

Article

Optimization of Clustering and Trajectory for Minimizing Age of Information in Unmanned Aerial Vehicle-Assisted Mobile Edge Computing Network

Huicong Shen , Die Wang , Zhen Huang and Yunjian Jia * 

School of Microelectronics and Communication Engineering, Chongqing University, Chongqing 401331, China; huicongshen@stu.cqu.edu.cn (H.S.); wangdie@cqu.edu.cn (D.W.); zhenhuang@cqu.edu.cn (Z.H.)

* Correspondence: yunjian@cqu.edu.cn

Abstract: With the development of the Internet of Things (IoT) technology, massive amounts of sensor data in applications such as fire monitoring need to be transmitted to edge servers for timely processing. However, there is an energy-hole phenomenon in transmitting data only through terrestrial multi-hop networks. In this study, we focus on the data collection task in an unmanned aerial vehicle (UAV)-assisted mobile edge computing (MEC) network, where a UAV is deployed as the mobile data collector for the ground sensor nodes (SNs) to ensure high information freshness. Meanwhile, the UAV is equipped with an edge server for data caching. We first establish a rigorous mathematical model in which the age of information (AoI) is used as a measure of information freshness, related to both the data collection time and the UAV's flight time. Then a mixed-integer non-convex optimization problem is formulated to minimize the peak AoI of the collected data. To solve the problem efficiently, we propose an iterative two-step algorithm named the AoI-minimized association and trajectory planning (AoI-MATP) algorithm. In each iteration, the optimal SN-collection point (CP) associations and CP locations for the parameter ϵ are first obtained by the affinity propagation clustering algorithm. The optimal UAV trajectory is found using an improved elite genetic algorithm. Simulation results show that based on the optimized ϵ , the AoI-MATP algorithm can achieve a balance between data collection time and flight time, reducing the peak AoI of the collected data.

Keywords: Internet of Things; mobile edge computing; unmanned aerial vehicle; age of information; clustering; trajectory



Citation: Shen, H.; Wang, D.; Huang, Z.; Jia, Y. Optimization of Clustering and Trajectory for Minimizing Age of Information in Unmanned Aerial Vehicle-Assisted Mobile Edge Computing Network. *Sensors* **2024**, *24*, 1742. <https://doi.org/10.3390/s24061742>

Received: 18 January 2024

Revised: 6 March 2024

Accepted: 6 March 2024

Published: 7 March 2024



Copyright: © 2024 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

In the era of 5G, the flourishing progress of the Internet of Things (IoT) technology is instigating transformative shifts in emerging application domains. Leveraging the high-speed and low-latency attributes of 5G networks [1], wireless sensor networks (WSNs) containing a large number of sensors have become the backbone of the IoT, demonstrating an impressive capability to transmit massive amounts of data with exceptional efficiency. Consequently, WSNs can provide unprecedented real-time sensing and decision support for status-update systems with stringent real-time demands. It has emerged as the central driving force behind applications such as smart transportation [2], environmental monitoring [3], and smart agriculture [4]. However, ground sensor nodes (SNs) used for monitoring in WSNs usually suffer from small size and limited battery capacity. In some remote areas, SNs are far away from data centers. Transmitting the data sensed by SNs only through the ground-based multi-hop network is prone to the energy hole phenomenon [5]. It cannot satisfy the high information freshness in the state update system, affecting the timeliness and accuracy of decision-making.

Therefore, in real-time sensing scenarios such as forest fire monitoring [6] and disaster monitoring [7], the use of mobile data collectors like unmanned aerial vehicles (UAV) is considered a promising solution [8]. Recently, UAVs have attracted extensive attention due

to their significant advantages [9,10]. First, UAVs can fly to inaccessible places to collect data, improving the completeness and accuracy of data collection compared to traditional collectors. Second, they have a high flight speed and bandwidth, making data collection fast and efficient. Third, they can be equipped with mobile edge computing (MEC) servers for real-time sensing, data processing, and decision-making. By processing data instantly on the fly, the latency of data transmission to the cloud is reduced, resulting in a more rapid response. Moreover, since UAVs usually have limited computing, storage, and power resources, some computationally intensive tasks may exceed their processing capabilities. Equipping MEC servers can provide additional computational and storage resources to optimize the efficiency of task execution. Finally, they can move close to the ground SNs, making it easy to establish line of sight (LoS) links and reduce the transmit power and packet loss rate [11]. Currently, data collection in UAV-assisted MEC networks has become a popular research topic, and scholars have carried out a lot of work accordingly.

Due to the limited energy onboard the UAV, a part of the existing work focuses on minimizing energy consumption in data collection tasks [12–16]. Some scholars have considered the impact of different data collection modes on UAV energy consumption, such as hover mode [12], flight mode [13], and hybrid mode [14]. In comparison, the flight mode has the lowest energy consumption but requires a long task completion time, while the hover mode requires higher energy consumption but achieves the minimum task completion time. In [15], a framework for data collection in UAV-assisted WSNs was considered and a sensor-energy-based method for initial UAV trajectory determination was proposed to maximize the minimum residual energy of the SNs after data transmission. To improve the security and energy efficiency of data collection in IoT applications, an adaptive linear prediction algorithm was designed to reduce the energy consumption of the WSN [16]. Furthermore, in [17], a binary search algorithm based on successive convex approximation (SCA) was designed to minimize the collection time, and the Frequency Division Multiple Access (FDMA) scheme was found to be superior to the simpler Time Division Multiple Access (TDMA) scheme by comparison. The algorithm proposed in [18] maximized the minimum data collection rate by jointly optimizing the 3D trajectory of the UAV and the time allocation.

However, in state update systems, outdated data can lead to wrong decisions, so there is an extremely high requirement for data freshness. Generally, the age of information (AoI) is used to characterize data freshness, which portrays the time elapsed since the generation of the latest incoming update [19]. Therefore, many scholars have considered minimizing the peak AoI or average AoI in data collection tasks. On the one hand, when the number of sensors is large, the data collection strategy plays an important role in reducing the AoI. References [20–24] utilized clustering algorithms to divide the network into multiple sub-networks, and the UAV only needed to collect data at the corresponding data collection points (CPs) of each sub-network, thus reducing the AoI. Commonly used algorithms include simple adaptive algorithms [22], k-means clustering algorithms [23], and affinity propagation (AP) algorithms [24]. For different system models and performance metrics, it is necessary to choose the appropriate clustering algorithm. Moreover, the authors proposed an aerial collaborative relay-based data collection scheme to obtain better AoI performance in [25]. In [26], the data collection method, the energy consumption, and the age evolution were comprehensively optimized to minimize the average AoI.

On the other hand, the UAV's flight trajectory greatly affects the time required for the data collection task. The high flexibility of UAVs imposes challenges on their trajectory planning, and much attention has been paid to deriving the optimal trajectory of UAVs from source to destination under given conditions. The authors of [27] jointly optimized the UAV's trajectory and the service time allocation to minimize the SN's average AoI. In [28], an algorithm based on graph theory and the kernel K-means method and an energy-constrained trajectory planning algorithm were proposed to reduce the average AoI and the energy consumption of each UAV. The authors jointly optimized the collection time and the trajectory of the UAV to achieve a trade-off between AoI and energy consumption

in [29]. By decomposing the optimization problem, the sub-problems were solved using SCA and Lagrange duality methods, respectively. An iterative algorithm was proposed for minimizing the peak and average AoI of multi-UAV-assisted data collection in [30], where an improved ant colony optimization (ACO) algorithm was used to optimize the UAVs' trajectories. In [31], the authors proposed an average AoI-minimizing trajectory design based on the Safe-Twin Delayed Deep Deterministic policy gradient. In general, UAV trajectory planning algorithms can be broadly classified into graph theory-based algorithms, optimization theory-based algorithms, and artificial intelligence (AI)-based algorithms in UAV-assisted data collection scenarios. Since the computational complexity of traditional optimization-based algorithms mostly grows exponentially with the system size, heuristic algorithms such as the ACO algorithm and genetic algorithm (GA) become better choices that can obtain a near-optimal solution in a shorter time. However, they are not suitable for real-time planning in rapidly changing environments, making online trajectory optimization considered to be implemented using machine learning algorithms [32].

The findings of the above work show that the collection method and the trajectory of the UAV have a significant impact on the AoI performance in data collection tasks. However, in related work [25,27,33], the non-line of sight (NLoS) communication link between the UAV and the ground SN was ignored, resulting in a model that may not be universal. Our work, on the other hand, builds a more rigorous mathematical model based on actual scenarios by taking into account the probability of both LoS and NLoS links occurring. Moreover, considering the case of a large number of SNs, different from directly performing trajectory planning [34], we first adopt a suitable clustering algorithm to divide the SNs into clusters. Meanwhile, similar work [24] failed to strike a balance between the accuracy and computational complexity of the solution, resulting in the computational complexity becoming intolerable when the problem size is large. Accordingly, our work is inspired to combine SN clustering with heuristic trajectory planning algorithms. The optimal collection strategies and offline flight trajectories for data collection scenarios in UAV-assisted MEC networks are investigated to minimize the peak AoI of the collected data, and the main contributions are as follows:

- We establish a rigorous mathematical model for the considered UAV-assisted MEC network, and we formulate a minimization problem for the peak AoI of the collected data, which is solved by jointly optimizing the SN–collection point (CP) association, the positions of the CPs, and the UAV trajectory.
- To solve this non-convex optimization problem accurately and efficiently, we propose a two-step iterative solution algorithm. First, an AP-based algorithm is applied to solve the optimal SN-CP associations and locations of CPs. In the second step, an improved elite genetic algorithm is used to obtain the AoI-optimal UAV trajectory.
- We control the number of clusters by varying the clustering parameter ε to achieve an optimal balance between data collection time and flight time. Accordingly, the optimal solution can be obtained by the ε that minimizes the peak AoI.

The remainder of this article is organized as follows. In Section 2, the system model for UAV-assisted data collection in MEC networks is presented. Section 3 formulates a minimization problem on the peak AoI of SNs' collected data. Section 4 describes the proposed iterative optimization algorithms in detail. Section 5 demonstrates the simulation results to evaluate the performance of the proposed algorithms. Finally, this article is concluded in Section 6. The overall framework diagram of this article is presented in Figure 1.

Notations: In this paper, scalars are denoted by italic letters and sets by calligraphic letters, such as \mathcal{M} . Furthermore, vectors and matrices are denoted by bold letters. $\mathbb{R}^{M \times K}$ denotes the space of $M \times K$ -dimensional real-valued matrices. $\|\cdot\|$ denotes the Euclidean norm of a vector. $|\cdot|$ represents the cardinality of a set, i.e., the number of elements in a finite set. $[\cdot]$ denotes an array, i.e., an ordered sequence of elements. $O(\cdot)$ denotes the standard big-O notation.

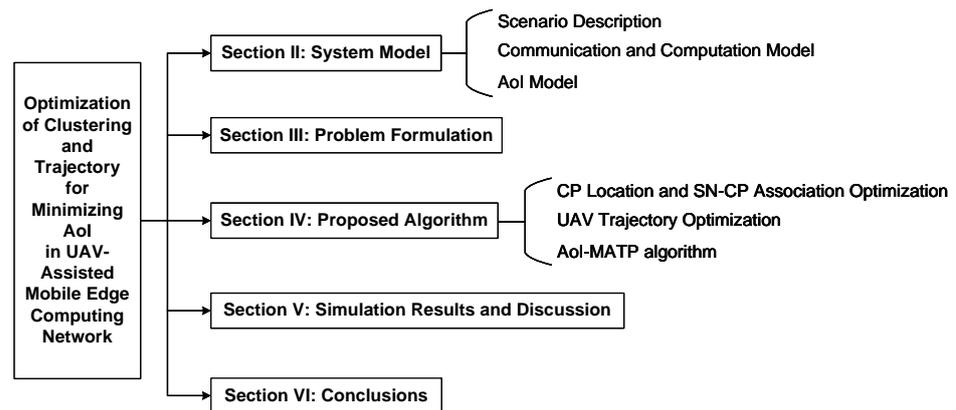


Figure 1. The overall framework diagram of this article.

2. System Model

2.1. Scenario Description

We consider a UAV-assisted MEC network for fire monitoring, as shown in Figure 2, in which a rotary-wing UAV is deployed as a mobile data collector for the M ground SNs. The UAV is outfitted with an edge server to enable real-time environmental sensing and data caching. It is assumed that the UAV has ample storage for the collected data, which it then transmits to the MEC server s_0 , thereby facilitating subsequent information processing steps. The SNs are denoted by $\mathcal{M} = \{1, 2, \dots, M\}$. Within a two-dimensional Cartesian coordinate system, s_0 occupies a fixed position at $\mathbf{w}_0 = (0, 0)$. The SNs are scattered randomly across the specified geographical area, and the location of SN m is expressed as $\mathbf{w}_m = (x_m, y_m)$. The UAV maintains a constant flight altitude H_u , ensuring operational safety by avoiding collisions with other environmental objects. Data collection by the UAV is conducted across a set of data CPs, denoted by $\mathcal{K} = \{1, 2, \dots, K\}$, where K is an element in the set \mathcal{M} , i.e., $K \in \mathcal{M}$. The coordinates of CP k projected on the ground are denoted by $\mathbf{u}_k = (X_k, Y_k)$.

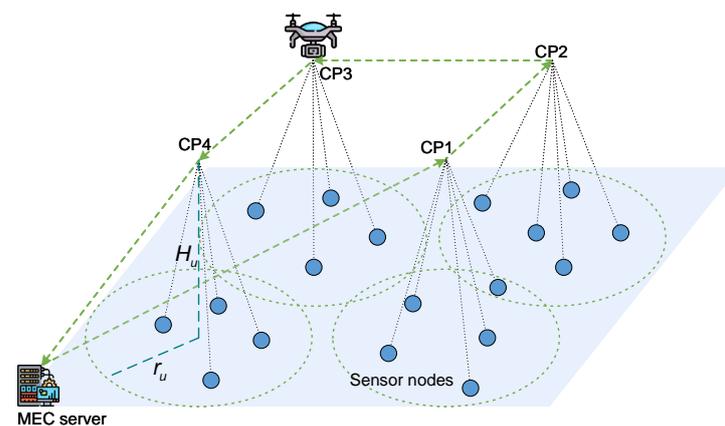


Figure 2. Illustration of the system scenario when $K = 4$.

Note that during the data collection phase, the SNs are grouped into K clusters corresponding to the CPs. The UAV starts from s_0 and sequentially visits K CPs to collect data, subsequently returning to s_0 to conclude a data collection cycle. Specifically, at each CP, the UAV hovers to establish communication links with all detectable SNs within that vicinity until all pertinent data are captured. Then the UAV flies at a constant speed V to the next CP. The monitoring radius accessible to the UAV is denoted by r_u . To ensure

accurate data collection, the distance between CP k and any associated SN m should satisfy the following conditions:

$$\|\mathbf{w}_m - \mathbf{u}_k\| \leq r_u, \quad m \in \mathcal{M}, k \in \mathcal{K}. \quad (1)$$

For ease of exposition, the binary indicator o_m^k is used to represent the association relationship between any CP k and SN m , i.e., whether the data of SN m can be collected when the UAV hovers at CP k . More specifically, $o_m^k = 1$ when SN m is associated with CP k ; otherwise, $o_m^k = 0$. Then, the association relationship between the SNs and CPs can be denoted by a sparse matrix $\mathbf{o} \in \mathbb{R}^{M \times K}$. At each CP, the UAV can cover multiple SNs, but each SN should only be associated with one CP. Therefore, for any SN m , we have

$$\sum_{k=1}^K o_m^k = 1, \quad k \in \mathcal{K}, m \in \mathcal{M}, \quad (2)$$

$$o_m^k \in \{0, 1\}, \quad k \in \mathcal{K}, m \in \mathcal{M}. \quad (3)$$

Let \mathcal{A}_k indicate the set of SNs associated with CP k ; then, the expression $\mathcal{A}_k = \{m \mid o_m^k = 1, m \in \mathcal{M}\}$ holds. The cardinality of the set \mathcal{A}_k is denoted by $|\mathcal{A}_k|$, which in this case parallels the number of SNs associated with CP k .

2.2. Communication and Computation Model

In this section, a popular ground-to-air path loss model is used to illustrate the communication between the UAV and SNs on the ground. Specifically, the distance between CP k and SN m is denoted d_m^k , which can be obtained by $d_m^k = \sqrt{\|\mathbf{w}_m - \mathbf{u}_k\|^2 + H_u^2}$. Our model takes into account both LoS and NLoS links between the UAV and its target. It should be noted that LoS links and NLoS links are two fundamental types of signal propagation paths in wireless communication. LoS links refer to situations where there is a direct line of sight between the transmitter and receiver, allowing wireless signals to propagate directly without any obstructions. These links typically offer higher signal quality and lower propagation loss. NLoS links refer to situations where there is no direct line of sight between the transmitter and receiver, requiring signals to reach the receiver through diffraction, reflection, or refraction. These links may encounter more attenuation and interference during signal propagation. Referring to the widely used channel model in [35,36], the probability of an LoS link occurring can be obtained by $P_{LoS}(d_m^k) = \frac{1}{1+a \exp(-b(\theta(d_m^k)-a))}$, where a and b are constant parameters determined by the environment and $\theta(d_m^k)$ is the degree of elevation angle between the UAV and the SN, expressed as $\theta(d_m^k) = \frac{180}{\pi} \arcsin(\frac{H_u}{d_m^k})$. The probability of an NLoS link occurring can be given by $P_{NLoS}(d_m^k) = 1 - P_{LoS}(d_m^k)$.

Meanwhile, the channel coefficient between SN m and the UAV hovering at CP k is denoted $h(d_m^k)$. It can be derived from $h(d_m^k) = \sqrt{\beta(d_m^k)}\lambda(d_m^k)$, where $\beta(d_m^k)$ reflects the large-scale propagation effects of the channel, such as path loss and shadowing, and $\lambda(d_m^k)$ reflects the small-scale propagation effects, including multi-path effects. Moreover, $\lambda(d_m^k)$ is usually a complex random variable with $\mathbb{E}[|\lambda(d_m^k)|^2] = 1$. The large-scale propagation model in [37,38] is applied as

$$\beta(d_m^k) = \begin{cases} \beta_0(d_m^k)^{-\alpha}, & \text{LoS link,} \\ \kappa\beta_0(d_m^k)^{-\alpha}, & \text{NLoS link,} \end{cases} \quad (4)$$

where β_0 is the path loss at the reference distance $d_0 = 1\text{m}$, α is the path loss exponent, and $\kappa < 1$ is the additional fading factor when the NLoS link is established. Therefore,

considering the randomness of LoS and NLoS link occurrences and the stochastic small-scale propagation effects, the average channel power gain is denoted

$$\begin{aligned}\mathbb{E}\left[|h(d_m^k)|^2\right] &= P_{LoS}(d_m^k)\beta_0(d_m^k)^{-\alpha} + P_{NLoS}(d_m^k)\kappa\beta_0(d_m^k)^{-\alpha} \\ &= \hat{P}_{LoS}(d_m^k)\beta_0(d_m^k)^{-\alpha}, \quad m \in \mathcal{M}, k \in \mathcal{K},\end{aligned}\quad (5)$$

where $\hat{P}_{LoS}(d_m^k) = P_{LoS}(d_m^k) + (1 - P_{LoS}(d_m^k))\kappa$ represents the regularized LoS link occurrence probability. Using the homogeneous approximation method described in [37], we can set $\hat{P}_{LoS}(d_m^k) \approx \bar{P}_{LoS}(d_m^k)$, where $\bar{P}_{LoS}(d_m^k)$ is taken as the probability value corresponding to the most likely elevation angle.

Furthermore, when the UAV hovers at CP k to collect the data of SN m , the data rate is denoted $R(d_m^k) = B \log_2 \left(1 + \frac{P_u |h(d_m^k)|^2}{\sigma^2 \Gamma}\right)$, where B represents the channel bandwidth, P_u denotes the transmit power of the UAV, σ^2 is the noise power at the BS, and Γ refers to the gap between the SNR required between the actual system and the theoretical model to achieve a specified performance, usually with $\Gamma > 1$. Then we can obtain an approximation of the mean data transmission rate from Jensen's inequality, expressed as $\mathbb{E}\left[R(d_m^k)\right] \approx B \log_2 \left(1 + \frac{P_u \mathbb{E}\left[|h(d_m^k)|^2\right]}{\sigma^2 \Gamma}\right)$. After that, it is assumed that the sensed data in SN m are compressed and packaged into data packets of defined size L_m . According to Equation (5) and the related description in this section, the data collection time required for SN m at CP k can be given by

$$\tau_m^k = \frac{L_m}{\mathbb{E}\left[R(d_m^k)\right]} = L_m / \left(B \log_2 \left(1 + \frac{P_u \bar{P}_{LoS}(d_m^k) \beta_0(d_m^k)^{-\alpha}}{\sigma^2 \Gamma}\right)\right), \quad m \in \mathcal{M}, k \in \mathcal{K}, \quad (6)$$

where $\mathbb{E}\left[R(d_m^k)\right]$ represents the mean data transmission rate.

The computational model needs to be discussed in two cases. When the energy is sufficient, since the UAV is equipped with an edge server, it acts as both the mobile data collector and the MEC server to further ensure information freshness in computationally intensive scenarios. Specifically, during the flight, the UAV can complete part of the computation task based on the collected data, the percentage of which is denoted P_{com} . Then the remaining part is finally offloaded to the MEC server s_0 for processing. When the energy is limited, to ensure that the data of all SNs are collected and transmitted, i.e., to complete the data collection task while reducing energy consumption, the UAV does not perform the computation task during the flight. Rather, the UAV offloads all data to the MEC server s_0 until the end of the flight.

2.3. AoI Model

At CP k , it is assumed that the UAV uses the TDMA scheme for data collection of $|\mathcal{A}_k|$ SNs. Let $\Delta_m^k(t)$ denote the instantaneous AoI of SN m at time $t (\geq 0)$. Assuming that SN m is associated with CP k , its AoI can be expressed as

$$\Delta_m^k(t) = \max\{0, t - t_m^k\}, \quad m \in \mathcal{M}, k \in \mathcal{K}, \quad (7)$$

where t_m^k denotes the instant when the data of SN m are generated. Note that the UAV completes only one round of data collection task, so the data at each SN are updated once during the period T in which the UAV performs the task. As a result, for any SN m , we are only interested in the AoI between its data generation and its reception by the monitoring center, i.e., $t_m^k \leq t \leq T$. Moreover, the communication link establishment time and the compression and sampling time are small enough to be ignored. Therefore, denoting the instant by T_u^k when the UAV reaches CP k , it is also equal to the data generation time of the first SN at CP k whose data are collected, $k \in \mathcal{K}$.

For illustrative purposes, let $\mathbf{q} = [(1), \dots, (k), \dots, (K)]$ denote the vector of the UAV's visiting sequences, where (k) gives the subscript of the k -th CP in the sequence and $(k) \in \mathcal{K}$. Moreover, assuming that k is the $\phi(k)$ -th CP visited by the UAV, then $\phi(k) \in \mathcal{K}$ and $k = (\phi(k))$. As shown in Figure 3, the interval between T_u^k and $T_u^{\phi(k)+1}$ can be divided into two segments, where T_c^k denotes the instant when the UAV completes the collection of data from all the associated SNs at CP k .

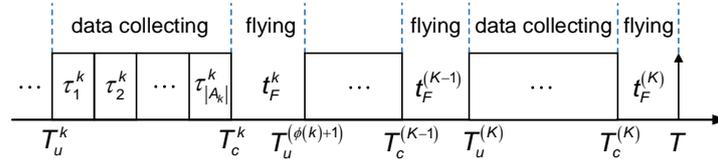


Figure 3. The timeline of the UAV accomplishing the data collection task.

The data collection time of the UAV can be derived from (6) as

$$t_c^k = \sum_{m=1}^M o_m^k \tau_m^k = \sum_{m=1}^M o_m^k \left[\frac{L_m}{B \log_2 \left(1 + \frac{P_u \bar{P}_{LoS} (d_m^k)^\beta \beta_0 (d_m^k)^{-\alpha}}{\sigma^2 \Gamma} \right)} \right], \quad k \in \mathcal{K}, m \in \mathcal{M}, \quad (8)$$

where o_m^k and τ_m^k , respectively, represent the association relationship and data collection time required for SN m at CP k .

The flight time of the UAV can be expressed as

$$t_F^k = \frac{D_c^k}{V} = \frac{\|\mathbf{u}_{\phi(k)+1} - \mathbf{u}_k\|}{V}, \quad k \in \mathcal{K}, \quad (9)$$

where D_c^k represents the distance between CP k and CP $(\phi(k) + 1)$. In particular, since the UAV eventually returns to the initial position \mathbf{w}_0 , it can be obtained that $\mathbf{u}_{(K+1)} = \mathbf{w}_0$.

Notice that the data at each SN are continuously getting older since they are generated. During the period T , the peak AoI of all SNs can be denoted

$$AoI_p = \Delta^{(1)}(T) = T - T_u^{(1)} = \sum_{k=1}^K (t_c^k + t_F^k), \quad k \in \mathcal{K}, \quad (10)$$

where $\Delta^{(1)}(t)$ represents the AoI of the first SN whose data are collected at CP (1).

3. Problem Formulation

In this section, we present a trajectory planning problem in a UAV-assisted information freshness-oriented MEC network. The peak AoI is minimized by jointly optimizing the locations of CPs, SN-CP associations, as well as the flight trajectory of the UAV. Specifically, the trajectory of the UAV consists of the hover points for data collection $\mathbf{L} = \{\mathbf{u}_k | k \in \mathcal{K}\}$ and the visiting sequence \mathbf{q} . Let \mathcal{F} represent a set of $K!$ (the factorial of K) permutations for the visiting sequences of K CPs. Then we have

$$\mathbf{q} = [(1), \dots, (k), \dots, (K)] \in \mathcal{F}. \quad (11)$$

Analyzing (10), it is clear that AoI_p is strongly correlated with K . Furthermore, (8) shows that t_c^k is only related to the SN-CP association relationship \mathbf{o} and the distance between CP k and the associated SNs. In addition, from (9), it can be found that AoI_p is closely related to the trajectory of the UAV. Therefore, we formulate the optimization problem as

$$(\mathbf{P0}) : \min_{K, \mathbf{L}, \mathbf{q}, \mathbf{o}} \sum_{k=1}^K (t_c^k + t_F^k) \quad (12)$$

s.t. (1)–(3), and (11).

Denote the optimal solution of (P0) as $(K^*, \mathbf{L}^*, \mathbf{q}^*, \mathbf{o}^*)$, corresponding to the optimal strategy. Note that the problem (P0) is a very difficult mixed-integer non-convex optimization problem to solve because the variables \mathbf{L} and \mathbf{q} have dimensions that vary with the integer variable K ; (2), (3), and (11) are non-convex discrete constraints, and the variables \mathbf{L} , \mathbf{q} , and \mathbf{o} are coupled.

4. Proposed Algorithm

In this section, we elaborate on how to solve this optimization problem accurately and efficiently. Since the objective function in (12) is a weighted sum of the data collection time and the UAV's flight time, (P0) can be decomposed into two sub-problems: the CP location and SN-CP association problem and the UAV trajectory planning problem. Then these two sub-problems are discussed separately to give feasible algorithms. Finally, an iterative two-step algorithm called the AoI-minimized association and trajectory planning (AoI-MATP) algorithm is designed to find the optimal solution to the proposed problem (P0).

4.1. CP Location and SN-CP Association Optimization

Firstly, the SN-CP association problem can be described as

$$(\mathbf{P1}) : \min_{K, \mathbf{L}, \mathbf{o}} \sum_{k=1}^K \sum_{m=1}^M o_m^k \left[\frac{L_m}{B \log_2 \left(1 + \frac{P_u \bar{P}_{LoS}(d_m^k) \beta_0 (d_m^k)^{-\alpha}}{\sigma^2 \Gamma} \right)} \right] \quad (13)$$

s.t. (1)–(3).

According to (13), since the objective function and constraint (1) are both nonlinear, the problem (P1) is a nonlinear programming problem. Furthermore, due to the binary indicator o_m^k , both (2) and (3) are discrete constraints, so Problem (P1) is a very challenging mixed-integer non-convex optimization problem for which there is no generalized algorithm to find an optimal solution. To decrease the complexity of solving this problem, we first choose a suitable clustering algorithm to find the optimal value of K and the corresponding SN-CP association. In this context, the binary indicator $o_m^k = 1$ indicates that the data of SN m are collected by the UAV at the CP directly above SN k and τ_m^k denotes the required data collection time; otherwise, $o_m^k = 0$. In addition, constraint (2) ensures that each SN can only be associated with one SN in its neighborhood, where the set \mathcal{N}_m is used to denote the neighboring nodes of the SN m , defined as $\mathcal{N}_m = \{k \mid \|\mathbf{w}_m - \mathbf{w}_k\| \leq r_u, k \in \mathcal{M}\}$. Therefore, the problem to be solved transforms into

$$(\mathbf{P2}) : \min_{\mathbf{o}} \sum_{k=1}^M \sum_{m=1}^M o_m^k \tau_m^k$$

s.t. $\sum_{k \in \mathcal{N}(m)} o_m^k = 1, \forall m,$

$$\sum_{k=1}^M \max_{m \in \mathcal{N}(k)} o_m^k = K,$$

$$o_m^k \in \{0, 1\}, \forall m, k. \quad (14)$$

For a given K , it is easy to realize that (P2) is an integer linear programming (ILP) problem that is NP-hard. We propose an algorithm named AP clustering-based collection time optimization algorithm to obtain a suboptimal solution. The algorithm is insensitive to the initial positions of the data points and considers all SNs as candidate exemplars (i.e., cluster centers). Moreover, its input consists of a similarity matrix, and the similarity of SN k as the exemplar of SN m is denoted $s(m, k)$, which is given by

$$s(m, k) = \begin{cases} -\tau_m^k, & m \neq k, m, k \in \mathcal{M}, \\ -\varepsilon, & m = k, m, k \in \mathcal{M}, \end{cases} \quad (15)$$

where ε denotes a positive clustering parameter that determines the size of K .

The algorithm contains two main iterative messages: (1) Responsibility $r(m, k)$ is passed from SN m to the candidate exemplar SN k , reflecting the degree to which SN k is suitable to serve as the exemplar of SN m . (2) Availability $a(m, k)$ is passed from the candidate exemplar SN k to SN m , which reflects the suitability of SN m to select SN k as its exemplar, taking into account the support of other SNs for SN k as the exemplar. The AP algorithm iteratively updates the above two messages as

$$r_{t+1}(m, k) = s(m, k) - \max_{k' \in \mathcal{N}(m) \setminus \{k\}} \{a_t(m, k') + s(m, k')\}, \quad (16)$$

$$a_{t+1}(m, k) = \begin{cases} \min\{0, r_t(k, k) + \sum_{m' \in \mathcal{N}(k) \setminus \{m, k\}} \max\{0, r_t(m', k)\}\}, & m \neq k, \\ \sum_{m' \in \mathcal{N}(k) \setminus \{m, k\}} \max\{0, r_t(m', k)\}, & m = k, \end{cases} \quad (17)$$

where $r_t(\cdot)$ and $a_t(\cdot)$, respectively, represent the responsibility and availability in the t -th iteration, k' represents one of the neighboring nodes of SN m other than k , and m' represents one of the neighboring nodes of SN k other than m and k .

The algorithm converges until the sum of similarities of all the data points to their nearest class-representative points is maximized. At that moment, a stable set of exemplars can be obtained, whose indexes are denoted by

$$\mathcal{C} = \{k | r(k, k) + a(k, k) > 0\}, \quad (18)$$

so the SNs are divided into $K = |\mathcal{C}|$ clusters. For any $m \in \mathcal{M}$, the index of its exemplar can be given by $k = \arg \max s(m, k)$, $k \in \mathcal{C}$, whereby all SN-CP association variables \mathbf{o} are easily obtained. Based on this, to find the specific locations of the K CPs, for each cluster, the problem of determining the CP location is modeled as a single-facility siting problem and resolved by a simple iterative process of the center of gravity method:

$$\mathbf{u}_k = \operatorname{argmin} \sum_{m=1}^M o_m^k \tau_m^k, \quad k \in \mathcal{C}. \quad (19)$$

The complete procedures are shown in Algorithm 1. Since the algorithm runs for at most t_{max} rounds before convergence, assuming that the computation of (16) and (17) is performed sequentially in each round, the worst-case time complexity is $O(M^3 t_{max})$, which is polynomial complexity about M . In addition, the space complexity of the algorithm is $O(M^2 t_{max})$, since the storage space for the variables can be reused in each round.

Algorithm 1 AP-based CP location and SN-CP association optimization algorithm

- 1: Initialize $\mathbf{w}_m, \varepsilon, \tau_m^k, t_{max}, r(m, k) = a(m, k) = 0$, and $t = 0$.
- 2: Calculate τ_m^k between any two SNs, and set $s(m, k)$ according to (15).
- 3: **repeat**
- 4: Update $r(m, k), a(m, k)$ according to (16) and (17), respectively.
- 5: Update $t \leftarrow t + 1$.
- 6: **until** Convergence or $t \geq t_{max}$.
- 7: Find \mathcal{C} by (18) and establish K and \mathbf{o} .
- 8: Find \mathbf{L} by (19) according to the center of gravity method.

Remark 1. For the proposed Algorithm 1, the advantages are as follows. First, it does not need to specify the number of final clusters and different clustering results can be obtained by adjusting the preference $-\varepsilon$. Moreover, the algorithm is insensitive to the initial value of the data and the obtained clustering results are more stable. Furthermore, compared to other clustering algorithms, such as k -centers clustering, the squared error of its results is smaller.

4.2. UAV Trajectory Optimization

Based on the cluster center locations obtained from Algorithm 1, the trajectory planning problem for a UAV is formulated as

$$\begin{aligned}
 (\mathbf{P3}) : \quad & \min_{\mathbf{q}} \sum_{k=1}^K \frac{\|\mathbf{u}_{(\phi(k)+1)} - \mathbf{u}_k\|}{V} \\
 & \text{s.t. (11)}.
 \end{aligned} \tag{20}$$

Noting that the objective function and constraints of (P3) are discrete and nonlinear, it is a non-convex optimization problem hard to find a globally optimal solution. Fortunately, it is not difficult to find that it belongs to the combinatorial optimization problem, particularly a variation of the traveling salesman problem (TSP). Thus, there is no generalized polynomial time solution for (P3). To avoid the surge in algorithmic complexity due to the excessive problem size, we propose an elite genetic algorithm (eGA)-based algorithm to find a suboptimal solution to this problem.

The algorithm uses a sequence coding approach, where the visiting sequence of a UAV \mathbf{q} is viewed as a chromosome, and each value in \mathbf{q} represents a different gene. The objective function in (P3) can be used to construct the fitness function of the algorithm

$$f(\mathbf{q}) = \frac{1}{\sum_{k=1}^K \frac{\|\mathbf{u}_{(\phi(k)+1)} - \mathbf{u}_k\|}{V}} = \frac{V}{\sum_{k=1}^K \|\mathbf{u}_{(\phi(k)+1)} - \mathbf{u}_k\|}, \quad \mathbf{q} \in \mathcal{F}. \tag{21}$$

The function in (21) is used to evaluate the degree of excellence of an individual: the higher the fitness, the higher the probability that it contributes to finding the optimal solution. Following the principle of survival of the fittest in evolutionary theory, we use the elitist-preserving strategy to select the resulting breeding offspring. More specifically, the most adaptable individual in the current population is retained as an elitist, and a binary tournament selection operator is used for mating selection. After performing the ordered crossover and shuffle indexes mutation operations, if the previously retained elitist of the parent is better than the most adapted individual in the current population, it will be used to replace the worst individual in the current population. To improve the algorithm's ability to search for local optima, the 2-OPT algorithm is used to optimize the elites in the current population. Then N_p optimal individuals in the current population are selected for the next round of evolution. The specific steps are described in Algorithm 2.

Algorithm 2 eGA-based UAV trajectory optimization algorithm

- 1: Initialize flight time t_F with L , eGA-related parameters $N_p, N_g, P_c, P_m, P_{opt}$.
- 2: Generate the initial population with N_p chromosomes.
- 3: Set the index of generation $n = 0$.
- 4: **repeat**
- 5: Update the fitness values of N_p chromosomes according to (21) and select the best chromosome remaining.
- 6: Select the parent chromosomes with the binary tournament operator for mating.
- 7: Perform the ordered crossover operations with probability P_c and shuffle indexes mutation operations with probability P_m on parent chromosomes.
- 8: Perform the 2-OPT algorithm for local optimization with probability P_{opt} .
- 9: Update the fitness values of new chromosomes and generate the new population with N_p chromosomes according to the elitist-preserving strategy.
- 10: Update $n \leftarrow n + 1$.
- 11: **until** $n \geq N_g$.
- 12: Select the best chromosome in the final generation as the best visiting sequence, i.e., $\mathbf{q} = \arg \max f(\mathbf{q})$.

Analyzing Algorithm 2, its time complexity mainly depends on lines 7–8, which is approximately $O(N_g N_p K)$ at worst, and the space complexity is $O(N_p K)$.

4.3. AoI-MATP Algorithm

Based on Algorithms 1 and 2, we propose an AoI-MATP algorithm to find the optimal solution to the problem (P0) approximately.

The algorithm is an iterative process, where each round corresponds to a different ϵ , leading to varying K . The optimal collection time and the corresponding optimal flight time can be obtained by executing Algorithms 1 and 2 alternately. When the algorithm converges, the optimal solution $(K^*, \mathbf{L}^*, \mathbf{q}^*, \mathbf{o}^*)$ can be determined based on the ϵ corresponding to the minimum AoI_p . The detailed procedures of the algorithm are described in Algorithm 3. It is well-understood that the complexity of Algorithm 3 is determined by Algorithms 1 and 2, which it contains.

Algorithm 3 AoI-MATP algorithm

- 1: Initialize w_m , a small positive constant $\Delta\epsilon$, and $\epsilon = \epsilon_0$.
- 2: Calculate collection time τ_m^k between any two SNs.
- 3: **repeat**
- 4: Run Algorithm 1 to obtain K , SN-CP association \mathbf{o} , and locations of CPs L with ϵ .
- 5: Calculate the optimal collection time t_c^k in each CP k by (8) and flight time t_F between any two CPs according to their distances.
- 6: Run Algorithm 2 to obtain the visiting sequence \mathbf{q} .
- 7: Calculate the optimal peak AoI by (9) and (10).
- 8: Update $\epsilon \leftarrow \epsilon - \Delta\epsilon$.
- 9: **until** $\epsilon < 0$.
- 10: Find the minimum AoI_p^* with the optimal ϵ^* , and the according optimal solutions $(K^*, \mathbf{L}^*, \mathbf{q}^*, \mathbf{o}^*)$ are all found.

5. Simulation Results and Discussion

To evaluate the performance of the proposed algorithm, we present and analyze the simulation results in this section. The simulations were executed on a desktop computer equipped with an Intel Core i7-12700 CPU@2.10 GHz, based on Python 3.8. The manufacturer of the device is HP, sourced from Beijing, China. We consider M SNs randomly distributed in a square area with a side length of 2000 m, and the departure time of the UAV from the MEC server s_0 is $t = 0$. If not specified, the system parameters are set as

follows: $H_u = 50$ m, $V = 20$ m/s, $r_u = 1000$ m, $(a, b) = (9.61, 0.16)$, $\beta_0 = -60$ dB, $\alpha = 2.2$, $\kappa = 0.2$, $B = 1$ MHz, $P_u = 20$ dBm, $\sigma^2 = -110$ dBm, $\Gamma = 8.2$ dB.

Running our proposed Algorithm 3, the convergence curve of the GA algorithm is shown in Figure 4 when K reaches a maximum value of 50. It can be seen that the flight time shows an overall significant decrease with the number of iterations. Convergence is reached when the algorithm has been iterated about 68 times. Figure 5a shows how the number of CPs K changes regarding the parameter ε for $M = 50$. Experimentally, it is found that different ε may lead to the same K . To demonstrate the results more clearly, we choose the ε that makes the peak AoI the smallest for each K . It is obvious from Figure 5a that K tends to decrease as ε increases. According to (15), this is due to the fact that $-\varepsilon$ represents the likelihood of each SN to be an exemplar, and the larger $-\varepsilon$ is, the more SNs are likely to be exemplars. Similarly, the larger ε is, the fewer SNs are likely to be exemplars, i.e., the smaller K is.

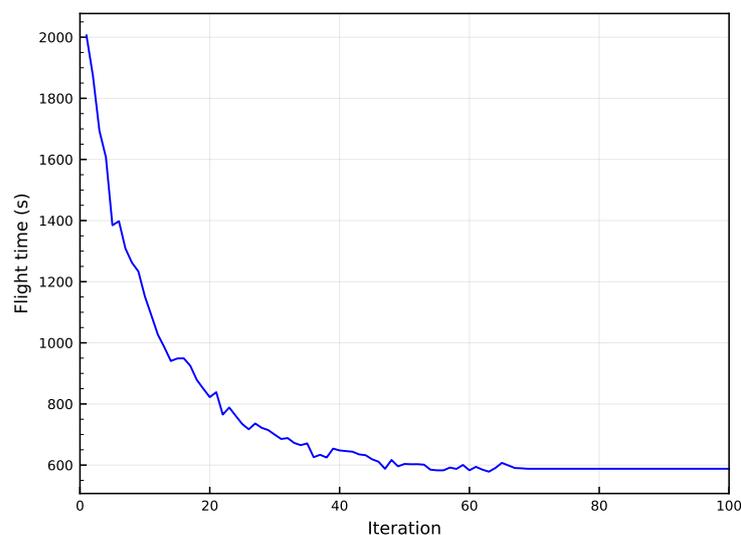


Figure 4. The convergence curve of elite genetic algorithm (eGA) algorithm when $K = 50$.

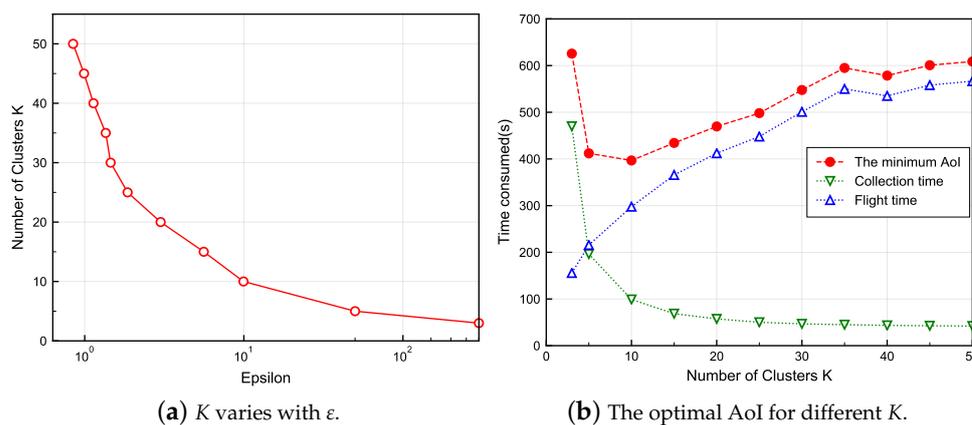


Figure 5. Clustering results and corresponding optimal age of information (AoI) obtained by Algorithm 3 when $M = 50$.

Figure 5b demonstrates the optimal peak AoI and the relative collection and flight times for different K . The specific simulation results when $M = 50$ are listed in Table 1. In particular, the parameters in Algorithm 2 are set as $N_p = 300$, $N_g = 300$, $P_c = 0.99$, $P_m = 0.15$, $P_{opt} = 0.1$. First, it can be found from both subfigures of Figure 5 that K does not start at 1 but at 3. The reason is that the clustering result with $K < 3$ cannot satisfy the distance limit in (1). Second, it can be observed that the collection time decreases with increasing K , while the flight time increases with it. In addition, the collection time

decreases fast and dominates at the beginning, but the rate of decrease becomes flat in the latter half, while the flight time changes more uniformly, which leads to a trend of decreasing and then increasing peak AoI. Analyzing the reason, when K is larger, there are fewer SNs associated at each CP, the distances between SNs and CPs are shorter, and the collection time decreases consequently. In contrast, the UAV's flight trajectory contains more CPs, increasing its flight distance and yielding a longer flight time. In particular, it is noted that the peak AoI for the $K = 50$ case reaches 608.76 s. This case implies that the data of each SN are collected directly above itself, and the UAV needs to fly over each SN one by one for data collection. Therefore, the poorer peak AoI is due to the longer flight time caused by the fact that clustering was not performed. From the above analysis, it is easy to conclude that our proposed clustering idea can well strike a balance between the collection time and the flight time, so the optimal peak AoI can be achieved by choosing a suitable K .

Table 1. Simulation results when $M = 50$.

Metric	$K = 3$	$K = 10$	$K = 20$	$K = 30$	$K = 40$	$K = 50$
Minimum Peak AoI (s)	625.70	396.77	469.65	547.69	578.53	608.76
Collection Time (s)	469.96	98.98	57.38	46.78	43.51	42.11
Flight Time (s)	155.74	297.79	412.27	500.92	535.02	566.65

The optimal and suboptimal clustering results and the flight trajectories are respectively visualized in Figure 6a,b. Note that the different colored circles in the figure represent SNs in different clusters. The specific simulation results are as follows. When $K = 6$, the optimal peak AoI is 393.20 s, corresponding to a collection time of 156.27 s and a flight time of 236.93 s. When $K = 10$, the suboptimal peak AoI is 396.77 s, corresponding to a collection time of 98.98 s and a flight time of 297.79 s.

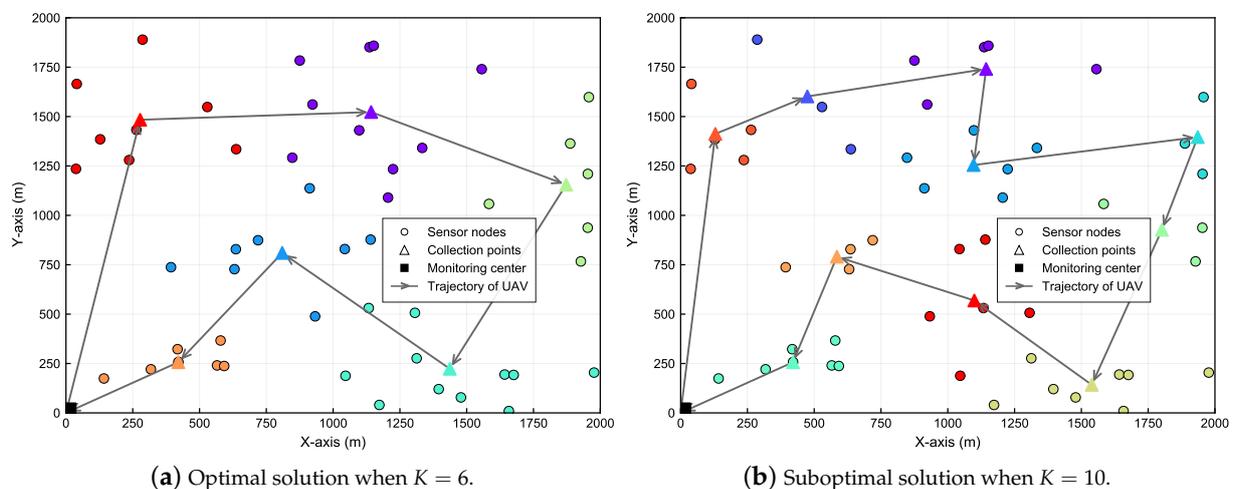


Figure 6. Optimal and suboptimal clustering results and trajectories when $M = 50$.

Further, to better evaluate the performance of the proposed Algorithm 3, the following four schemes are considered as comparisons:

1. Solve the ILP problem (P2) using the branch-and-cut (BC) method and find the optimal trajectory using Algorithm 2.
2. Solve the problem (P2) using Algorithm 1 and find the optimal trajectory using the dynamic programming (DP) algorithm.
3. The first step is the same as in Scheme 2, and then solve the TSP to obtain the optimal trajectory.
4. The first step is the same as above, and the optimal trajectory is solved using the greedy algorithm.

Considering that the complexity of the DP algorithm can be up to $O(2^M)$, M is set to 20 to ensure that the experiment can be carried out properly. Moreover, the parameters in Algorithm 2 are set as $N_p = 300$, $N_g = 500$, $P_c = 0.9$, $P_m = 0.2$, $P_{opt} = 0.1$. Running Algorithm 3, the convergence curve of the GA algorithm is shown in Figure 7 when K reaches a maximum value of 20. It can be seen that the flight time shows an overall significant decrease with the number of iterations. Convergence is reached when the algorithm has been iterated about ten times, less than the number of iterations at $K = 50$. Figures 4 and 7 illustrate that our proposed algorithm can solve the problem efficiently with a small number of CPs and also obtains an asymptotically optimal solution with an acceptable number of iterations when the number of CPs is large.

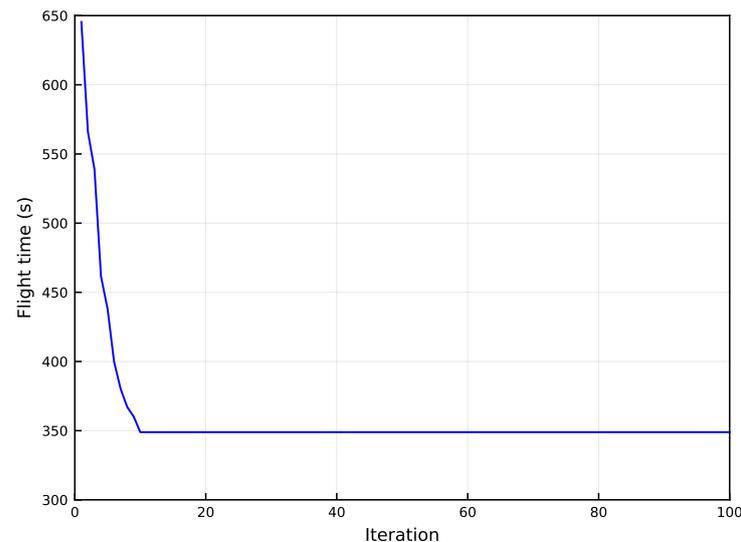


Figure 7. The convergence curve of Algorithm 2 when $K = 20$.

The five AoI curves derived from all the schemes are shown in Figure 8 and the specific simulation results are listed in Table 2. It is easily observed that the five curves in the figure have the same trend with K . First, there is a small decrease until the optimum is reached, then there is a general upward trend, and all of them reach the worst peak AoI at $K = 20$ except Scheme 4. Slightly different from the AoI curves in Figure 6b, except for Scheme 3 which achieves the optimal peak AoI at $K = 5$, the other three schemes as well as our proposed algorithm all achieve the optimal solution at $K = 4$, and the optimal peak AoI becomes smaller. Understandably, this is because the reduction in M leads to the association of fewer SNs at each CP as well. This further results in a lower percentage of data collection time, and the flight time dominates in this case. It can be noticed that Scheme 4 provides the worst peak AoI performance in all cases. This is due to it only considering the nearest unvisited CPs at each decision step and not being motivated by a global perspective, causing it to obtain usually locally optimal solutions.

Table 2. Simulation results when $M = 20$.

Scheme	K = 3	K = 4	K = 5	K = 10	K = 15	K = 20
Proposed Algorithm	278.50	239.38	246.14	296.37	345.66	365.70
Comparison Scheme 1	278.50	239.38	246.14	301.09	348.14	365.70
Comparison Scheme 2	278.50	239.38	246.14	296.37	345.66	365.70
Comparison Scheme 3	294.86	270.64	265.29	322.77	347.22	396.19
Comparison Scheme 4	377.02	349.27	359.28	367.77	441.64	531.04

On the contrary, Schemes 1 and 2 and the proposed algorithm demonstrate the best performance, and the three curves have identical results for $K < 9$ as well as $K > 16$. Comparing Scheme 1 and the proposed algorithm, since BC is an exact algorithm for

solving the ILP problem, the clustering results it obtains must be optimal. However, its peak AoI is not always smaller than the proposed algorithm, such as when $K = 11$ and 14 . This is mainly related to the fact that peak AoI includes both collection time and flight time, and the optimal collection time does not necessarily correspond to the optimal flight time. The comparison of Scheme 2 and the proposed algorithm shows that Scheme 2 has better performance at some points with the same clustering results. This is because the DP algorithm included in Scheme 2 obtains the optimal solution by comparing all the candidate paths. Therefore, DP has an exponential algorithm complexity that increases rapidly as the problem size grows. The space required by the algorithm exceeds the memory capacity of the computer when $M > 20$. However, GA belongs to heuristic algorithms; it usually obtains the optimal solution or suboptimal solution but has a better computational complexity than DP. The proposed algorithm can achieve a performance extremely close to the exact algorithm near the optimal number of clusters or when K is very large. Overall, Scheme 2 does not apply to the large-scale SNs' data collection problem, while our proposed algorithm is still able to solve with high accuracy by adjusting the parameters.

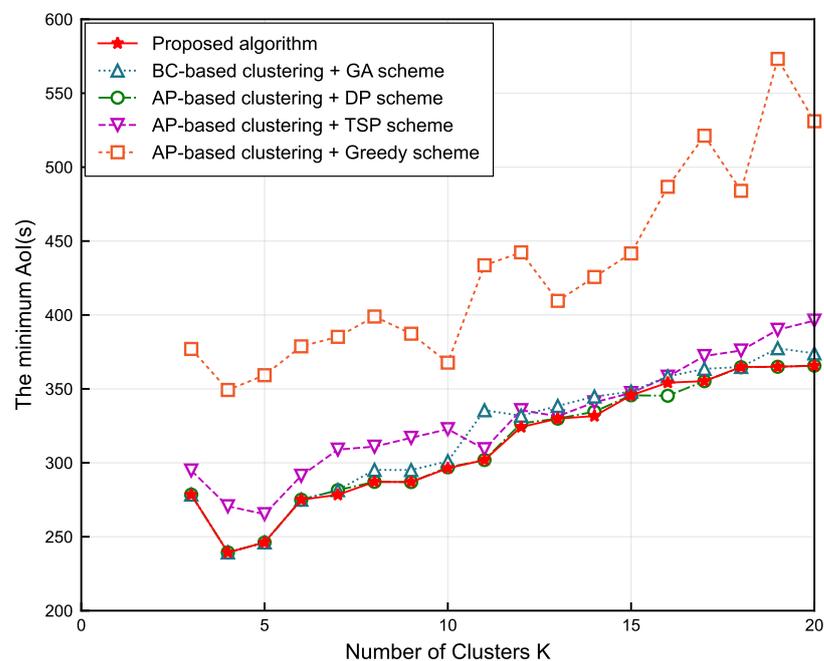


Figure 8. The minimum peak AoI versus K when $M = 20$.

Lastly, the performance of Scheme 3 is substantially better than that of Scheme 4, but it still has some gaps with the optimal performance. This can be explained because the TSP always considers the optimal solution of a loop that traverses all CPs from the starting point and back to the starting point. Even if we have subtracted the distance from the starting point to the first visited CP, the result obtained is not the path that makes the peak AoI optimal.

As presented in Table 3, we also evaluated the running time of the proposed algorithm. Combined with the results in Figure 8, it is easy to notice that Schemes 1, 3, and 4 obtain relatively poor solutions in a faster time. The remaining two better-performing schemes vary greatly in their running time. Consistent with our theoretical analysis, Scheme 2 obtains the optimal result with the longest running time. When $M = 20$, the proposed algorithm obtains very similar results to Scheme 2 in a much shorter time. When $M > 20$, the computational complexity of Scheme 2, which contains the DP algorithm, is too high for the simulation to proceed. We give the running times of the remaining schemes for $M = 30$. The four schemes spend more time on computations when M is increased, but it is still acceptable.

Table 3. The execution time between the compared algorithms.

	Proposed	Scheme 1	Scheme 2	Scheme 3	Scheme 4
M = 20	78.06 s	17.24 s	569.62 s	118.46 s	61.94 s
M = 30	411.15 s	98.96 s	/	354.62 s	158.75 s

To better explain the generality of our algorithm, the main simulation results of the four schemes for $M = 30$ are presented in Table 4. The minimum AoI for all schemes is obtained at $K = 5$, and the first three schemes have the same result of 336.25. Only Scheme 4 has a worse result of 413.47. From the simulation results, we can conclude that our algorithm can still give a better feasible solution in an acceptable time at $M = 30$. In agreement with our theoretical analysis of the complexity of the GA algorithm, this is because K is very small compared to N_p and N_g , so our algorithm is equally applicable to larger problem sizes. Asymptotically optimal solutions can be obtained by adjusting the size of the hyperparameters in the proposed algorithm if high-precision results are required and computational resources are sufficient.

Table 4. Simulation results when $M = 30$.

Scheme	K = 4	K = 5	K = 6	K = 10	K = 20	K = 30
Proposed Algorithm	356.38	336.25	346.33	387.07	442.97	495.97
Comparison Scheme 1	356.38	336.25	348.48	400.54	466.49	577.23
Comparison Scheme 3	356.38	336.25	346.33	387.98	445.37	683.11
Comparison Scheme 4	436.47	413.47	471.17	534.57	535.50	625.52

6. Conclusions

In this paper, we mainly consider a UAV-assisted MEC network in which the peak AoI of the collected data is minimized by jointly optimizing the SN-CP associations, location of the data CPs, and UAV trajectory. To solve this complex mixed-integer non-convex optimization problem, we propose an iterative algorithm to first obtain the optimal SN-CP associations and locations of CPs, based on which the optimal UAV trajectory is found. Simulation results show that our proposed algorithm can achieve an excellent balance between the data collection time and the flight time of the UAV and has a superior performance close to several exact algorithms. In future work, we will consider the case where the UAV acts as the MEC server in computation-intensive tasks, e.g., by optimizing the allocation of communication resources to reduce the AoI and energy consumption.

Author Contributions: Conceptualization, H.S. and Y.J.; methodology, H.S. and Y.J.; software, H.S.; validation, H.S., D.W. and Y.J.; formal analysis, H.S. and Y.J.; investigation, Z.H.; resources, H.S., D.W. and Y.J.; writing—original draft preparation, H.S. and Z.H.; writing—review and editing, H.S., D.W. and Y.J.; visualization, H.S. and D.W.; supervision, Y.J.; project administration, H.S. and Y.J. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are contained within the article.

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

References

1. Feng, D.; Lai, L.; Luo, J.; Zhong, Y.; Zheng, C.; Ying, K. Ultra-Reliable and Low-Latency Communications: Applications, Opportunities and Challenges. *Sci. China Inf. Sci.* **2021**, *64*, 60–71. [[CrossRef](#)]
2. Cui, Q.; Wang, Y.; Chen, K.C.; Ni, W.; Lin, I.C.; Tao, X.; Zhang, P. Big Data Analytics and Network Calculus Enabling Intelligent Management of Autonomous Vehicles in a Smart City. *IEEE Internet Things J.* **2019**, *6*, 2021–2034. [[CrossRef](#)]
3. Shu, T.; Chen, J.; Bhargava, V.K.; de Silva, C.W. An Energy-Efficient Dual Prediction Scheme Using LMS Filter and LSTM in Wireless Sensor Networks for Environment Monitoring. *IEEE Internet Things J.* **2019**, *6*, 6736–6747. [[CrossRef](#)]
4. Kassim, M.R.M. IoT Applications in Smart Agriculture: Issues and Challenges. In Proceedings of the 2020 IEEE Conference on Open Systems (ICOS), Kota Kinabalu, Malaysia, 17–19 November 2020; pp. 19–24. [[CrossRef](#)]
5. Li, J.; Han, Q.; Wang, W. Characteristics Analysis and Suppression Strategy of Energy Hole in Wireless Sensor Networks. *Ad Hoc Netw.* **2022**, *135*, 102938. [[CrossRef](#)]
6. Jiao, Z.; Zhang, Y.; Xin, J.; Mu, L.; Yi, Y.; Liu, H.; Liu, D. A Deep Learning Based Forest Fire Detection Approach Using UAV and YOLOv3. In Proceedings of the 2019 1st International Conference on Industrial Artificial Intelligence (IAI), Shenyang, China, 23–27 July 2019; pp. 1–5. [[CrossRef](#)]
7. Nugroho, G.; Satrio, M.; Rafsanjani, A.A.; Trio Sadewo, R.R. Avionic System Design Unmanned Aerial Vehicle for Disaster Area Monitoring. In Proceedings of the 2015 International Conference on Advanced Mechatronics, Intelligent Manufacture, and Industrial Automation (ICAMIMIA), Surabaya, Indonesia, 15–17 October 2015; pp. 198–201. [[CrossRef](#)]
8. Ma, Y.; Sun, H.; Li, S.; Wang, X.; Quek, T.Q.S. Performance Analysis of IoT Networks with Mobile Data Collectors. In Proceedings of the 2021 IEEE Wireless Communications and Networking Conference (WCNC), Nanjing, China, 29 March–1 April 2021; pp. 1–7. [[CrossRef](#)]
9. Lyu, M.; Zhao, Y.; Huang, C.; Huang, H. Unmanned Aerial Vehicles for Search and Rescue: A Survey. *Remote Sens.* **2023**, *15*, 3266. [[CrossRef](#)]
10. Al-Qubaydhi, N.; Alenezi, A.; Alanazi, T.; Senyor, A.; Alanezi, N.; Alotaibi, B.; Alotaibi, M.; Razaque, A.; Abdelhamid, A.A.; Alotaibi, A. Detection of Unauthorized Unmanned Aerial Vehicles Using YOLOv5 and Transfer Learning. *Electronics* **2022**, *11*, 2669. [[CrossRef](#)]
11. Li, M.; Li, H.; Ma, P.; Wang, H. Energy Maximization for Ground Nodes in UAV-Enabled Wireless Power Transfer Systems. *IEEE Internet Things J.* **2023**, *10*, 17096–17109. [[CrossRef](#)]
12. Baek, J.; Han, S.I.; Han, Y. Optimal UAV Route in Wireless Charging Sensor Networks. *IEEE Internet Things J.* **2020**, *7*, 1327–1335. [[CrossRef](#)]
13. Zhan, C.; Zeng, Y. Completion Time Minimization for Multi-UAV-Enabled Data Collection. *IEEE Trans. Wirel. Commun.* **2019**, *18*, 4859–4872. [[CrossRef](#)]
14. Zhu, K.; Xu, X.; Han, S. Energy-Efficient UAV Trajectory Planning for Data Collection and Computation in mMTC Networks. In Proceedings of the 2018 IEEE Globecom Workshops (GC Wkshps), Abu Dhabi, United Arab Emirates, 9–13 December 2018; pp. 1–6. [[CrossRef](#)]
15. Baek, J.; Han, S.I.; Han, Y. Energy-Efficient UAV Routing for Wireless Sensor Networks. *IEEE Trans. Veh. Technol.* **2020**, *69*, 1741–1750. [[CrossRef](#)]
16. Xu, X.; Zhao, H.; Yao, H.; Wang, S. A Blockchain-Enabled Energy-Efficient Data Collection System for UAV-Assisted IoT. *IEEE Internet Things J.* **2021**, *8*, 2431–2443. [[CrossRef](#)]
17. Zong, J.; Shen, C.; Cheng, J.; Gong, J.; Chang, T.H.; Chen, L.; Ai, B. Flight Time Minimization via UAV's Trajectory Design for Ground Sensor Data Collection. In Proceedings of the 2019 16th International Symposium on Wireless Communication Systems (ISWCS), Oulu, Finland, 27–30 August 2019; pp. 255–259. [[CrossRef](#)]
18. Luo, W.; Shen, Y.; Yang, B.; Wang, S.; Guan, X. Joint 3-D Trajectory and Resource Optimization in Multi-UAV-Enabled IoT Networks with Wireless Power Transfer. *IEEE Internet Things J.* **2021**, *8*, 7833–7848. [[CrossRef](#)]
19. Kaul, S.; Yates, R.; Gruteser, M. Real-Time Status: How Often Should One Update? In Proceedings of the 2012 IEEE INFOCOM, Orlando, FL, USA, 25–30 March 2012; pp. 2731–2735. [[CrossRef](#)]
20. Alfattani, S.; Jaafar, W.; Yanikomeroğlu, H.; Yongacoglu, A. Multi-UAV Data Collection Framework for Wireless Sensor Networks. In Proceedings of the 2019 IEEE Global Communications Conference (GLOBECOM), Waikoloa, HI, USA, 9–13 December 2019; pp. 1–6. [[CrossRef](#)]
21. Ghorbel, M.B.; Rodríguez-Duarte, D.; Ghazzai, H.; Hossain, M.J.; Menouar, H. Joint Position and Travel Path Optimization for Energy Efficient Wireless Data Gathering Using Unmanned Aerial Vehicles. *IEEE Trans. Veh. Technol.* **2019**, *68*, 2165–2175. [[CrossRef](#)]
22. Chen, Y.; Shen, S. A UAV-based Data Collection Approach for Wireless Sensor Network. In Proceedings of the 2020 7th International Conference on Information Science and Control Engineering (ICISCE), Changsha, China, 18–20 December 2020; pp. 2298–2302. [[CrossRef](#)]
23. Ebrahimi, D.; Sharafeddine, S.; Ho, P.H.; Assi, C. Data Collection in Wireless Sensor Networks Using UAV and Compressive Data Gathering. In Proceedings of the 2018 IEEE Global Communications Conference (GLOBECOM), Abu Dhabi, United Arab Emirates, 9–13 December 2018; pp. 1–7. [[CrossRef](#)]

24. Tong, P.; Liu, J.; Wang, X.; Bai, B.; Dai, H. UAV-Enabled Age-Optimal Data Collection in Wireless Sensor Networks. In Proceedings of the 2019 IEEE International Conference on Communications Workshops (ICC Workshops), Shanghai, China, 20–24 May 2019; pp. 1–6. [[CrossRef](#)]
25. Huang, X.; Fu, X. Fresh Data Collection for UAV-Assisted IoTs Based on Aerial Collaborative Relay. *IEEE Sens. J.* **2023**, *23*, 8810–8825. [[CrossRef](#)]
26. Jia, Z.; Qin, X.; Wang, Z.; Liu, B. Age-Based Path Planning and Data Acquisition in UAV-Assisted IoT Networks. In Proceedings of the 2019 IEEE International Conference on Communications Workshops (ICC Workshops), Shanghai, China, 20–24 May 2019; pp. 1–6. [[CrossRef](#)]
27. Abd-Elmagid, M.A.; Dhillon, H.S. Average Peak Age-of-Information Minimization in UAV-Assisted IoT Networks. *IEEE Trans. Veh. Technol.* **2019**, *68*, 2003–2008. [[CrossRef](#)]
28. Zhou, Z.; Liu, J.; Mao, C. Age of Information Oriented Data Collection via Energy-Constrained UAVs in Wireless Sensor Networks. *IEEE Access* **2024**, *12*, 11897–11908. [[CrossRef](#)]
29. Zhang, X.; Chang, Z.; Hämmäläinen, T.; Min, G. AoI-Energy Tradeoff for Data Collection in UAV-Assisted Wireless Networks. *IEEE Trans. Commun.* **2023**, *1*. [[CrossRef](#)]
30. Gao, X.; Zhu, X.; Zhai, L. AoI-Sensitive Data Collection in Multi-UAV-Assisted Wireless Sensor Networks. *IEEE Trans. Wirel. Commun.* **2023**, *22*, 5185–5197. [[CrossRef](#)]
31. Sun, H.; Zhou, Y.; Tang, J.; Kang, Z.; Wang, X.; Quek, T.Q.S. Average AoI-Minimal Trajectory Design for UAV-Assisted IoT Data Collection System: A Safe-TD3 Approach. *IEEE Wirel. Commun. Lett.* **2024**, *13*, 530–534. [[CrossRef](#)]
32. Al-Iqubaydhi, N.; Alenezi, A.; Alanazi, T.; Senyor, A.; Alanezi, N.; Alotaibi, B.; Alotaibi, M.; Razaque, A.; Hariri, S. Deep Learning for Unmanned Aerial Vehicles Detection: A Review. *Comput. Sci. Rev.* **2024**, *51*, 100614. [[CrossRef](#)]
33. Wu, Q.; Zeng, Y.; Zhang, R. Joint Trajectory and Communication Design for Multi-UAV Enabled Wireless Networks. *IEEE Trans. Wirel. Commun.* **2018**, *17*, 2109–2121. [[CrossRef](#)]
34. Liu, J.; Wang, X.; Bai, B.; Dai, H. Age-Optimal Trajectory Planning for UAV-Assisted Data Collection. In Proceedings of the IEEE INFOCOM 2018—IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), Honolulu, HI, USA, 15–19 April 2018; pp. 553–558. [[CrossRef](#)]
35. Hourani, A.; Kandeepan, S.; Jamalipour, A. Modeling Air-to-Ground Path Loss for Low Altitude Platforms in Urban Environments. In Proceedings of the GLOBECOM'14, Satellite & Space Communication, Austin, TX, USA, 8–12 December 2014.
36. Al-Hourani, A.; Kandeepan, S.; Lardner, S. Optimal LAP Altitude for Maximum Coverage. *IEEE Wirel. Commun. Lett.* **2014**, *3*, 569–572. [[CrossRef](#)]
37. Zeng, Y.; Xu, J.; Zhang, R. Energy Minimization for Wireless Communication With Rotary-Wing UAV. *IEEE Trans. Wirel. Commun.* **2019**, *18*, 2329–2345. [[CrossRef](#)]
38. Mozaffari, M.; Saad, W.; Bennis, M.; Debbah, M. Mobile Unmanned Aerial Vehicles (UAVs) for Energy-Efficient Internet of Things Communications. *IEEE Trans. Wirel. Commun.* **2017**, *16*, 7574–7589. [[CrossRef](#)]

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.