gen\_next(i, t')$  of device i at current time t', as shown in (11).

$$tm\_gen\_next(i, t') = ARMA\_prednext\left(\forall tm\_gen(\tau'') \quad \tau'' \in ST \sim starve(i, t')\right) \tag{11}$$

ARMA\_prednext is the function that applies the ARMA model for predicting the next task that is expected not to starve.

### 3.5. Traversal of UAV- on the Set of Clusterhead (CH)

Let the set of CHs under one UAV, is denoted as  $cls(1), cls(2), \dots, cls(\alpha)$ . For any CH  $cls(\alpha)$  s.t.  $1 \leq u \leq \alpha$ . Let, (lat(u), long(u)) denote the geographical position of the CHs in terms of latitude and longitude. Initially, the UAV starts traversal with  $cls(v)$  provided, and the following two conditions are satisfied.

$$1 \leq v \leq \alpha, \text{ and}$$

$$\sqrt{(x_{UAV} - lat(v))^2 + (y_{UAV} - long(v))^2} \leq \sqrt{(x_{UAV} - lat(u))^2 + (y_{UAV} - long(u))^2}$$

for all u s.t.  $1 \leq u \leq \alpha$

The conditions are self-explanatory. Here,  $cls(u)$  specifies the CH closest to the UAV. As soon as the UAV arrives on the first CH, it executes all of its tasks in the task queue and insets the information in a table TASK-EXEC, where the attributes are as bellow:

- i. *CLS\_ID* — This specifies the unique identification number of the CH whose task in the task queue has recently been executed by the UAV.
- ii. *LATTITUDE*—Geographical latitude of the current CH is located by the UAV.
- iii. *LONGITUDE*— Geographical longitude of the current CH is located by the UAV.
- iv. *TSK\_DISTINCT\_DEV*— Number of distinct IoT devices whose tasks have been executed by the UAV.
- v. *TSK\_NUM*— This specifies the number of tasks that were submitted.
- vi. *TSK\_NUM*— This specifies the number of tasks successfully completed.
- vii. *ENRG\_CONSUMED*— The amount of energy spent by the UAV in the execution of these tasks, is given by this attribute.

For the second CH, the entire *TASK\_EXEC* of the first CH is carried by the UAV to the second cluster and is ultimately inserted in *TASK\_EXEC* table of the second one. Then all tasks in the task queue of the second CH will be executed and the corresponding entry is also inserted in *TASK\_EXEC* of the second CH. Please note that the task of the first CH contains only task execution information of its own whereas the same table of the second CH contains task execution information of both the first and second CH. Like this, *TASK\_EXEC* of the third CH will contain task execution information of all three of the first,

second, and third CHs. In general, we can say that *TASK\_EXEC* table of *u*-th CH where  $1 \leq u \leq \alpha$  will contain information about executed tasks in all CHs 1 to (*u*-1). This will continue for *m* number of rounds, where *m* is the minimum number such that every CH has every other CHs task execution information and at least thirty observations exist in *TASK\_EXEC* of every CH. At this point, the UAV changes its strategy of traversing always the nearest CH unvisited in the current round. Rather it asks the current CH to know the identification number of the next CH to visit and the current CH applies a GA to answer that question. This GA applied for task execution (GATE) is illustrated in Section 3.5.

Details of Genetic Algorithm for Task Execution (GATE)

Flowchart of GA for Task Execution (GATE) appears in Figure 2. In this example, the identification numbers of CHs are in binary to form individual solutions which are called chromosomes. So, chromosomes are binary numbers whose values between 1 and  $\alpha$ , therefore the number of bits present in each chromosome is  $\log_2 \alpha$ . The fitness function of chromosome *CS* depends on the factors like the average number of tasks that could be successfully computed, the average number of IoT devices that could be served, the energy consumed in each task execution session, and the physical distance of *CS* from the immediate previously visited CH. The mathematical expression of fitness function  $fn(CS)$  of chromosome *CS*, appears in (12).

$$fn(CS) = \frac{\frac{f1(CS)}{|task\_ses(CS)|} - \frac{f2(CS)}{|task\_ses(CS)|^2}}{f3(CS)\sqrt{f4(CS)}} \quad (12)$$

where,

$$\begin{aligned} f1(CS) &= \sum_{\rho \in task\_ses(CS)} num\_of\_tasks(\rho) \\ f2(CS) &= \sum_{\rho \in task\_ses(CS)} num\_of\_dev(\rho) \\ f3(CS) &= \sum_{\rho \in task\_ses(CS)} energy\_consumed(\rho) \\ f4(CS) &= \{(lat(CS) - lat(curCS))^2 + (long(CS) - long(curCS))^2\} \end{aligned}$$

Here  $num\_of\_tasks(\rho)$  and  $num\_of\_dev(\rho)$  specify the number of tasks successfully completed in  $\rho$ -th task execution session and the number of different IoT devices served in the same  $\rho$ -th session.  $task\_ses(CS)$  is the set of all task execution sessions in the *TASK\_EXEC* table of *CS*. The average number of tasks served is  $\frac{\sum_{\rho \in task\_ses(CS)} num\_of\_tasks(\rho)}{|task\_ses(CS)|}$ .

Similarly average number of IoT device is  $\frac{\sum_{\rho \in task\_ses(CS)} num\_of\_dev(\rho)}{|task\_ses(CS)|}$ .

Average energy consumed in each session is  $\frac{\sum_{\rho \in task\_ses(CS)} energy\_consumed(\rho)}{|task\_ses(CS)|}$ .

*curCS* is the current CH and *CS* is a probable next solution i.e.,  $CS \in [(\cup_{i=1}^{\alpha} CLS(i)) - \{curCS\}]$ .

Therefore Cartesian distance from current CH is given by

$$\sqrt{(lat(CS) - lat(curCS))^2 + (long(CS) - long(curCS))^2}$$

Selection of the next generation suitable chromosome for crossover and mutation is performed using Roulette Wheel Selection. Based on the efficiency of individual chromosomes as a percentage of contribution to the fitness of the overall population, chromosomes are propagated to the next generation after mating.

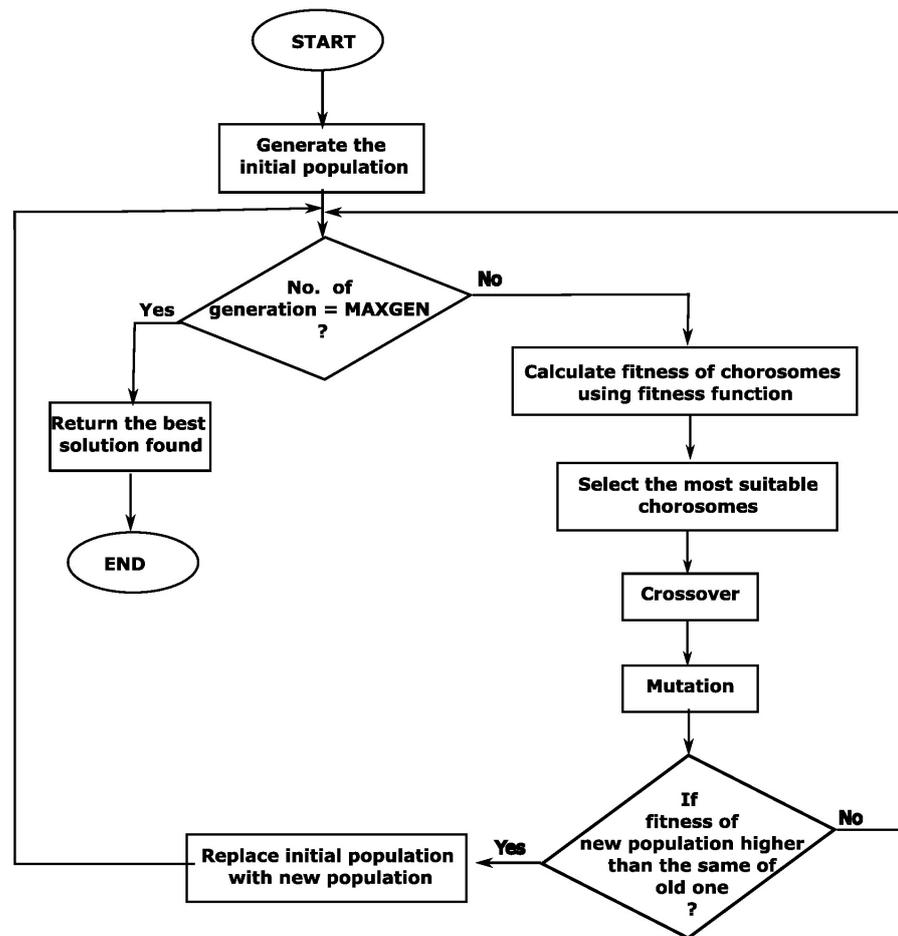


Figure 2. Flowchart of the GATE.

#### 4. Simulation Experiments, Results, and Discussion

The simulation environment is discussed in Section 4.1, whereas results are appeared in Section 4.2.

##### 4.1. Simulation Environment

Table 3 summarize values of all simulation parameters. The number of UAVs is 1, 3, 5, 7 and 9 in five different simulation runs and here we have not considered the possibility of the intersection of multiple-UAV except in the first simulation run where the number of UAVs is 1. As our system transmits from single UAV to multiple-UAV systems and the number of UAVs also increases in various simulation runs, they are able to serve more and more IoT devices improving the scalability of the entire system. That is why the number of IoT devices has also increased in different simulation runs. Our proposed technique EETO-GA is compared with other state-of-art trajectory optimization protocols, such as “comprehensive energy consumption (CEC)”, “propulsive energy minimization (PEM)” and “Age of information based tragedy planning (AoI-IP)”. Metrics of performance are UAV-Energy-Consumed(UAVE), IoT Device-Energy-Consumed(IoTDEC), Average Delay (AD), and Percentage of Successfully Completed Tasks (PSTK).

**Table 3.** Names and Values of Various Simulation Parameters.

Name of the Parameter	Value
Coverage Area	300 × 300 m <sup>2</sup>
Number of IoT devices	100, 300, 500, 700, 900
Number of UAVs	1, 3, 5, 7, 9
Min no. of devices in a cluster	3
Max no. of devices in a cluster	20
Hovering altitude of UAV	6m
No. of CPU cycles/s in an UAV	0.5G cycles/s
Channel gain	−20 dB
UAV speed	20 m/s
Noise power	−50 dBm
Transmission bandwidth	150 khz

4.2. Formulation of Metrics and Results

We have formulated four metrics for the measurement of results that are briefly mentioned below:

UAV-Energy Consumption(UAVE): UAVE is the summation of the energy consumption of all UAVs throughout the simulation period. Consumed energy is the difference between the initial energy of a UAV and its residual energy after the simulation period. So,

$$UAVE = \sum_{Cur\_UAV \in UAV\_Set} (init\_energy(cur\_UAV) - final\_energy(cur\_UAV)) \quad (13)$$

here Cur\_UAV denotes the current UAV that is being considered. UAV\_Set is the set of all UAVs applied in the current simulation scenario. init\_energy(cur\_UAV) and final\_energy(cur\_UAV) specify initial and final powers of the UAV named cur\_UAV, after the end of the simulation process. Using the formula in (13), UEC has been measured in other state-of-the-art related to EETO-GA. However, in EETO-GA, we maintain a table TASK\_EXEC in each UAV where the last column is mentioned as ENRG\_CONSUMED which specifies the amount of energy spent by the current UAV in the execution of these tasks (it is also specified in Section 3.5). So, big summing this column, we get the total energy consumption of the current UAV throughout the simulation period summing over all UAVs, we get UEC as shown in (14).

$$UAVE = \sum_{Cur\_UAV \in UAV\_Set} \sum (ENRG\_CONSUMED in TASK\_EXEC (cur\_UAV)) \quad (14)$$

IoT Device Energy Consumed (IoTDEC):

In state-of-the-art related to EETO-GA, energy consumption in IoT devices was not separately considered, So, for those algorithms, we have measured energy consumption as the difference between initial energy and residual energy, as shown in (15).

$$IoTDevice = \sum_{cur\_dev \in IoTDevice\_set} (init\_energy(cur\_dev) - final\_energy(cur\_dev)) \quad (15)$$

Here, init\_energy(cur\_dev and final energy cur\_dev denote the initial and final power of the current IoT device cur\_dev. IoTDevice\_set is the total number of IoT devices in the network. On the other hand, in EETO-GA, each IoT device maintains energy consumption information C-Ei where C-Ei is the energy consumed so far by the ith device which was discussed in Section 3.2, so, it can be formulated as in (16).

$$IoTDevice = \sum_{cur\_dev \in IoTDevice\_set} (C - E_{cur\_dev}) \quad (16)$$

Average Delay in Execution of the Task (ADET):

Average delay, for all the algorithms calculated as in (17). This is nothing but the difference between timestamps of initiation of a task and its execution, averaged over all the tasks for all CHs.

$$ADET = \sum_{cur\_cs \in \cup_{i=1}^{1000} CLS(i)} \sum_{\rho \in task\_ses(cur\_cs)} (tm\_exec(\rho) - tm\_gen(\rho)) \quad (17)$$

Here, *cur\_CS* is current CH,  $\rho$  is a task submitted to it  $tm\_exec(\rho)$  and  $tm\_gen(\rho)$  are execution completion time and task generation time for the task  $\rho$ .

Percentage of tasks that could be completed (PTSK): As per Section 3.5, it can be formulated in (18).

We experimented with proposed EETO-GA, and other state-of-the-art, namely CEC, PEM, and AoI-IP in the summation setup mentioned in Table 3. Comparative results are presented in graph formats in Figures 3–6. In this simulation scenario, if we considered highest results of CEC, PEM, and AoI-IP in each setup as standard system performance indicator, then proposed EETO-GA improved average UAVE: 43%, IoTDEC: 56%, PTSK: 7.5%, and ADET: 38%. The performance improvements are explained in the next Subsection.

$$PSTK = \frac{\sum_{Cur\_UAV \in UAV\_Set} (cur\_UAV)}{\sum_{Cur\_UAV \in UAV\_Set} TSK\_SUB(CUR\_UAV)} \times 100\% \quad (18)$$

#### 4.3. Explanation

In CEC and PEM, only the propulsion energy consumption in UAVs is considered, whereas in EETO-GA, propulsion energy in UAVs as well as energy spent in task propagation (submission and series of subsequent resubmission in case of starvation) is considered. In the fitness function of the GA in our proposed scheme, we have included a number of tasks that could be successfully completed, and an increased number of tasks means, the opportunity of arrival of UAV has been utilized close to the fullest. Otherwise, another arrival from UAE would have been required to complete those pending tasks and those tasks would have to be resubmitted from member IoT devices to the respective CHs requiring more energy consumption in those devices. This would not only lead to an increased amount of energy consumption in UAVs, as well as IoT devices. Therefore, energy consumption in both UAV and IoT devices in our proposed scheme EETO-GA, is far lesser than than state of the art, namely, CEC, PEM, and AoI-TP, as shown in Figures 3 and 4.

High energy consumption means a lesser lifetime for devices and exhausted IoT devices would not be able to submit all the jobs they intended to submit to the CH for possible execution by the UAV. Hence lesser the number of tasks that will be submitted in CEC, PEM, and AoI-TP compare to proposed EETO-GA. That is, more tasks will be executed by the EETO-GA than its competitors as shown in Figure 5. Another reason for the improvement of our proposed scheme EETO-GA is that it considers the criteria of starvation of tasks. Based on timestamps of generation of nonstarved tasks, an ARMA-model-based prediction of timestamps of generating a new task is prescribed by CH to its members in EETO-GA, so that the probability of starving is reduced. Reduction in starvation also contributes to energy saving in IoT devices otherwise tasks would have to be resubmitted again and again, especially when the number of IoT devices is as high as 700 or 900. For this reason, as the number of IoT devices increase, the improvement of EETO-GA in terms of energy consumption, over its competitors, become bigger and bigger as shown in Figures 3 and 4.

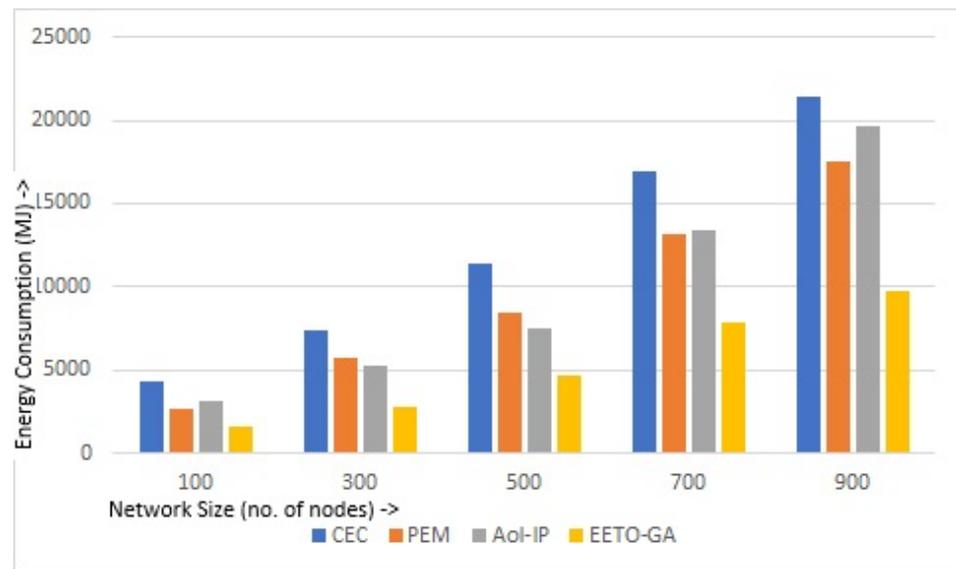


Figure 3. UAV -Energy-Consumed (UAV-E) in Proposed EETO-GA and Other State-of-the-art.

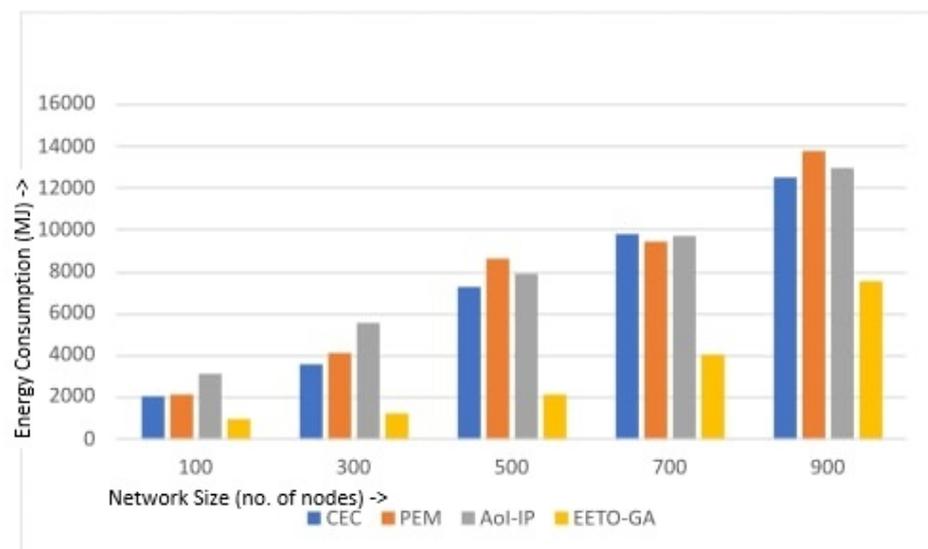
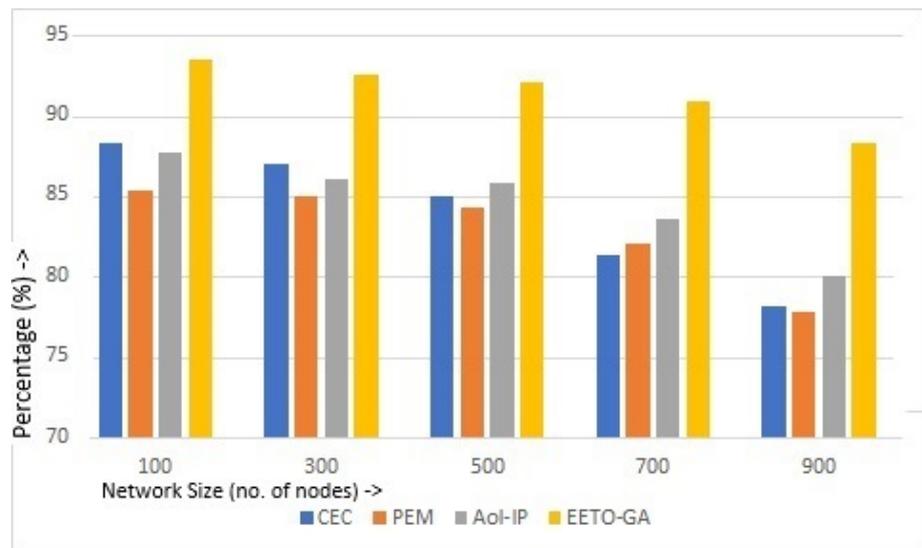


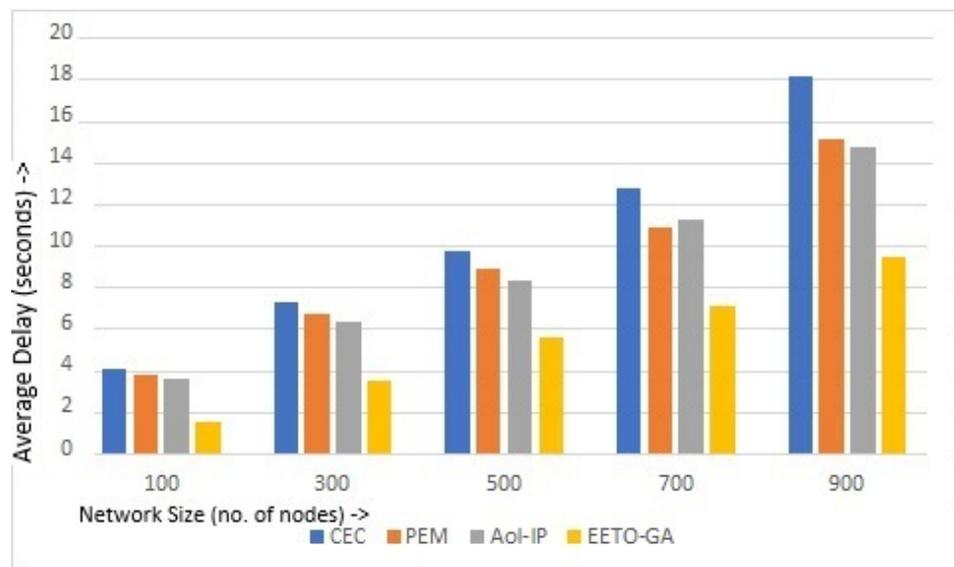
Figure 4. IoT Device Energy Consumed (IoTDEC) in EETO-GA and Other State-of-the-art.

The average delay in the execution of a task is also an important performance parameter. As the number of starving tasks and their serving times has been consciously reduced in EETO-GA, by increasing their priorities (maximum up to  $Max_{pr}$ ), so the average time required for the execution of each of these tasks, is reduced as shown in Figure 6. Tasks are submitted at suitable times prescribed by autoregressive moving average on the ARMA model, for which chances of their starvation get reduced, decreasing their average delay. Although the proposed technique is not real-time, but it tried to minimize the average delay as much as possible, however, for this particular simulation study maximum acceptable delay was set to 20 s.

Combining together all the Figures 3–6, we can emphatically say that EETO-GA outperforms all its above-mentioned competitors as far as the matrices like energy and delay concerned.



**Figure 5.** Percentage of successfully completed tasks (PTSK) in proposed EETO-GA and other state-of-the-art.



**Figure 6.** Average delay in execution of task (ADET) in proposed EETO-GA and other state-of-the-art.

### 5. Conclusions and Future Scopes

This article concentrates on two things energy efficient trajectory planning and proper scheduling of tasks generated by IoT devices so that amount of starvation can be reduced and the average time required for the execution of a task can be decreased. This makes the system efficient both from an energy and delay perspective although energy has been specifically highlighted because propulsion energy in UAVs as well as energy consumed by IoT devices have both been considered in this scheme. In order to predict the possibility of starvation have implemented a decision tree and the priority of starving task are increased so that delay is reduced in our system. However, the limitation of our proposed work is that it has been implemented in the simulator and therefore, in the future, should be implemented in real-life physical situations so that performance metrics can be computed and evaluated in a more reliable manner in alignment with the standard system performance indicator.

In this research work, we have implemented EETO-GA and the state of the art in a simulator. In the future, we would like to implement all these using real-life infrastructure. In that case, practical issues like rate, time of use, etc. will be considered. Obtaining

good results during the current simulation work encourages us to proceed further toward procuring a real-life experimental setup, which is justified by performance enhancement produced by EETO-GA in the simulated environment.

**Author Contributions:** Conceptualization, A.B. and A.S.; Methodology, M.M.H.R.; Validation, M.A.-N.; Investigation, M.M.H.R., A.B. and A.S.; Data curation, A.B. and A.S.; Writing—original draft, A.B. and A.S.; Writing—review & editing, M.M.H.R. and M.A.-N.; Project administration, M.A.-N.; Funding acquisition, M.M.H.R. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported through the Deputyship for Research & Innovation, Ministry of Education in Saudi Arabia through the project number INST026. The authors extend their appreciation to the Deputyship for Research and Innovation, Ministry of Education in Saudi Arabia, for funding this research work (Project number INST026).

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

**Acknowledgments:** The authors of the papers acknowledge the editor and anonymous reviewers for giving valuable comments and suggestions on the initial submission. Of such comments and suggestions, the paper got this present shape.

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

## References

1. Mohsan, S.A.H.; Khan, M.A.; Noor, F.; Ullah, I.; Alsharif, M.H. Towards the unmanned aerial vehicles (UAVs): A comprehensive review. *Drones* **2022**, *6*, 147. [\[CrossRef\]](#)
2. Liu, Y.; Dai, H.N.; Wang, Q.; Shukla, M.K.; Imran, M. Unmanned aerial vehicle for internet of everything: Opportunities and challenges. *Comput. Commun.* **2020**, *155*, 66–83. [\[CrossRef\]](#)
3. Nazib, R.A.; Moh, S. Routing protocols for unmanned aerial vehicle-aided vehicular ad hoc networks: A survey. *IEEE Access* **2020**, *8*, 77535–77560. [\[CrossRef\]](#)
4. Jia, Z.; Qin, X.; Wang, Z.; Liu, B. Age-based path planning and data acquisition in UAV-assisted IoT networks. In *ICC Workshops, Proceedings of the 2019 IEEE International Conference on Communications Workshops, Shanghai, China, 20–24 May 2019*; IEEE: New York, NY, USA, 2019; pp. 1–6.
5. Maddikunta, P.K.R.; Hakak, S.; Alazab, M.; Bhattacharya, S.; Gadekallu, T.R.; Khan, W.Z.; Pham, Q.V. Unmanned aerial vehicles in smart agriculture: Applications, requirements, and challenges. *IEEE Sensors J.* **2021**, *21*, 17608–17619. [\[CrossRef\]](#)
6. Hildmann, H.; Kovacs, E. Using unmanned aerial vehicles (UAVs) as mobile sensing platforms (MSPs) for disaster response, civil security and public safety. *Drones* **2019**, *3*, 59. [\[CrossRef\]](#)
7. Kumar, N.; Puthal, D.; Theocharides, T.; Mohanty, S.P. Unmanned aerial vehicles in consumer applications: New applications in current and future smart environments. *IEEE Consum. Electron. Mag.* **2019**, *8*, 66–67. [\[CrossRef\]](#)
8. Chaturvedi, S.K.; Sekhar, R.; Banerjee, S.; Kamal, H. Comparative review study of military and civilian unmanned aerial vehicles (UAVs). *INCAS Bull.* **2019**, *11*, 183–198. [\[CrossRef\]](#)
9. Li, X.; Savkin, A.V. Networked unmanned aerial vehicles for surveillance and monitoring: A survey. *Future Internet* **2021**, *13*, 174. [\[CrossRef\]](#)
10. Banerjee, A.; Sufian, A.; Paul, K.K.; Gupta, S.K. EDTP: Energy and Delay Optimized Trajectory Planning for UAV-IoT Environment. *Comput. Netw.* **2022**, *202*, 108623. [\[CrossRef\]](#)
11. Li, B.; Na, Z.; Lin, B. UAV Trajectory Planning from a Comprehensive Energy Efficiency Perspective in Harsh Environments. *IEEE Netw.* **2022**, *36*, 62–68. [\[CrossRef\]](#)
12. Aggarwal, S.; Kumar, N. Path planning techniques for unmanned aerial vehicles: A review, solutions, and challenges. *Comput. Commun.* **2020**, *149*, 270–299. [\[CrossRef\]](#)
13. Zhang, H.; Xin, B.; Dou, L.H.; Chen, J.; Hirota, K. A review of cooperative path planning of an unmanned aerial vehicle group. *Front. Inf. Technol. Electron. Eng.* **2020**, *21*, 1671–1694. [\[CrossRef\]](#)
14. Cabreira, T.M.; Brisolara, L.B.; Paulo R, F.J. Survey on coverage path planning with unmanned aerial vehicles. *Drones* **2019**, *3*, 4. [\[CrossRef\]](#)
15. Goh, C.Y.; Leow, C.Y.; Nordin, R. Energy Efficiency of Unmanned Aerial Vehicle with Reconfigurable Intelligent Surfaces: A Comparative Study. *Drones* **2023**, *7*, 98. [\[CrossRef\]](#)
16. Li, X.; Li, J.; Liu, D. Energy-Efficient UAV Trajectory Design with Information Freshness Constraint via Deep Reinforcement Learning. *Mob. Inf. Syst.* **2021**, *2021*, 1430512. [\[CrossRef\]](#)

17. Debnath, S.K.; Omar, R.; Latip, N.B.A. A review on energy efficient path planning algorithms for unmanned air vehicles. In Proceedings of the Computational Science and Technology: 5th ICCST 2018, Kota Kinabalu, Malaysia, 29–30 August 2018; pp. 523–532.
18. Zeng, Y.; Xu, J.; Zhang, R. Energy minimization for wireless communication with rotary-wing UAV. *IEEE Trans. Wirel. Commun.* **2019**, *18*, 2329–2345. [[CrossRef](#)]
19. Dong, F.; Li, L.; Lu, Z.; Pan, Q.; Zheng, W. Energy-efficiency for fixed-wing UAV-enabled data collection and forwarding. In *ICC Workshops, Proceedings of the 2019 IEEE International Conference on Communications Workshops, Shanghai, China, 20–24 May 2019*; IEEE: New York, NY, USA, 2019; pp. 1–6.
20. Johnson, D.B. A note on Dijkstra’s shortest path algorithm. *JACM* **1973**, *20*, 385–388. [[CrossRef](#)]
21. Forrest, S. Genetic algorithms. *ACM Comput. Surv. CSUR* **1996**, *28*, 77–80. [[CrossRef](#)]
22. Mirjalili, S. Genetic algorithm. In *Evolutionary Algorithms and Neural Networks*; Springer: Berlin/Heidelberg, Germany, 2019; pp. 43–55.
23. Ljung, G.M.; Box, G.E. The likelihood function of stationary autoregressive-moving average models. *Biometrika* **1979**, *66*, 265–270. [[CrossRef](#)]
24. Quinlan, J.R. Generating production rules from decision trees. In Proceedings of the IJCAI, Milan, Italy, 23–28 August 1987; Volume 87; pp. 304–307.
25. Ueyama, J.; Freitas, H.; Faiçal, B.S.; Geraldo Filho, P.; Fini, P.; Pessin, G.; Gomes, P.H.; Villas, L.A. Exploiting the use of unmanned aerial vehicles to provide resilience in wireless sensor networks. *IEEE Commun. Mag.* **2014**, *52*, 81–87. [[CrossRef](#)]
26. Arafat, M.Y.; Moh, S. A survey on cluster-based routing protocols for unmanned aerial vehicle networks. *IEEE Access* **2018**, *7*, 498–516. [[CrossRef](#)]
27. Huang, S.; Teo, R.S.H.; Tan, K.K. Collision avoidance of multi unmanned aerial vehicles: A review. *Annu. Rev. Control.* **2019**, *48*, 147–164. [[CrossRef](#)]
28. Yasin, J.N.; Mohamed, S.A.; Haghbayan, M.H.; Heikkonen, J.; Tenhunen, H.; Plosila, J. Unmanned aerial vehicles (uavs): Collision avoidance systems and approaches. *IEEE Access* **2020**, *8*, 105139–105155. [[CrossRef](#)]
29. Wu, J.; Zou, L.; Zhao, L.; Al-Dubai, A.; Mackenzie, L.; Min, G. A multi-UAV clustering strategy for reducing insecure communication range. *Comput. Netw.* **2019**, *158*, 132–142. [[CrossRef](#)]
30. Sun, Y.; Uysal-Biyikoglu, E.; Yates, R.D.; Koksal, C.E.; Shroff, N.B. Update or Wait: How to Keep Your Data Fresh. *IEEE Trans. Inf. Theory* **2017**, *63*, 7492–7508. [[CrossRef](#)]
31. Talak, R.; Karaman, S.; Modiano, E. Optimizing information freshness in wireless networks under general interference constraints. In Proceedings of the Eighteenth ACM International Symposium on Mobile Ad Hoc Networking and Computing, Los Angeles, CA, USA, 26–29 June 2018; pp. 61–70.
32. Zhang, L.; Celik, A.; Dang, S.; Shihada, B. Energy-efficient trajectory optimization for UAV-assisted IoT networks. *IEEE Trans. Mob. Comput.* **2021**, *21*, 4323–4337. [[CrossRef](#)]
33. Qadir, Z.; Zafar, M.H.; Moosavi, S.K.R.; Le, K.N.; Mahmud, M.P. Autonomous UAV Path-Planning Optimization Using Metaheuristic Approach for Predisaster Assessment. *IEEE Internet Things J.* **2021**, *9*, 12505–12514. [[CrossRef](#)]
34. Cui, J.; Ding, Z.; Deng, Y.; Nallanathan, A.; Hanzo, L. Adaptive UAV-trajectory optimization under quality of service constraints: A model-free solution. *IEEE Access* **2020**, *8*, 112253–112265. [[CrossRef](#)]
35. Li, Y.; Zhang, H.; Long, K.; Jiang, C.; Guizani, M. Joint resource allocation and trajectory optimization with QoS in UAV-based NOMA wireless networks. *IEEE Trans. Wirel. Commun.* **2021**, *20*, 6343–6355. [[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.