


## Research article

# Time-efficient approximate trajectory planning for AoI-centered multi-UAV IoT networks

Amirahmad Chapnevis, Eyuphan Bulut \*

Department of Computer Science, Virginia Commonwealth University, 401 West Main St. Richmond, VA 23284, USA

## ARTICLE INFO

## Keywords:

UAV  
Trajectory planning  
IoT  
Age of information  
Efficiency

## ABSTRACT

The gathering of data produced by ground Internet of Things (IoT) devices can be facilitated with the assistance from Unmanned Aerial Vehicles (UAVs) especially in hard-to-reach areas. However, the limited battery of UAVs requires a careful planning of their trajectories. As the timely delivery of data can be critical in certain applications, Age of Information (AoI) should also be integrated during this planning. Most of the existing works that study AoI-centered UAV trajectory planning focus on the timing of the data gathering by the UAV, without considering the time UAV needs to deliver it to a specific point. This study broadens the perspective by incorporating multiple UAVs and Ground Base Stations (GBSs) throughout the region, to be used for the delivery of data collected by UAVs, defining the AoI. We also allow UAVs to visit IoT locations only after a data is generated, which can happen during the mission of UAVs. Our goal is to optimize the UAV trajectories considering multiple prioritized goals, namely, minimization of maximum AoI, then the minimization of sum of AoI for all collected data and finally the sum of UAV path lengths. Using Integer Linear Programming (ILP), we first find out the optimal solution. In order to avoid the long running times and provide a scalable yet time-efficient solution, we propose a heuristic based method. Extensive simulation results under various setups show that the heuristic approach provides results with reasonable margins to ILP results and is also scalable, making the proposed solution more practical.

## 1. Introduction

Unmanned Aerial Vehicles (UAVs) have revolutionized various fields such as wireless communications, disaster management, agriculture and healthcare, thanks to their agility and ability to reach inaccessible areas [1–4]. The recently developed standards (e.g., 3GPP Release 17 [5], IEEE 802.11ah [6] (Wi-Fi HaLow)) also support their usage in various different conditions (e.g., long durations covering larger areas.) despite their limited energy resources. This study focuses on a scenario where UAVs serve a crucial role in data collection from ground sensors or Internet of Things (IoT) networks, acting as a bridge to relay this information to the intended destinations. Due to the limited communication range of UAVs and IoT devices, UAVs need to navigate close to each of the IoT devices along their designated paths to collect the data of IoT devices before concluding their mission.

Since the limited battery life of UAVs restricts their flight duration, it is essential to meticulously plan their routes to ensure efficient data collection. This planning must also take into account the specific times when data becomes available at each IoT device, necessitating visits to these devices only after data generation. However, most of the existing studies (e.g., [7–9]) assume that the data is available before the UAVs commence their mission. Contrary to these studies, we consider a more practical scenario and allow data generation even during the UAV missions.

\* Correspondence to: 401 W. Main St. Richmond, VA 23059, USA

E-mail addresses: [chapnevisa@vcu.edu](mailto:chapnevisa@vcu.edu) (A. Chapnevis), [ebulut@vcu.edu](mailto:ebulut@vcu.edu) (E. Bulut).

<https://doi.org/10.1016/j.iot.2024.101461>

Received 28 August 2024; Received in revised form 30 October 2024; Accepted 3 December 2024

Available online 10 December 2024

2542-6605/© 2024 Elsevier B.V. All rights are reserved, including those for text and data mining, AI training, and similar technologies.

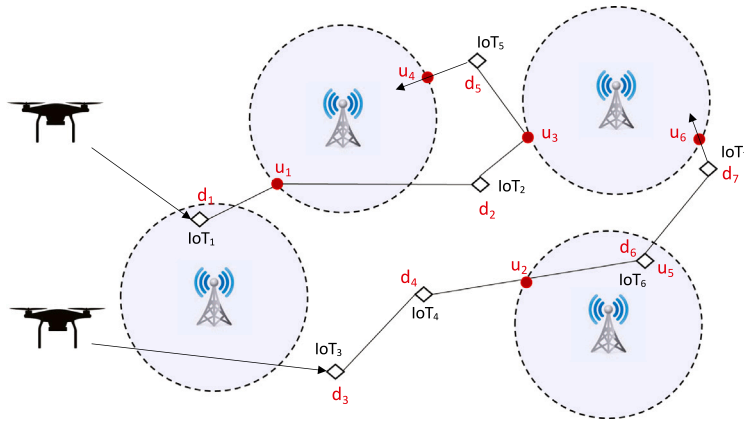


Fig. 1. An example scenario with two UAVs collecting data from seven IoT devices and delivering them to nearby base stations. AoI for a data is defined from the time IoT device generates the data to its delivery time to a GBS by a UAV.

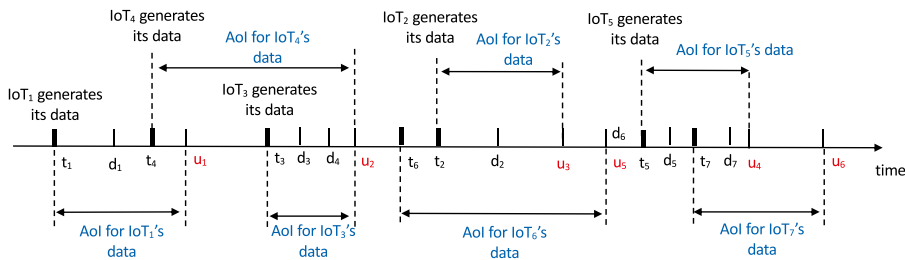


Fig. 2. Calculation of AoI for each data delivery in Fig. 1.

Once IoT devices generate data, the UAVs need to collect that data and then deliver it to their respective destinations, taking into consideration the requirements of the application at hand. In this work, we study a broader scenario by including multiple UAVs and multiple ground base stations (GBS) which are considered as the delivery points of the collected IoT data. As a result of this flexible delivery, we redefine the Age of Information (AoI) as the period starting from the data creation at IoT device to its upload to a GBS, ensuring the timeliness and relevance of the information relayed.

Recent studies have delved into UAV path planning within IoT networks with varying aims, such as minimizing energy consumption [10], maximizing coverage [11], reducing connection outage times [12,13], and maximizing data collection efficiency [14]. In order to take into account the data freshness upon delivery, which can be critical in various application contexts (e.g., healthcare, logistics), AoI is also considered as a metric during the path planning of UAVs. While previous research has largely focused on AoI in scenarios where data is delivered to a single endpoint, this study proposes a more practical and broader scenario where a UAV can upload the collected data to any nearby GBS, thus offering a more flexible and realistic approach to data delivery and finalizing mission requirements.

In Fig. 1, we illustrate an example scenario with two UAVs and seven ground IoT devices. UAVs are tasked to collect and deliver the data to a nearby GBS. The AoI for each collected data is computed in Fig. 2. Note that a UAV collects the data of an IoT  $i$  at time  $d_i$ , which is after its generation time  $t_i \leq d_i$ , and uploads the data to a GBS at a later time denoted by  $u$ . It is possible that multiple IoT data can be uploaded at the same time to reduce the UAV path length without increasing maximum AoI.

The contributions of this study<sup>1</sup> are as follows:

- We study the multiple UAV trajectory optimization problem with three prioritized goals (i.e., minimizing max AoI, reducing sum of AoI and the UAV path lengths) considering an AoI definition that covers the duration until UAV delivers the collected data to a nearby GBS, which is different from current UAV-assisted data collection studies.
- We model the problem and find the optimal UAV paths in a given scenario using Integer Linear Programming (ILP).
- We introduce a computationally efficient but approximate greedy heuristic algorithm for both single and multiple UAV scenarios.
- We perform extensive simulations considering various scenarios and evaluate the performance of proposed heuristic based approach, showing its benefits.

<sup>1</sup> In the preliminary version of this study [15], we explored only a single UAV scenario with a limited evaluation.

**Table 1**  
Comparison of the proposed work to the most recent and closest related work in the literature.

Study	Year	AoI	Multiple GBSs	Multiple UAVs	Path optimization
[28]	2022	IoT to users	✗	✓	✓
[29]	2023	IoT to GBS	✗	✗	✓
[30]	2024	IoT to UAV	✗	✓	✗
[31]	2023	IoT to UAV	✗	✗	✗
[32]	2024	IoT to UAV	✗	✗	✗
[33]	2024	IoT to UAV	✗	✓	✓
[34]	2022	IoT to UAV	✗	✗	✗
[35]	2024	IoT to UAV	✗	✗	✓
[36]	2024	IoT to GBS	✗	✓	✓
[37]	2023	IoT to GBS	✓	✗	✗
[38]	2022	IoT to UAV	✓	✗	✓
This study	2024	IoT to GBS	✓	✓	✓

The rest of the article is outlined as follows. We start with summarizing the existing AoI-centered UAV path planning studies in Section 2, and discuss how the proposed approach is different from them. In Section 3, we continue with our assumptions and describe the problem formally. Next, in Section 4, we first go through the ILP based solution, and elaborate on the proposed greedy heuristic based approach that aims to reduce the complexity while providing an approximate solution. We then provide an evaluation of the proposed solutions using simulations in different scenarios in Section 5. At the end, we conclude with summarizing our contributions and discussing some potential future directions in Section 6.

## 2. Related work

UAVs have been considered in various wireless communication applications [16] together with security [17], edge computing [18] and learning-based components [19]. Since in this work we study the UAV path planning and trajectory optimization problem, in this section, we cover only the related studies on this specific problem. UAV path planning and trajectory optimization problem have recently been studied extensively through various constraints and parameters and within different application scenarios including collection of data from ground IoT devices or sensors [2,20–22]. The objective in these trajectory optimization studies have varied from minimization of UAV mission time [12], energy consumption [23], and connection outage [24] to maximization of throughput [25] and coverage [26].

AoI has recently been considered within the context of UAV trajectory planning problems, in particular for obtaining more fresh data from the ground IoT devices [27]. UAVs are considered as data collecting agents from these ground sensors which are considered to be deployed in areas that are not covered by cellular infrastructure e.g., rural areas. AoI values the freshness and timely delivery of the data to their destinations thus many researchers have recently studied AoI-centered UAV trajectory planning problems. However, even within such studies, various application domains and slightly different objectives are targeted.

In [28], authors consider multiple UAVs that depart from a central location to collect the data of ground sensors and deliver them to the users in the area before returning to the start location. Thus, the AoI is described within a different context and cellular connectivity of UAVs (to any of the GBSs in the area) and delivery of data to GBSs is not considered. A single rotary-wing UAV is considered in [29], and the UAV path is optimized through a careful selection of its hovering points and using a transformer weighted A-star algorithm. While the AoI defined in this study matches with our description, there is only one base station and one UAV considered in the system model, making the problem different. In [30], energy-constraints of the UAVs have also been taken into account and using a graph theoretical and kernel K-means based method, the trajectory optimization of multiple UAVs have been studied while aiming both the reduction of the average AoI and UAV energy consumption. A similar work that consider energy consumption and AoI trade-off is presented in [31]. Besides graph-theory based approaches, some studies also consider clustering-based approaches to assign UAVs to data collection points [32,33]. Furthermore, there are some studies that consider wireless charging of the sensor nodes by UAVs [34,35], visits of UAVs to charging stations or charging of UAVs by unmanned ground vehicles (UGV) [36] while also considering AoI.

Despite several studies that consider AoI in UAV trajectory planning, optimization of AoI mostly considers the time period until the data is collected by the UAV, without much focus on the time needed by the UAV to upload it to the Internet or cellular infrastructure. This is because most of the time there is only one base station or data center considered where UAV just needs to visit to upload collected data. However, in more practical scenarios, especially in wide-area deployments, there can be multiple GBSs and UAVs can potentially deliver the collected data from IoTs to any of these GBSs. Based on our literature coverage, there is only one other study [37] that consider multiple GBSs in the environment and allow uploading of the data from a UAV to any of them during its mission. However, that study does not consider multiple UAVs and only targets optimization of AoI, without considering the path optimization jointly. Also, only an optimization solution with very long running times is presented there, while in this work, we target a cost-efficient and scalable solution through a heuristic based approach.

Learning based solutions have also been explored in UAV trajectory optimization during IoT data collection applications, especially when there are unknowns in the environment or in the data generation schedules of the IoT devices. To this end, different neural networks (e.g., Deep Q Network (DQN) [39], Long short-term memory (LSTM) [40], deep reinforcement learning [38,41])

**Table 2**  
The descriptions of notations used in this study.

Notation	Description
$\mathcal{U}, \mathcal{I}, \mathcal{G}$	The set of UAVs, ground sensors/IoT devices and ground base stations, respectively.
$V_i^D$	The time the IoT device $i$ is visited by a UAV and its data is captured.
$V_i^U$	The time the IoT device $i$ 's data is delivered to one of the GBSs by the UAV that collected that data.
$L_u^S, L_u^F$	The locations where UAV $u$ starts and finishes its travel, respectively.
$L_u, T_u$	The set of waypoints and visit times on the UAV $u$ 's path, respectively.
$L_u(t)$	UAV $u$ 's location at time $t \in T$ .
$l_i$	IoT device $i$ 's location, which is static.
$k_i$	GBS $i$ 's location, which is static.
$t_i$	The time that IoT device $i$ generates its data.
$e_i^u(t)$	A boolean that becomes 1 if the UAV $u$ is in the range of IoT $i$ at time $t \in T$ .
$d_i^u(t)$	A boolean that becomes 1 if the UAV $u$ captures the data of IoT device $i$ at time $t \in T$ .
$p_i^u(t)$	A boolean that becomes 1 if the UAV $u$ delivers IoT $i$ 's data to a GBS at time $t \in T$ .
$g_i^u(t)$	A boolean that becomes 1 if UAV $u$ is in the range of GBS $i$ at time $t \in T$ .
$G_u(t)$	A boolean that becomes 1 if UAV $u$ can communicate with at least one of the GBS at time $t \in T$ .
$U_D(t)$	The id of the UAV that captures the data of IoT device $i$ .
$U_P(t)$	The id of the UAV that delivers the data of IoT device $i$ to a GBS.
$R_I$	The communication range for an IoT to UAV link with minimum desired SNR.
$R_G$	The communication range for a UAV to GBS link with minimum desired SNR.
$T_{max}, V$	Total maximum time a UAV can fly and its possible maximum speed.
$T_u^F$	The first time slot that UAV $u$ reaches its final location and completes the mission.

have been utilized depending on the application and uncertainty setting considered. While such studies can potentially help in unknown scenarios, the training process can potentially take long, making such solutions not scalable as well. However, we plan to integrate learning-based components in our future efforts on this problem.

In Table 1, we provide a comparison of our study to other existing and close works and highlight the differences. In summary, this study differs from existing studies by considering both multiple UAVs and GBSs (for data delivery) and not only considers AoI in design but also aims to optimize UAV paths.

### 3. System model

#### 3.1. Assumptions

Let  $\mathcal{U}, \mathcal{G}, \mathcal{I}$  denote the set of UAVs, GBSs and IoT devices, respectively. We assume that each IoT device produces a data based on the requirements defined by the application it is used for. We denote the location of an IoT device  $i$  with  $l_i$  and the time it generates a data by  $t_i$ . It is assumed that a UAV  $u$  starts its mission at location  $L_u^S$  and ends it at  $L_u^F$ . The UAVs are tasked to gather data from ground IoT devices and deliver them to one of the nearby GBSs, before the allowed time, which is denoted as  $T_{max}$ . We assume a discrete model to define the time and use unit time slots. The maximum speed each UAV can reach is considered as  $V$  units per a time slot. The location of a UAV  $u$  at a time slot  $t$  is denoted by  $L_u(t) = (x_u(t), y_u(t), H)$  and each GBS  $i$ 's location is denoted by  $k_i = (x_i, y_i, H_G)$ . Without loss of generality, we consider that the UAVs perform their flights at the same altitude ( $H$ ), which is above the height of the GBSs, i.e.,  $H > H_G$ .

We assume that each UAV, GBS and IoT device has a single omni-directional antenna to be used in communication with others. Each (UAV, IoT) link or (UAV, GBS) link uses a separate band that is orthogonal to others to avoid the interference. A UAV needs to be in the vicinity of an IoT device to form a stable Line-of-Sight (LoS) connectivity and download its data. We assume that this is achieved when the distance between them is less than  $R_I$ . In order to deliver the collected data to a nearby GBS, a UAV should be in the GBS's range, defined by  $R_G$ . Here, note that, these range values can be derived from an analysis of the signals and considering the bandwidth required by the application [12,42]. For example, the communication range ( $R_G$ ) for a GBS–UAV link can be computed by  $\sqrt{\frac{\gamma_0}{S_{min}} - (H - H_G)^2}$ , where  $\gamma_0 = \frac{P\beta_0}{\sigma^2}$  denotes the baseline Signal-to-Noise Ratio (SNR) with  $P$  denoting the transmission power of a GBS,  $\sigma^2$  denoting the noise power at the receiver UAV, and  $\beta_0$  denoting the channel power gain at the reference distance. The minimum SNR,  $S_{min}$  is used to make sure the communication quality between a UAV and a GBS is at the level required by the specific application.

The symbols and their corresponding descriptions used in this work are provided in Table 2 for the convenience of readers.

#### 3.2. Problem statement

Given the sets  $\mathcal{U}, \mathcal{I}$ , and  $\mathcal{G}$ , together with the locations of IoT devices ( $l_i$ ), GBS centers ( $k_i$ ), starting and final locations of UAVs ( $L_u^S, L_u^F$ ) and the times that the IoT devices create their data ( $t_i$ ), our goal is to minimize the maximum AoI (i.e.,  $A_{max}$ ) of the data

collected from IoT devices by the UAVs. In addition to this primary goal, in order to obtain reasonable and energy-saving UAV paths in particular between the waypoints that do not affect the overall AoI, we also aim to minimize the sum of AoI (i.e.,  $A_{sum}$ ) as a secondary objective, and the total length of all UAV trajectories (i.e.,  $D_{sum}$ ) as the tertiary objective. These prioritized goals are integrated to our joint objective function using the scalarization method as follows:

$$\min ((A_{max})\lambda + A_{sum})\Theta + D_{sum} \tag{1}$$

s.t.

$$A_{max} = \max \{(V_i^U - t_i)\}, \forall i \in \mathcal{I} \tag{2}$$

$$A_{sum} = \sum_{\forall i \in \mathcal{I}} (V_i^U - t_i) \tag{3}$$

$$D_{sum} = \sum_{u \in \mathcal{U}'} (L_u^S \rightarrow L_u^F) \tag{4}$$

where  $V_i^U$  is the upload or delivery time of the IoT device  $i$ 's data to a GBS by the UAV that downloaded its data, and  $L_u^S \rightarrow L_u^F$  denotes the trajectory for the UAV  $u$ . We use scalars  $\lambda$  and  $\theta$ , to prioritize different goals, where  $\lambda$  and  $\theta$  are selected to be larger than what  $A_{sum}$  and  $D_{sum}$  can be, respectively.

#### 4. Proposed solutions

In this section, we first describe the ILP based solution, then discuss the heuristic based more computationally efficient and approximate solution.

##### 4.1. ILP solution

The goal of the ILP based approach is to obtain the optimal UAV paths under the aforementioned goals. Each UAV path is defined between a starting point and a final location, which can be the same as the starting point. We then try to identify the optimal UAV paths using waypoints on the route of each UAV  $u$ . Let  $L_u = \{L_u^0, L_u^1, L_u^2, \dots, L_u^{2|I|}, L_u^{2|I|+1}\}$  be the set of these waypoints on the optimal path of UAV  $u$ , where  $L_u^0 = L_u^S$  and  $L_u^{2|I|+1} = L_u^F$ . Each of these waypoints are critical locations where the UAV either captures the data from an IoT device or uploads the collected data. We also define  $T_u = \{T_u^0 = 0, T_u^1, T_u^2, \dots, T_u^{2|I|}, T_u^{2|I|+1} = T_u^F\}$  as the visit times of UAV  $u$  to the locations in  $L_u$ , i.e.,  $L_u(T_u^i) = L_u^i$ . Here,  $T_u^F$  is defined as UAV  $u$ 's mission time or the time it reaches its destination.

It is essential to ensure that each UAV commences and concludes its mission at specified locations. Furthermore, in order to take into account the limited UAV batteries during their mission, we consider a constraint on the maximum flight duration allowed. Thus, we have

$$L_u(0) = L_u^S \tag{5}$$

$$L_u(T_u^F) = L_u^F \text{ \& } T_u^F \leq T_{max}, \forall u \in \mathcal{U}. \tag{6}$$

For each consecutive time moment during the mission of UAVs, we ensure that the distance traveled by the UAVs is equal to or less than their maximum speed capability by:

$$\text{dist}_{L_u(i)}^{L_u(i+1)} \leq (T_u(i+1) - T_u(i)) \times V, \tag{7}$$

$$\forall i \in [0, 2|I|], \forall u \in \mathcal{U},$$

where  $\text{dist}_{l_1}^{l_2}$  denotes the distance between two locations  $l_1$  and  $l_2$ .

Each IoT device generates data at a specific time.<sup>2</sup> A UAV needs to visit an IoT device after its data generation time (i.e.,  $t_i$ ) to capture its data. Furthermore, the delivery of this data (i.e.,  $V_i^U$ ) to a GBS must occur only after it has been captured (i.e.,  $V_i^D$ ) from the IoT device. These are ensured by

$$V_i^D \geq t_i, \forall i \in \mathcal{I} \tag{8}$$

$$V_i^U \geq V_i^D, \forall i \in \mathcal{I}. \tag{9}$$

Since the UAVs and IoT devices have limited communication ranges, a UAV need to get sufficiently close the IoT device (i.e., in communication range) to capture its data successfully. Consequently, a variable (i.e.,  $c_i^u(t)$ ) has been established to monitor the connectivity status between a UAV and an IoT device. Downloading of data by a UAV (i.e.,  $d_i^u(t)$ ) is permissible only if a connectivity link is established. Nevertheless, there may be instances where a UAV is within the range of an IoT device but chooses not to download any data. This may happen when the resulting AoI is better in case any of other UAVs downloads and uploads that IoT's data.

$$c_i^u(t) = \begin{cases} 1, & \text{if } \text{dist}_{l_i}^{L_u(t)} \leq R_l \\ 0, & \text{otherwise.} \end{cases} \tag{10}$$

$$d_i^u(t) \leq c_i^u(t), \tag{11}$$

$$\forall t \in T_u, \forall i \in \mathcal{I}, \forall u \in \mathcal{U}.$$

<sup>2</sup> If an IoT device generates data multiple times, we can simply consider as if we have separate IoT devices, each generating one data, at the same location.

Each IoT device's data should be captured by one and only one UAV.

$$\sum_{\forall u \in \mathcal{U}} \sum_{\forall t \in T_u} d_i^u(t) = 1, \forall i \in \mathcal{I}. \quad (12)$$

Next, we need to record the time when a UAV captures the data of an IoT device (i.e.,  $V_i^D$ ). Since the value of  $d_i^u(t)$  is set to 1 only for one UAV throughout the entire timeline of all UAVs, by multiplying this value with the time variable, we obtain the exact moment when data is downloaded from an IoT device by a UAV.

$$V_i^D = \sum_{\forall u \in \mathcal{U}} \sum_{\forall t \in T_u} (d_i^u(t) \times t), \forall i \in \mathcal{I}. \quad (13)$$

Then, by multiplying the value of  $d_i^u$  by  $u$ , we can find out the id of the UAV that captures the IoT device  $i$ 's data.

$$U_D(i) = \sum_{u \in \mathcal{U}} \sum_{t \in T_u} (d_i^u \times u), \forall i \in \mathcal{I}. \quad (14)$$

Similar to the UAV-IoT connection, for a UAV to upload the collected data to a GBS, it must be within the communication range of that GBS. We monitor the UAV-GBS connection status using another variable,  $G_u(t)$ , which indicates whether the UAV  $u$  is connected to at least one GBS at time slot  $t$ . Furthermore, it should be mentioned that when the UAV is in the communication range of a GBS, it can upload data to that GBS (defined by  $p_i^u(t)$ ).

$$g_i^u(t) = \begin{cases} 1, & \text{if } \text{dist}_{k_i}^{L_u(t)} \leq R_G \\ 0, & \text{otherwise.} \end{cases} \quad \forall u \in \mathcal{U}, \forall t \in T_u, \forall i \in \mathcal{G} \quad (15)$$

$$G_u(t) = \min(1, \sum_{\forall i \in \mathcal{G}} g_i^u(t)), \forall u \in \mathcal{U}, \forall t \in T_u \quad (16)$$

$$p_i^u(t) \leq G_u(t), \forall u \in \mathcal{U}, \forall i \in \mathcal{I}, \forall t \in T_u. \quad (17)$$

Additionally, we impose a constraint to ensure that the UAVs upload the captured data. All data captured by the UAVs from the IoT devices must be uploaded to the GBSs.

$$\sum_{\forall u \in \mathcal{U}} \sum_{\forall t \in T_u} p_i^u(t) = 1, \forall i \in \mathcal{I}. \quad (18)$$

In order to compute the AoI of each data, we need to record its delivery time by UAV to a GBS (i.e.,  $V_i^U$ ). Given that each IoT's data is delivered only once, we leverage this condition by first multiplying  $u_i(t)$  by  $t$ , and summing this over all critical times.

$$V_i^U = \sum_{\forall u \in \mathcal{U}} \sum_{\forall t \in T_u} (p_i^u(t) \times t), \forall i \in \mathcal{I}. \quad (19)$$

Similarly, by multiplying the value of  $p_i^u$  by  $u$ , we obtain the id of the UAV that delivers the data obtained from the IoT device  $i$  to a GBS.

$$U_P(i) = \sum_{\forall u \in \mathcal{U}} \sum_{t \in T_u} (p_i^u \times u), \forall i \in \mathcal{I}. \quad (20)$$

This information is needed, as when there are multiple UAVs, we need to verify that the data is uploaded by the UAV that gathered the data from that IoT device. To this end, we also consider the following constraint.

$$U_D(i) = U_P(i), \forall i \in \mathcal{I}. \quad (21)$$

Finally, the path lengths of the UAV trajectories can be computed by

$$D_{sum} = \sum_{u \in \mathcal{U}} \sum_{i=0}^{2|I|} \text{dist}_{L_u^i}^{L_u^{i+1}}. \quad (22)$$

Note that these formulations will apply when there is only one UAV too. However, to speed up the running time of the ILP based solution for one UAV scenario, some constraints (e.g., (21)) can be removed as they will be always satisfied and will not be necessary.

## 4.2. Greedy heuristic approach

While the ILP based solution can provide the optimal UAV paths, it comes with high computational complexity, thus it is not practical. To address that, in this section, we present our heuristic based solution. We first start with the scenario where there is only one UAV, then discuss how it scales to the multi-UAV scenario.

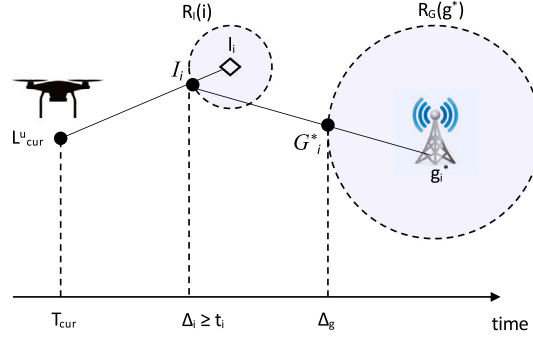


Fig. 3. Finding the IoT device that a UAV can deliver its data earliest from the current UAV location.

---

**Algorithm 1:** EarliestIoTDelivery ( $L_{cur}^u, T_{cur}^u$ )
 

---

**Input :**  $L_{cur}^u$ : Current location of UAV  
 $T_{cur}^u$ : Current time passed since start  
**Output:**  $B_i$ : Best IoT index  
 $B_g$ : Best/Nearest GBS to the best IoT  
 $T_{del}$ : Delivery time to the best IoT

```

1  $T_{del} = \infty$ 
2 foreach  $i \in I$  do
   // Find intersection point of UAV's trajectory with IoT i's range
3  $I_i \leftarrow \overline{L_{cur}^u} \cap R_I(i)$ 
   // Find time to reach IoT i's range
4  $\Delta_i \leftarrow \frac{\text{dist}(L_{cur}^u, I_i)}{V} + T_{cur}^u$ 
   // Update download time if arrived before data generation
5 if  $\Delta_i \geq t_i$  then
6   |  $\Delta_i \leftarrow t_i$ 
7 end
   // Closest GBS for IoT i
8  $g^* \leftarrow \arg \min_{g \in \mathcal{G}} \text{dist}(I_i, g)$ 
   // Find intersection point with the closest GBS's range
9  $G_i^* \leftarrow \overline{I_i g^*} \cap R_G(g^*)$ 
   // Delivery time for data of IoT i
10  $\Delta_g \leftarrow \Delta_i + \frac{\text{dist}(I_i, G_i^*)}{V}$ 
   // Keep the best one
11 if  $\Delta_g < T_{del}$  then
12   |  $B_i \leftarrow i$ 
13   |  $B_g \leftarrow g^*$ 
14   |  $T_{del} \leftarrow \Delta_g$ 
15 end
16 end
17 return ( $B_i, B_g, T_{del}$ )
  
```

---

#### 4.2.1. Single UAV

Our greedy heuristic based approach relies on sequentially integrating IoT devices and GBSs into the UAV's trajectory, while also avoiding exhaustive examination of all permutations. Thus, it enables us to approximate the optimal solution with significantly reduced computational time.

We start with a path that includes only the UAV's mission start and end locations. We then select an IoT device and try to place it in one of the available spots on the current trajectory. The available spots are considered as the positions between the consecutive elements of the current trajectory. That is, in the initial trajectory, there is only one position possible (i.e., between the starting



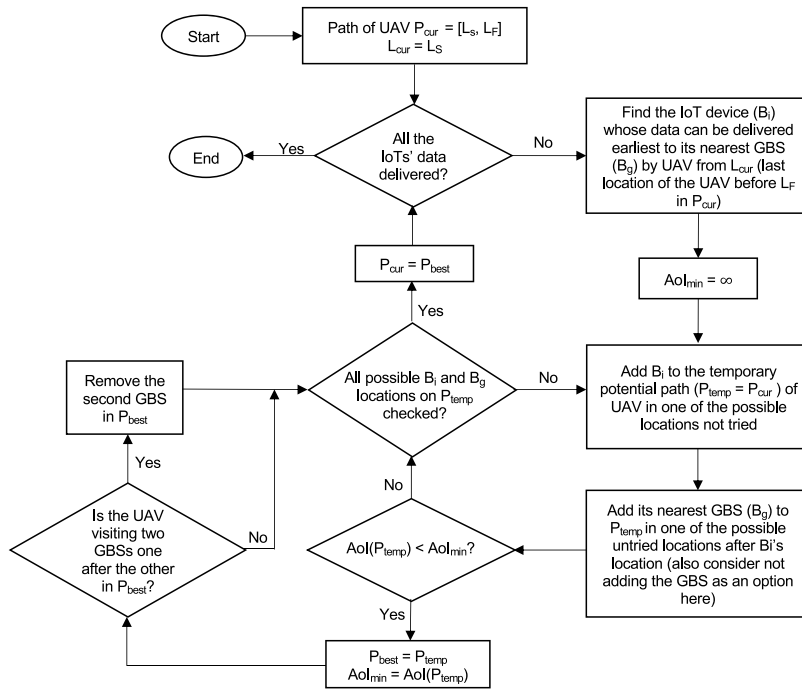


Fig. 4. Overview of heuristic based solution.

and end points of the UAV). In order to select the next IoT device to be added onto the trajectory, we use the strategy described in Alg. 1 and Fig. 3. That is, we find out the IoT device whose data can be delivered (to the nearest GBS) the earliest based on the UAV's current position. This greedy based selection of next IoT device aims to reduce AoI during the UAV's mission. Once an IoT device is added onto the UAV trajectory, we then consider inserting the GBS that allows that IoT device quickest delivery of its data. Here, we only consider the positions (between any consecutive element) after the last added IoT location. For example, once the first IoT is added, we only consider the one and only one position after this IoT device for a possible GBS insertion. However, we also consider not even inserting this GBS to the path. This is because it is possible that there may be other GBSs already in the current UAV trajectory which can upload the recently added IoT device's data without increasing the current AoI.

The process continues similarly until all the IoT devices are added in the UAV trajectory. That is, the next IoT to be added on the trajectory is determined by finding the IoT device whose data could be delivered the earliest from the current final position (and time) of the UAV on the current trajectory. This current final position is described as the position of the UAV before its final destination (e.g.,  $L_u^F$ ) in the current trajectory. Once the IoT to be added is determined, we try all possible spots to add it on the trajectory. As the UAV trajectory grows with the addition of new IoT devices and GBSs, the set of possible spots to add the next IoT device expands. For example, for the second IoT device insertion, since there will be the first IoT device and a GBS are already in the trajectory, there are three positions from which we need to find out the one that gives the smallest AoI and insert the second IoT there. Once the second IoT is added on the trajectory, depending on where it is added, there will be one to three different spots for the GBS (i.e., the closest GBS that would be used to upload the data of that IoT) to be added. Again, we try not only inserting the GBS to any of the spots after the second IoT on the trajectory but also try not inserting the GBS at all (so give chance to the UAV to upload this IoT device's data to an existing GBS on the trajectory) and proceed with the option that provides the best AoI. In the cases where inserting the GBS to the trajectory provides the same AoI (compared to not adding it), we opt to omit its insertion as we also want to minimize the travel distance of the UAV.

Upon finishing the insertion of all IoT devices, we calculate the AoI for the UAV's ultimate trajectory, which must include the defined start and end points, all IoT devices, and at least one GBS. This entire process is illustrated in Algorithm 2. In line 6, we find the next IoT to be inserted to the path and in lines 12–27, we try all possible options for both this IoT and its nearby GBS to find the best spots. In lines 28–34, we update the UAV trajectory with these best spots. If there are two GBSs consecutively, we need to remove the latter. In line 31, we check this condition to ensure that there are no two consecutive GBSs along the UAV's path. In lines 35–36 we update the location and the time for the UAV to be used in the next iteration.<sup>3</sup> An overview of this heuristic based approach is also illustrated in the flowchart given in Fig. 4.

<sup>3</sup> Note that both  $L_{cur}$  and  $T_{cur}$  can be calculated easily (using UAV speed and distances between waypoints) once the visit order in  $\mathcal{P}$  is known, and thus we omit the details for the sake of brevity.



**Algorithm 2:** GreedySingleUAVPathFormation

---

```

Input:  $L_S^u, L_F^u, I, G, U$ 
Output: UAV trajectory  $\mathcal{P}$ 

1  $C \leftarrow \emptyset$  // Checked IoTs
2  $\mathcal{P} \leftarrow [L_S^u, L_F^u]$  // Initialize UAV path with start and finish locations
3  $L_{cur}^u \leftarrow L_S^u$  // Current location of UAV
4  $T_{cur} \leftarrow 0$  // Current time
5 while  $|C| \neq |I|$  do
6    $(B_i, B_g) \leftarrow \text{EarliestIoTDelivery}(L_{cur}^u, T_{cur})$ 
7    $C \leftarrow C \cup \{B_i\}$ 
8    $P_{IoT} \leftarrow |\mathcal{P}| - 1$  // Max spots for IoT
9    $P_{GBS} \leftarrow |\mathcal{P}|$  // Max options for GBS
10   $L_{IoT} \leftarrow -1, L_{GBS} \leftarrow -1$ 
11   $\Delta \leftarrow \infty$ 
12  for  $i = 1$  to  $P_{IoT}$  do
13    for  $g = 0$  to  $P_{GBS}$  do
14      //  $g = 0$  is for not inserting the GBS
15       $\mathcal{P}_{temp} \leftarrow \mathcal{P}$ 
16      Insert  $B_i$  into  $\mathcal{P}_{temp}$  at spot  $i$ 
17      if  $g > i$  then
18        | Insert  $B_g$  into  $\mathcal{P}_{temp}$  at spot  $g$ 
19      else
20        |  $g \leftarrow -1$  // No GBS insertion
21      end
22      if  $\text{AoI}(\mathcal{P}_{temp}) < \Delta$  then
23        |  $\Delta \leftarrow \text{AoI}(\mathcal{P}_{temp})$ 
24        |  $L_{IoT} \leftarrow i$ 
25        |  $L_{GBS} \leftarrow g$ 
26      end
27    end
28   $\mathcal{P} \leftarrow (\mathcal{P}[1 : L_{IoT} - 1], B_i, \mathcal{P}[L_{IoT} : \text{end}])$  // Insert  $B_i$  at spot  $L_{IoT}$  in  $\mathcal{P}$ 
29  if  $L_{GBS} \neq -1$  then // Insert  $B_g$  at spot  $L_{GBS}$  in  $\mathcal{P}$ 
30     $\mathcal{P} \leftarrow (\mathcal{P}[1 : L_{GBS} - 1], B_g, \mathcal{P}[L_{GBS} : \text{end}])$ 
31    if  $\mathcal{P}[L_{GBS} + 1]$  is GBS then
32      | remove  $L_{GBS}$  from  $\mathcal{P}$ 
33    end
34  end
35   $L_{cur} \leftarrow \mathcal{P}[\text{length}(\mathcal{P}) - 1]$  // Update current location as the last UAV location before end point in  $\mathcal{P}$ 
36   $T_{cur} \leftarrow \text{TimeAt}(L_{cur})$  // Update current time as the UAV's arrival to  $L_{cur}$ 
37 end

```

---

*Complexity.* In Alg. 1, we check all unvisited IoT devices and the GBSs to find out the IoT device whose data can be delivered the earliest from the current UAV location. This takes  $\mathcal{O}(|I||G|)$ . For each such selected next IoT device and GBS, we then find the best spot on the current UAV path. Since there is at most one GBS needed per IoT device to upload its collected data (in some cases the data of multiple IoT devices can even be uploaded to the same GBS), the path length will be  $\mathcal{O}(|I|)$ , and thus each spot finding search for an IoT device or a GBS will take  $\mathcal{O}(|I|)$ . Overall, the complexity of the entire process is  $\mathcal{O}(|I|^2|G|(|I| + |G|))$ .

#### 4.2.2. Multiple UAVs

In this subsection, we discuss how the greedy heuristic developed for a single UAV is adapted for multi-UAV scenarios with the goal of minimizing the maximum AoI from any of the UAV trajectories. This adaptation mainly depends on the selection of the UAV that can deliver the next IoT (whose data is not collected yet) the earliest and the rest follows the same procedure as in single UAV scenario.

In Algorithm 3, we describe a pseudocode of this adaptation for multiple UAV scenarios. Initially, each UAV's trajectory consists of its start and end locations (lines 2–5). After finding the IoT devices that can be delivered by each UAV (lines 7–9), we find out the UAV that can achieve this delivery the earliest among all other UAVs (line 10). Then, we update the trajectory of this UAV by

**Algorithm 3: GreedyMultiUAVPathFormation**


---

**Input:**  $L_S^u, L_F^u, I, G, \mathcal{U}$   
**Output:** Trajectories for each  $u$  in  $\mathcal{U}$

```

1  $C \leftarrow \emptyset$ 
2 foreach  $u$  in  $\mathcal{U}$  do
3    $\mathcal{P}^u \leftarrow [L_S^u, L_F^u]$ 
4    $T_{cur}^u \leftarrow 0$ 
5 end
6 while  $|C| \neq |I|$  do
7   foreach  $u$  in  $\mathcal{U}$  do
8      $(B_i^u, B_g^u, T_{del}^u) \leftarrow \text{EarliestIoTDelivery}(L_{cur}^u, T_{cur}^u)$ 
9   end
10   $u_{min} \leftarrow \arg \min_u T_{del}^u$ 
11   $B_i \leftarrow B_i^{u_{min}}, B_g \leftarrow B_g^{u_{min}}$ 
12   $C \leftarrow C \cup \{B_i\}$ 
13  Insert  $B_i$  and  $B_g$  to  $\mathcal{P}^{u_{min}}$  and update  $L_{cur}^u, T_{cur}^u$  for  $u_{min}$  as in lines 12-37 in Algorithm 2
14 end
15 foreach  $u$  in  $\mathcal{U}$  do
16    $A_u = \text{Calculate AoI for } \mathcal{P}^u$ 
17 end
18 return  $\max_{u \in \mathcal{U}} A_u$ 

```

---

considering all possible spots on its trajectory for both the IoT it can deliver the earliest and its closest GBS similar to the steps described in single UAV scenario. Once the current time and location for this UAV is updated, and this IoT is marked as processed (line 12), we then repeat the procedure and find out the next UAV who can deliver its IoT the earliest and so on. Here, note that the current time for each UAV can be different and we take this into account while calculating the time of each UAV's next earliest delivery of any of the unmarked IoT devices. This allows us distributing the IoT devices among all UAVs in parallel. At the end, we calculate the AoI for each UAV's path and return the maximum (lines 15–18). Note that the flowchart in Fig. 4 can also be considered for multiple UAV scenario once the selection of next IoT device is made considering the delivery possible by any of the UAVs from their last locations. Then, we only update that UAV's path by finding the best spot for the selected IoT and the GBS that provides the earliest delivery.

*Complexity.* Since we find the earliest delivery time of any unvisited IoT for each UAV and select the minimum one as the next IoT device to be added on the selected UAV's path, the complexity of this part will be  $\mathcal{O}(|I||G||\mathcal{U}|)$ . The remaining process for finding the best spots for the selected IoT and GBS on the selected UAV's path will be similar to the single UAV scenario. Thus, the overall complexity will be  $\mathcal{O}(|I|^2|G||\mathcal{U}|(|I| + |G|))$ .

### 4.3. Brute-force approach

For comparison purposes, we also obtain results using a brute-force approach too. To this end, we obtain all the permutations of the IoT devices to first get a visit order in the UAV trajectories, which always begin at the starting location of the UAV and completes in the final location of the UAV. In the single UAV scenario, since all IoT devices should be visited by the same UAV, we consider a permutation that includes all the IoT devices. However, in the multiple UAV scenario, we first distribute the IoT devices to each UAV. It is possible that some UAVs may not be assigned any IoT devices at all. Once a UAV knows the ordered set of IoT devices that will be visited, we then consider adding one GBS to be visited before or after each IoT device visited. It is possible that there may not be a GBS added between some IoT device visits, as this is a possible scenario. Also, note that we do not need to consider more than one GBS between IoT device visits, as we assume in the first GBS visited, all the data from the visited IoT devices will be uploaded. For all possible such permutations of IoT devices and potentially visited GBSs in between, we calculate the AoI for the given scenario and find out the best result. In multiple UAV scenario, we compute the max AoI for each UAV's path and take the max of all.

*Complexity.* This approach tries all permutations of IoT devices together with all permutations of GBS devices placed in between IoT devices (with at most one GBS in between two consecutive IoT devices). Thus, it takes  $\mathcal{O}(|I||G|!)$  for a single UAV scenario. For multiple UAV scenario, the complexity will be much more as each IoT will initially be assigned to one of the UAVs. The number of such groupings can be found with Stirling number of second kind  $S(|I|, |\mathcal{U}|)$ , which is usually approximated with  $|\mathcal{U}|^{|\mathcal{I}|}/|\mathcal{U}|!$ . Considering that some UAVs may not be assigned any IoT at all, we will have  $\sum_{k=1}^{|\mathcal{U}|} \frac{k^{|\mathcal{I}|}}{k!}$ . For each of these cases, there will be permutations for each IoT set assigned to each UAV, thus the complexity will be much higher, which can be upper bounded by  $\mathcal{O}(\sum_{k=1}^{|\mathcal{U}|} \frac{k^{|\mathcal{I}|}}{k!} |I||G|!)$ .

**Table 3**  
Locations of GBSs.

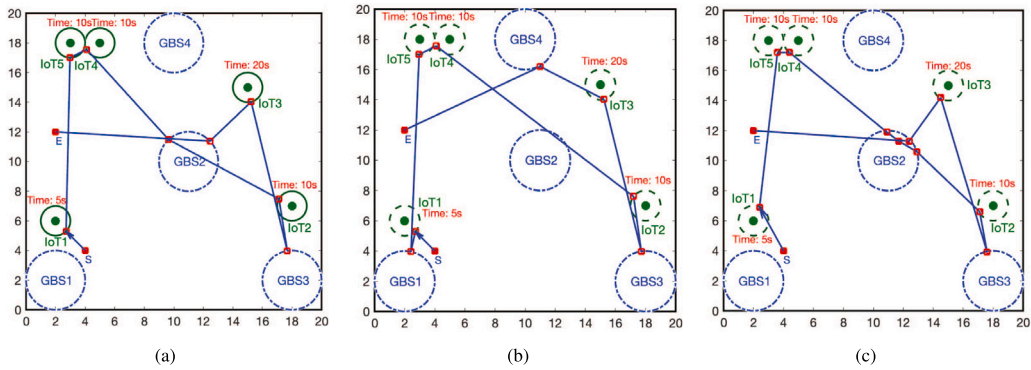
GBS ID	1	2	3	4
Coordinates	(2, 2)	(11, 10)	(18, 2)	(10, 18)

**Table 4**  
Locations of IoT devices and data generation times.

IoT ID	1	2	3	4	5
Coordinates	(2, 6)	(18, 7)	(15, 15)	(5, 18)	(3, 18)
Data generation times ( $t_i$ )	5	10	20	10	10

**Table 5**  
Simulation parameters.

Parameters	Values
UAV speed ( $V$ )	2 units/time unit
Map size	20 units $\times$ 20 units
GBS range ( $R_G$ )	2 units
IoT range ( $R_I$ )	1 unit
Number of IoTs	5
Number of GBSs	4
Scale (for ILP)	1,10,100



**Fig. 5.** Example with a single UAV: UAV trajectory obtained by (a) brute-force approach, (b) the greedy heuristic algorithm, and (c) the ILP based approach using CPLEX (with scale 10).

## 5. Simulation results

In this section, we provide our evaluations based on simulations. We start with showing how the proposed algorithm work in some specific example scenarios. We then look at their general performance using randomly created scenarios.

### 5.1. Toy examples

We begin with a hands-on example to demonstrate the step by step functionality of the proposed solutions, and how they compare to other solutions. We consider 5 IoT devices and 4 GBSs deployed on a map of 20 units by 20 units. The positions of GBSs and IoT devices are given in Tables 3 and 4, respectively.

#### 5.1.1. Single UAV

We start with a single UAV example where the UAV commences its mission from the coordinates (4,4), aiming to complete its mission at the location (2,12). The other parameter values are given in Table 5. Note that we use *unit* for the area and length metrics, and use *time unit* for time based measurements for the sake of simplicity. *Scale* for the ILP describes the granularity of the grid cell used for ILP based possible position descriptions. That is, for example, with scale 10, we consider 10 by 10 grid located on every 1 unit  $\times$  1 unit cell and thus can obtain locations with one decimal point. Similarly, with scale 100, we can get location information up to 2 decimal points and so on. More scale offers not only a smoother trajectory description but also can help find better solutions; however, it comes with the expense of much higher running time.

We start with brute-force results presented in Fig. 5(a), wherein all possible permutations of IoTs and GBSs on the trajectory of the UAV are explored and the one that provides the minimum AoI is obtained. In this scenario, it is observed that the UAV opts not

**Table 6**  
Steps in the Greedy heuristic algorithm in single UAV scenario.

Step	UAV path
0	$L_S \rightarrow L_F$
1	$L_S \rightarrow IoT_1 \rightarrow GBS_1 \rightarrow L_F$
2	$L_S \rightarrow IoT_1 \rightarrow GBS_1 \rightarrow IoT_2 \rightarrow GBS_3 \rightarrow L_F$
3	$L_S \rightarrow IoT_1 \rightarrow GBS_1 \rightarrow IoT_4 \rightarrow IoT_2 \rightarrow GBS_3 \rightarrow L_F$
4	$L_S \rightarrow IoT_1 \rightarrow GBS_1 \rightarrow IoT_5 \rightarrow IoT_4 \rightarrow IoT_2 \rightarrow GBS_3 \rightarrow L_F$
5	$L_S \rightarrow IoT_1 \rightarrow GBS_1 \rightarrow IoT_5 \rightarrow IoT_4 \rightarrow IoT_2 \rightarrow GBS_3 \rightarrow IoT_3 \rightarrow GBS_4 \rightarrow L_F$

**Table 7**  
Comparison of AoI for individual IoT devices with all algorithms in single UAV scenario (Maximum AoI is highlighted in bold).

Algorithms/IoT IDs	1	2	3	4	5
Brute-force	10.60	<b>11.61</b>	8.70	5.60	5.60
Greedy heuristic	0.68	<b>12.91</b>	12.17	12.91	12.91
ILP (Scale = 1)	8	<b>12</b>	10	3	3
ILP (Scale = 10*)	<b>10.3</b>	10.3	9.8	6	4.8
ILP (Scale = 100*)	9.96	<b>10.18</b>	8.84	<b>10.18</b>	<b>10.18</b>

to deliver data from  $IoT_1$  immediately. Instead, it proceeds to  $IoT_5$ , then to  $IoT_4$ , and delivers the data from all three IoT devices simultaneously at  $GBS_2$ . The UAV's path in this method is delineated as follows:

$$L_S \rightarrow IoT_1 \rightarrow IoT_5 \rightarrow IoT_4 \rightarrow GBS_2 \rightarrow IoT_2 \rightarrow GBS_3 \\ \rightarrow IoT_3 \rightarrow GBS_2 \rightarrow L_F.$$

In the greedy heuristic based approach we follow an iterative approach with greedy selections made as described in Algorithm 1. The final result is depicted in Fig. 5(b) with step by step changes made to the UAV path given in Table 6. The UAV path initially consists of the start and end locations. After finding out that  $IoT_1$  is the IoT whose data can be delivered earliest, it is added to the UAV path together with its closest GBS, i.e.,  $GBS_1$ . The algorithm then selects  $IoT_2$  as it can be delivered the earliest next, from the time and location at current final location before destination i.e.,  $GBS_1$ . After trying all possible spots, the algorithm then locates  $IoT_2$  after  $GBS_1$  and thus  $GBS_3$  is also added after that. At step 3,  $IoT_4$  is selected as the next IoT to be processed, which is then added to the spot between  $GBS_1$  and  $IoT_2$ . At this point, the nearby GBS associated with  $IoT_4$  (i.e.,  $GBS_4$ ) is omitted to be added on the trajectory as adding it to any of the spots does not help reduce the AoI. At step 4,  $IoT_5$  is selected and added after  $GBS_1$  and again its nearby GBS is omitted for the same reason. Finally, the algorithm adds  $IoT_3$  to the path and its associated GBS (i.e.,  $GBS_4$ ) to the end before the final destination.

Finally, we look at results obtained with ILP optimization. Here, we use different scales to obtain location information of the UAV on the map at different precision levels, as described earlier. As the scale grows, we can obtain more precise UAV location information and improved trajectories, however, the running time increases dramatically. Thus, we stop the CPLEX run after 2 h. ILP results are considered as the lowest AoI possible in a given scenario and are used as the baseline to compare the performance of other solutions. The UAV trajectory obtained when we use a scale of 10 is given Fig. 5(c). The path with scale 100 is very similar but is slightly smoother.

In Table 7, we present a comparative analysis of the AoI for individual IoT devices utilizing the aforementioned solutions. First of all, the ILP results with different scales show that the AoI decreases with more scale. Note that the ILP solutions for scale 10 and scale 100 do not reach 100% optimality (thus \* is used), as we have constrained the computation time to a maximum of 2 h for each scenario. Furthermore, it is evident from the comparison that ILP consistently yields the optimal and better solution relative to the other algorithms under consideration. Brute force and greedy heuristic methods result in closer but slightly higher AoI than ILP results. While the greedy heuristic result is the highest, as it will be discussed later, it can provide the result within seconds while both of the other methods have very long running times.

### 5.1.2. Multiple UAVs

For the multiple UAV scenario, we consider two UAVs with different starting and ending points. The first UAV begins and concludes its mission at a location with coordinates (2,11), while the starting and ending location for the other UAV is (2,12). We use the same GBSs as in single UAV scenario however, we consider six IoT devices with locations and data generation times given in Table 8.

In Fig. 6, we demonstrate the UAV paths resulted from each method together with individual AoI values for each IoT device on these trajectories given in Table 10. First of all, as shown on the map of each solution, the IoT devices end up being in two groups, with each group being served by one UAV. This is indeed the outcome of the two UAVs working in parallel to minimize the maximum AoI.

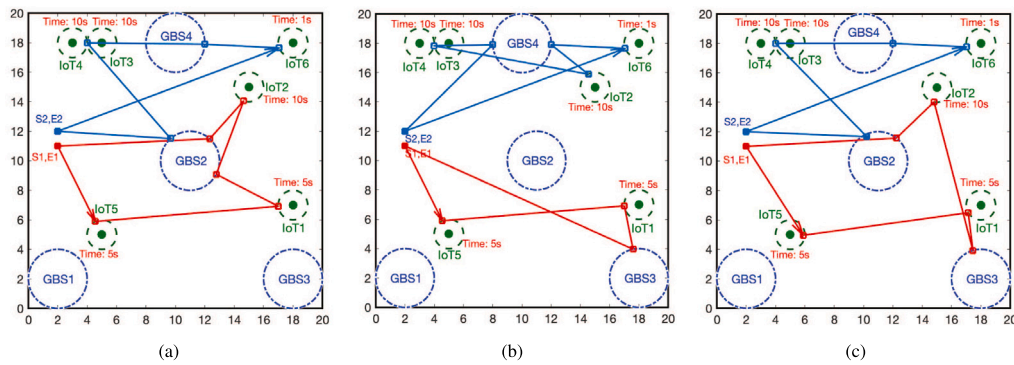


Fig. 6. Example with two UAVs: The UAV trajectories obtained by (a) brute force approach, (b) the greedy heuristic algorithm, and (c) the ILP based approach using CPLEX (scale = 10).

Table 8

Locations of IoT devices and data generation times in multi-UAV example.

IoT ID	Coordinates	Data generation times ( $t_i$ )
1	(18, 7)	5
2	(15, 15)	10
3	(5, 18)	10
4	(3, 18)	10
5	(5, 5)	5
6	(18, 18)	1

Table 9

Steps in the Greedy heuristic algorithm in multiple UAV scenario.

Step	UAV <sub>1</sub> 's path	UAV <sub>2</sub> 's path
0	$L_S^1 \rightarrow L_F^1$	$L_S^2 \rightarrow L_F^2$
1	$L_S^1 \rightarrow IoT_5 \rightarrow GBS_1 \rightarrow L_F^1$	$L_S^2 \rightarrow L_F^2$
2	$L_S^1 \rightarrow IoT_5 \rightarrow GBS_1 \rightarrow L_F^1$	$L_S^2 \rightarrow IoT_6 \rightarrow GBS_4 \rightarrow L_F^2$
3	$L_S^1 \rightarrow IoT_5 \rightarrow IoT_1 \rightarrow GBS_3 \rightarrow GBS_1 \rightarrow L_F^1$	$L_S^2 \rightarrow IoT_6 \rightarrow GBS_4 \rightarrow L_F^2$
4	$L_S^1 \rightarrow IoT_5 \rightarrow IoT_1 \rightarrow GBS_3 \rightarrow L_F^1$	$L_S^2 \rightarrow IoT_6 \rightarrow GBS_4 \rightarrow IoT_2 \rightarrow GBS_4 \rightarrow L_F^2$
5	$L_S^1 \rightarrow IoT_5 \rightarrow IoT_1 \rightarrow GBS_3 \rightarrow L_F^1$	$L_S^2 \rightarrow IoT_6 \rightarrow GBS_4 \rightarrow IoT_2 \rightarrow IoT_3 \rightarrow GBS_4 \rightarrow L_F^2$
6	$L_S^1 \rightarrow IoT_5 \rightarrow IoT_1 \rightarrow GBS_3 \rightarrow L_F^1$	$L_S^2 \rightarrow IoT_6 \rightarrow GBS_4 \rightarrow IoT_2 \rightarrow IoT_3 \rightarrow IoT_4 \rightarrow GBS_4 \rightarrow L_F^2$

Table 10

Comparison of AoI for individual IoT devices with all algorithms in multi-UAV scenario (Maximum AoI is highlighted in bold).

Algos/IoT IDs	1	2	3	4	5	6
Brute force	8.62	8.01	8.89	8.89	8.62	<b>9.58</b>
Greedy heuristic	7.75	<b>10.58</b>	<b>10.58</b>	<b>10.58</b>	7.75	9.58
ILP (Sc = 1*)	10	10	8	8	3	<b>11</b>
ILP (Sc = 10*)	8.8	3.6	9.4	9.4	7	<b>9.6</b>
ILP (Sc = 100*)	7.11	9.31	9.22	9.22	7.11	<b>9.57</b>

In brute-force results given in Fig. 6a, the first UAV collects data from  $IoT_5$ ,  $IoT_1$  and  $IoT_2$  while the second UAV collects data from  $IoT_6$ ,  $IoT_3$  and  $IoT_4$ . The  $GBS_2$  and  $GBS_4$  are also used to upload the collected data. Again, this is obtained by considering all possible distributions and permutations of IoT devices and GBSs among these two UAVs.

In greedy heuristic result given in Fig. 6b, we see a slight change in the distribution of IoTs among UAVs, with  $IoT_2$  being moved to the second UAV. The step by step growth of UAV paths are provided in Table 9. Initially, Algorithm 3 finds that the data of the  $IoT_5$  can be delivered the earliest by the first UAV; thus, it is added to the trajectory of the first UAV, together with its nearest GBS, i.e.,  $GBS_1$ . Then,  $IoT_6$  is identified as the next to be delivered earliest among the remaining IoT devices and this can be achieved by the second UAV. Thus, it is added to the path of the second UAV. The algorithm then selects  $IoT_1$  as the next one to be added to the first UAV's path, which is added to the best spot (i.e., after  $IoT_5$ ). Then, the algorithm checks if adding its nearest GBS (i.e.,  $GBS_3$ ) would help reduce the AoI. Since it does, it adds  $GBS_3$  to the best spot (i.e., after  $IoT_1$ ). At this moment, our algorithm also removes  $GBS_1$  as it is the second GBS there and unnecessary. In the rest of the steps, the algorithm adds  $IoT_2$ ,  $IoT_3$  and  $IoT_4$  onto the trajectory of the second UAV, while sometimes skipping the addition of their nearest GBS. In the final paths, the first UAV takes care of  $IoT_5$  and  $IoT_1$ , while the other IoT devices are handled by the second UAV.

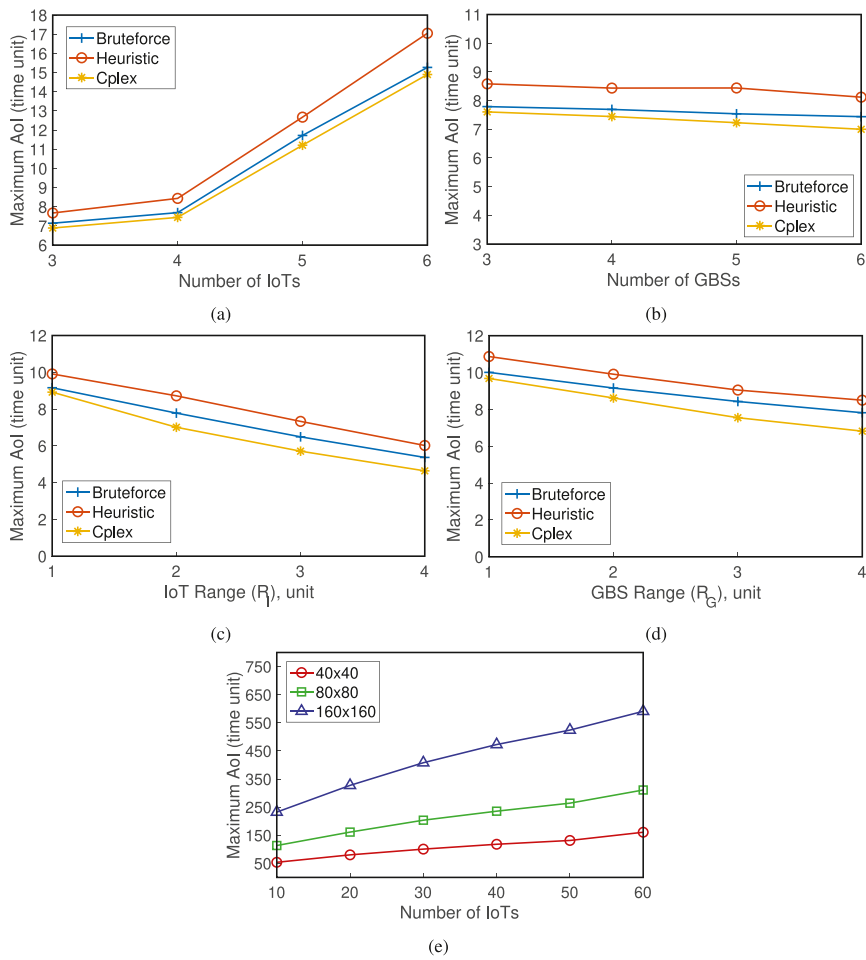


Fig. 7. Single UAV: Impact of varying (a) number of IoT devices (with  $|G| = 4$ ), (b) number of GBSSs (with  $|I| = 4$ ), (c)  $R_I$ , (d)  $R_G$  on maximum AoI. (e) Heuristic results with large number of IoT devices on different maps sizes.

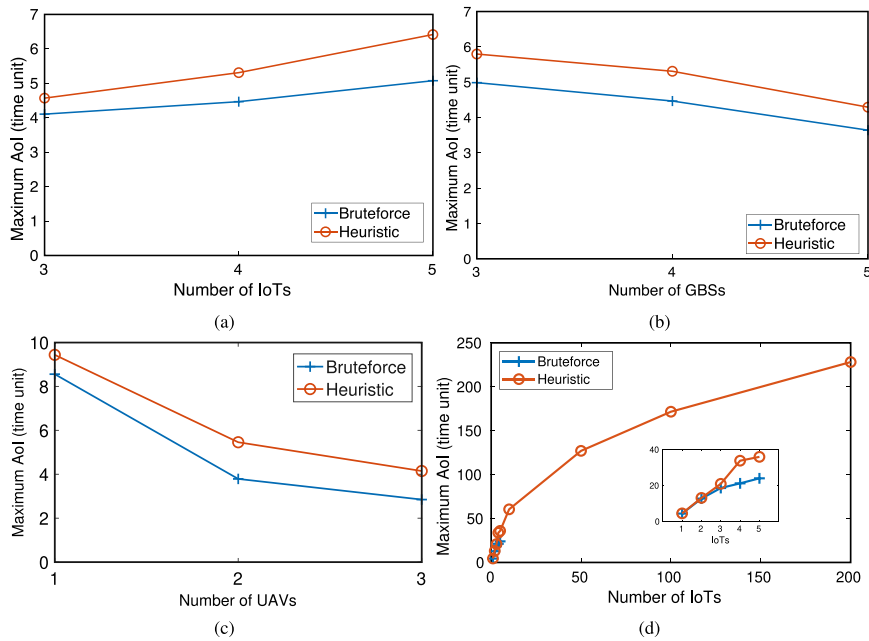
In Fig. 6c, we present the output of the ILP approach obtained by CPLEX using scale of 10. It provides very similar result to the brute-force solution with only a slight difference in the first UAV's path. However, it takes more time to produce this result. Looking at the results in Table 10, we see that brute force and ILP results are very close and the heuristic based approach can also give within around 10% of them. Considering that the running time benefit is huge, this difference however can be considered acceptable especially for real-time applications.

## 5.2. Random scenarios

In this part, in order to evaluate and compare the general performance of the heuristic solution with other methods, we consider 100 randomly determined scenarios using different settings.

### 5.2.1. Single UAV

We first consider the single UAV scenario with four GBSSs and with IoT device counts increasing from 3 to 5. We assign a data generation time between 0 and 20 time units for each IoT device. As Fig. 7(a) shows, the maximum AoI increases with increasing number of IoT devices for all algorithms. While the CPLEX results are the best as expected, the permutational brute force approach gives close to that, and the proposed greedy heuristic can provide slightly larger but similar results. Next, we check how the number of GBSSs affect the maximum AoI while the number of IoT devices is set to 4. Fig. 7(b) shows that as the number of GBSSs increases the maximum AoI reduces. This is because more GBSSs offer more coverage and thus faster delivery of data by UAVs. The relationship between the compared algorithms is also similar to the previous scenario. In Figs. 7(c) and 7(d), we look at the effect of IoT range ( $R_I$ ) and GBSS range ( $R_G$ ) on maximum AoI, respectively. As expected, maximum AoI decreases with increasing  $R_I$  or  $R_G$ , as they



**Fig. 8. Multi-UAV scenario:** Impact of varying number of (a) IoT devices (with  $|G| = 4$ ,  $|U| = 2$ ), (b) GBSs (with  $|I| = 4$ ,  $|U| = 2$ ), (c) UAVs (with  $|G| = 4$ ,  $|I| = 4$ ) on maximum AoI. (d) Effect of number of IoTs on the maximum AoI with 3 UAVs and when the map size is  $80 \times 80$ .

allow earlier capture and delivery of IoT data. The heuristic-based algorithm again follows a similar trend as other algorithms, with a slight difference in performance. Finally, in Fig. 7(e), we examine the scalability of our algorithm, specifically within the heuristic approach, as we could not get results with CPLEX and brute force for large number of IoT devices. The results demonstrate that as the number of IoT devices increases in the same area, maximum AoI increases but the increase in the slope decreases as the results converge. Note that when the map size gets larger, the AoI increases with the same number of IoT devices, but as new IoT devices are added to the area the convergence happens there eventually too.

### 5.2.2. Multiple UAVs

In this part, we obtain results for multiple UAV scenarios using the same randomized approach. To this end, we first consider  $|U| = 2$  UAVs and randomly create 100 different setups using a given number of IoT devices, while having a fixed number of 4 GBSs. We calculate the average maximum AoI. The data generation time for the IoT devices is again randomly set between 0 and 20 time units.

As depicted in Fig. 8(a), the maximum AoI increases as the number of IoT devices increases for both algorithms. We could not obtain CPLEX results for these scenarios as even after very long running times (e.g., 2 days in a server with an Intel Xeon X5680 CPU running at 3.33 GHz with 12 cores and 96 GB of memory), we could not get a result. Thus, we are presenting brute force results, for which we expect to be close to CPLEX results based on previous results presented. Heuristic approach provides more AoI than brute force, but the results are within 20%. We also observe that the gap between brute force and heuristic results increases with more number of IoTs, due to greedy design in the solution. Subsequently, we analyze the influence of the number of GBSs on this scenario, holding the number of IoT devices constant at 4. As the results in Fig. 8(b) show, the AoI decreases with more GBSs, as the UAVs find more opportunity to upload the collected data, and the gap between brute force and heuristic based approach gets smaller with more GBSs. Finally, in Fig. 8(c), we show the impact of different number of UAVs on AoI. As expected, with more number of UAVs, the data of IoTs are both collected and uploaded to a nearby GBS much faster, yielding a lower AoI. Greedy heuristic based approach can provide close to brute-force results, while running in very short time, which is pivotal for real-time applications requiring up-to-date information.

In Fig. 8(d), we also show the maximum AoI obtained with brute force and heuristic approaches with three UAVs and with larger number of IoTs. As the figure shows, with more number of IoT devices, the maximum AoI will increase but the increase rate will reduce. This is because the likelihood of getting larger AoI will reduce in a fixed space with a growing number of IoT devices. Note that it takes very long to get results with brute force, thus we could not obtain brute force results when IoT count is more than 5. However, as the trend with smaller number of IoTs show, the heuristic approach is inclined to follow the brute force results with some margin, while having very small computation cost.

### 5.2.3. Processing time

Next, we provide a comparison of the running times of different algorithms in different scenarios. We start with results with single UAV as presented in Fig. 9(a). The results clearly show that the processing time required for the ILP approach far exceeds



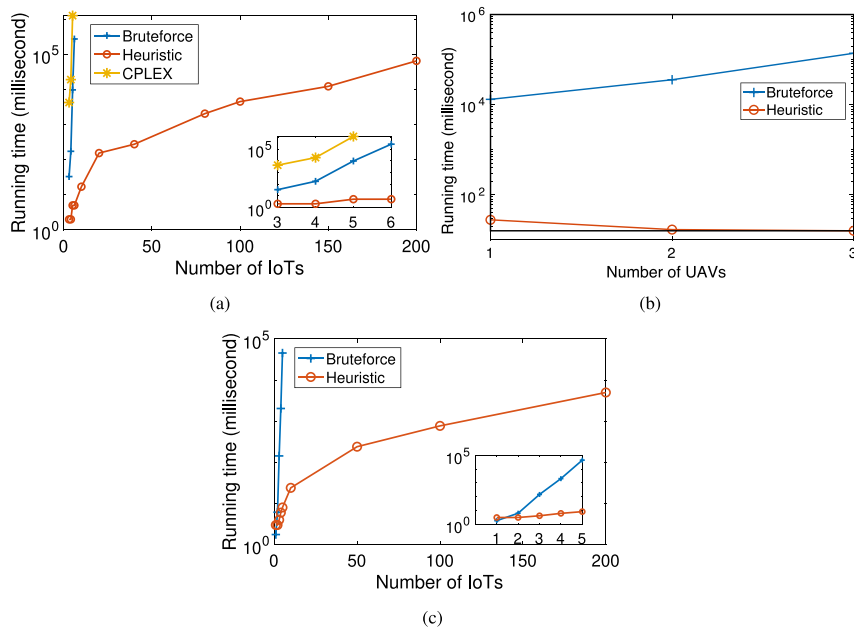


Fig. 9. Running time comparison (a) with different number of IoTs in single UAV case, (b) with different number of UAVs, (c) with different number of IoTs in three UAV scenario ( $M = 80 \times 80$ ).

that of the brute force method, which in turn significantly surpasses the time taken by the heuristic approach with small IoT counts, but becomes much higher than heuristic approach with large number of IoTs.

In Fig. 9(b), we show the running times of the heuristic and brute force approaches with increasing number of UAVs. As expected, the running time for the brute force approach escalates due to the increased number of potential scenarios that need exploration. In contrast, with the heuristic approach, increasing the number of UAVs leads to the distribution of IoT devices across more UAVs. This distribution simplifies the analysis by reducing the number of potential combinations that each UAV must evaluate, consequently decreasing the overall running time.

Next, we consider the scenario with 3 UAVs. As the results in Fig. 9(c) show, the running time increases heavily for brute force approach (CPLEX running times are much longer as we could not obtain results with running times significantly more than brute force running times), underscoring its impracticality for real-time operations within multi-UAV systems. In contrast, the heuristic approach demonstrates a remarkable reduction in processing time (compared to brute force), thereby making its suitability for scenarios demanding rapid decision-making and execution.

Overall, all these AoI and running time results show the trade-off involved in algorithm selection for multi-UAV scenarios. While the brute-force (and ILP based) method may offer better results its high computational cost renders it less feasible for dynamic and practical environments. On the other hand, the heuristic algorithm, by virtue of its design, offers a balanced compromise between information freshness and computational efficiency, making it a more viable choice for practical multi-UAV operations.

## 6. Conclusion

In this article, we explored the problem of AoI-centered trajectory planning for UAVs that have the mission of visiting the location of ground IoT devices and capturing their data in order to carry and deliver them to the nearby GBSSs in the area. The main objective was to optimize the UAV paths that will result in the minimum possible maximum AoI of any collected data by the UAVs, followed by some additional objectives for the smoothness of trajectories. Since we considered multiple delivery points (i.e., GBSSs) of the data, the definition of AoI was different from existing work, making the problem more challenging. The problem was first modeled and optimized using an ILP based solution approach. We then developed a heuristic based method with the goal of reducing the complexity of ILP based approach, while providing a cost-efficient approximate solution. Through extensive simulation results covering various scenarios, we have demonstrated that the proposed heuristic approach can produce UAV trajectories that result in AoI values that are close to the optimal values with much less running times, showing the practicality of the solution.

As the subject of our future efforts, we will aim to extend the proposed solution in an online manner. That is, the current study assumes that the data generation times of IoT devices are known in advance even if they are after the start of UAV missions. However, in a more practical world, this information can be known later, thus the UAV trajectories may need to be adapted during their missions. Additionally, for long UAV missions, charging needs of UAVs can be taken into account during these missions. Thus, we will also consider the availability of charging stations and online adaption of UAV trajectories accordingly.

## CRediT authorship contribution statement

**Amirahmad Chapnevis:** Writing – review & editing, Writing – original draft, Validation, Software, Methodology, Formal analysis, Conceptualization. **Eyuphan Bulut:** Writing – review & editing, Validation, Supervision, Software, Resources, Project administration, Investigation, Funding acquisition, Formal analysis, Conceptualization.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Data availability

No data was used for the research described in the article.

## References

- [1] Y. Zeng, Q. Wu, R. Zhang, Accessing from the Sky: A tutorial on UAV communications for 5G and beyond, *Proc. IEEE* 107 (12) (2019) 2327–2375.
- [2] K. Messaoudi, O.S. Oubati, A. Rachedi, A. Lakas, T. Bendouma, N. Chaib, A survey of UAV-based data collection: Challenges, solutions and future perspectives, *J. Netw. Comput. Appl.* (2023) 103670.
- [3] J. Su, X. Zhu, S. Li, W. Chen, AI meets UAVs: A survey on AI empowered UAV perception systems for precision agriculture, *Neurocomputing* 518 (2023) 242–270.
- [4] S. Ullah, K.-I. Kim, K.H. Kim, M. Imran, P. Khan, E. Tovar, F. Ali, UAV-enabled healthcare architecture: Issues and challenges, *Future Gener. Comput. Syst.* 97 (2019) 425–432.
- [5] A.S. Abdalla, V. Marojevic, Communications standards for unmanned aircraft systems: The 3GPP perspective and research drivers, *IEEE Commun. Stand. Mag.* 5 (1) (2021) 70–77.
- [6] M. Alam, N. Ahmed, R. Matam, F.A. Barbhuiya, IEEE 802.11 ah-enabled internet of drone architecture, *IEEE Internet Things Mag.* 5 (1) (2022) 174–178.
- [7] J. Liu, X. Wang, B. Bai, H. Dai, Age-optimal trajectory planning for UAV-assisted data collection, in: *IEEE Conference on Computer Communications Workshops, INFOCOM WKSHPS*, 2018, pp. 553–558.
- [8] M.A. Abd-Elmagid, A. Ferdowsi, H.S. Dhillon, W. Saad, Deep reinforcement learning for minimizing age-of-information in UAV-assisted networks, in: *IEEE Global Communications Conference, GLOBECOM*, 2019, pp. 1–6.
- [9] S. Zhang, H. Zhang, Z. Han, H.V. Poor, L. Song, Age of information in a cellular internet of UAVs: Sensing and communication trade-off design, *IEEE Trans. Wireless Commun.* 19 (10) (2020) 6578–6592.
- [10] C. Zhan, Y. Zeng, Aerial-ground cost tradeoff for multi-UAV-enabled data collection in wireless sensor networks, *IEEE Trans. Commun.* 68 (3) (2019) 1937–1950.
- [11] J. Li, Y. Xiong, J. She, UAV path planning for target coverage task in dynamic environment, *IEEE Internet Things J.* 10 (20) (2023) 17734–17745.
- [12] A. Chapnevis, I. Güvenç, L. Njilla, E. Bulut, Collaborative trajectory optimization for outage-aware cellular-enabled UAVs, in: *IEEE 93rd Vehicular Technology Conference, VTC2021-Spring*, 2021, pp. 1–6.
- [13] W.J. Lau, J.M.-Y. Lim, C.Y. Chong, N.S. Ho, T.W.M. Ooi, General outage probability model for UAV-to-UAV links in Multi-UAV networks, *Comput. Netw.* 229 (2023) 109752.
- [14] K.K. Nguyen, T.Q. Duong, T. Do-Duy, H. Claussen, L. Hanzo, 3D UAV trajectory and data collection optimisation via deep reinforcement learning, *IEEE Trans. Commun.* 70 (4) (2022) 2358–2371.
- [15] A. Chapnevis, E. Bulut, AoI-optimal cellular-connected UAV trajectory planning for IoT data collection, in: *IEEE 48th Conference on Local Computer Networks, LCN*, 2023, pp. 1–6.
- [16] M. Mozaffari, W. Saad, M. Bennis, Y.-H. Nam, M. Debbah, A tutorial on UAVs for wireless networks: Applications, challenges, and open problems, *IEEE Commun. Surv. Tutor* 21 (3) (2019) 2334–2360.
- [17] B. Li, Z. Fei, Y. Zhang, M. Guizani, Secure UAV communication networks over 5G, *IEEE Wirel. Commun.* 26 (5) (2019) 114–120.
- [18] Y. Ding, Q. Zhang, W. Lu, N. Zhao, A. Nallanathan, X. Wang, X. Yang, Collaborative communication and computation for secure UAV-enabled MEC against active aerial eavesdropping, *IEEE Trans. Wirel. Commun.* (2024).
- [19] Y. Ding, Y. Feng, W. Lu, S. Zheng, N. Zhao, L. Meng, A. Nallanathan, X. Yang, Online edge learning offloading and resource management for UAV-assisted MEC secure communications, *IEEE J. Sel. Top. Signal Process.* 17 (1) (2022) 54–65.
- [20] S. Aggarwal, N. Kumar, Path planning techniques for unmanned aerial vehicles: A review, solutions, and challenges, *Comput. Commun.* 149 (2020) 270–299.
- [21] M. Jones, S. Djahel, K. Welsh, Path-planning for unmanned aerial vehicles with environment complexity considerations: A survey, *ACM Comput. Surv.* 55 (11) (2023) 1–39.
- [22] X. Huang, X. Fu, Fresh data collection for UAV-assisted IoT based on aerial collaborative relay, *IEEE Sens. J.* 23 (8) (2023) 8810–8825.
- [23] Y. Zeng, R. Zhang, Energy-efficient UAV communication with trajectory optimization, *IEEE Trans. Wirel. Commun.* 16 (6) (2017) 3747–3760.
- [24] E. Bulut, I. Guevenc, Trajectory optimization for cellular-connected UAVs with disconnectivity constraint, in: *IEEE International Conference on Communications (ICC) Workshops*, 2018, pp. 1–6.
- [25] Q. Wu, Y. Zeng, R. Zhang, Joint trajectory and communication design for multi-UAV enabled wireless networks, *IEEE Trans. Wireless Commun.* 17 (3) (2018) 2109–2121.
- [26] J. Li, Y. Xiong, J. She, M. Wu, A path planning method for sweep coverage with multiple UAVs, *IEEE Internet Things J.* 7 (9) (2020) 8967–8978.
- [27] M.A. Abd-Elmagid, H.S. Dhillon, Average peak age-of-information minimization in UAV-assisted IoT networks, *IEEE Trans. Veh. Technol.* 68 (2) (2018) 2003–2008.
- [28] C. Liu, Y. Guo, N. Li, X. Song, AoI-Minimal task assignment and trajectory optimization in multi-UAV-assisted IoT networks, *IEEE Internet Things J.* 9 (21) (2022) 21777–21791.
- [29] B. Zhu, E. Bedeer, H.H. Nguyen, R. Barton, Z. Gao, UAV trajectory planning for AoI-Minimal data collection in UAV-aided IoT networks by transformer, *IEEE Trans. Wirel. Commun.* 22 (2) (2023) 1343–1358.
- [30] Z. Zhou, J. Liu, C. Mao, Age of information oriented data collection via energy-constrained UAVs in wireless sensor networks, *IEEE Access* 12 (2024) 11897–11908.

- [31] X. Zhang, Z. Chang, T. Hämäläinen, G. Min, AoI-Energy tradeoff for data collection in UAV-assisted wireless networks, *IEEE Trans. Commun.* (2023).
- [32] H. Shen, D. Wang, Z. Huang, Y. Jia, Optimization of clustering and trajectory for minimizing age of information in unmanned aerial vehicle-assisted mobile edge computing network, *Sensors* 24 (6) (2024) 1742.
- [33] O. Rahimi, A. Shafieinejad, Minimizing age of information in multi-UAV-assisted IoT networks: a graph theoretical approach, *Wirel. Netw.* 30 (1) (2024) 533–555.
- [34] O.S. Oubbati, M. Atiquzzaman, H. Lim, A. Rachedi, A. Lakas, Synchronizing UAV teams for timely data collection and energy transfer by deep reinforcement learning, *IEEE Trans. Veh. Technol.* 71 (6) (2022) 6682–6697.
- [35] Y. Li, H. Ding, Z. Yang, B. Li, Z. Liang, Integrated trajectory optimization for UAV-enabled wireless powered MEC system with joint energy consumption and AoI minimization, *Comput. Netw.* (2024) 110842.
- [36] K. Messaoudi, A. Baz, O.S. Oubbati, A. Rachedi, T. Bendouma, M. Atiquzzaman, UGV charging stations for UAV-assisted AoI-Aware data collection, *IEEE Trans. Cogn. Commun. Netw.* (2024).
- [37] G. Chen, C. Cheng, X. Xu, Y. Zeng, Minimizing the age of information for data collection by cellular-connected UAV, *IEEE Trans. Veh. Technol.* 72 (7) (2023) 9631–9635.
- [38] E. Eldeeb, D.E. Pérez, J.M. de Souza Sant’Ana, M. Shehab, N.H. Mahmood, H. Alves, M. Latva-aho, A learning-based trajectory planning of multiple UAVs for AoI minimization in IoT networks, in: 2022 Joint European Conference on Networks and Communications & 6G Summit, EuCNC/6G Summit 2022, Grenoble, France, June 7–10, 2022, IEEE, 2022, pp. 172–177.
- [39] M. Yi, X. Wang, J. Liu, Y. Zhang, B. Bai, Deep reinforcement learning for fresh data collection in UAV-assisted IoT networks, in: IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), 2020, pp. 716–721.
- [40] A. Ferdowsi, M.A. Abd-Elmagid, W. Saad, H.S. Dhillon, Neural combinatorial deep reinforcement learning for age-optimal joint trajectory and scheduling design in UAV-assisted networks, *IEEE J. Sel. Areas Commun.* 39 (5) (2021) 1250–1265.
- [41] Y. Yang, X. Liu, Deep reinforcement learning based trajectory optimization for UAV-enabled IoT with SWIPT, *Ad Hoc Netw.* 159 (2024) 103488.
- [42] M.M.U. Chowdhury, S.J. Maeng, E. Bulut, I. Güvenç, 3-D trajectory optimization in UAV-assisted cellular networks considering antenna radiation pattern and backhaul constraint, *IEEE Trans. Aerosp. Electron. Syst.* 56 (5) (2020) 3735–3750.