The Journal of Supercomputing (2025) 81:83
https://doi.org/10.1007/s11227-024-06574-z

A novel hybrid path planning method for sweep coverage of multiple UAVs

Recep Özdağ[1]

Accepted: 29 August 2024
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2024

Abstract

Sweep coverage (SC) is an NP-hard problem that needs to be solved with a small number of mobile sensors (sensor nodes) to monitor or observe targets in wireless sensor networks (WSNs) over a period of time. One of the proposed solutions to optimize the lifetime of WSNs is to monitor targets in the area of interest (AoI) by path planning of the sensor nodes. Considering the applications of unmanned aerial vehicles (UAVs) in military and civilian fields, in this study, we model multiple UAVs to solve the SC problem based on non-uniformly distributed targets in the AoI. In this modeling, we conduct effective coverage path planning for UAVs by performing an SC task for targets in the AoI with multiple asynchronously flying UAVs. To this end, we propose a new approach, which we call Weighted Targets Sweep Coverage, based on the metaheuristic Crow Search Algorithm and the heuristic Greedy algorithm. In our proposed approach, we define a new objective function by considering the UAV task completion time and the AoI coverage rate as metrics. In the Monte Carlo simulations that we perform to measure the performance, we run 5 different scenarios considering the number of targets defined for AoI and the alpha parameters defined for the objective function. The results of the scenario studies show that the proposed approach outperforms the other algorithms by up to 21% and 50% in terms of UAV task completion time and AoI coverage rate, respectively.

Keywords Coverage path planning · Sweep coverage · Crow Search Algorithm · Greedy algorithm · Unmanned aerial vehicle · Optimization

✉ Recep Özdağ
rozdag@yyu.edu.tr

[1] Department of Computer Engineering, Faculty of Engineering, Van Yüzüncü Yıl University, Van, Turkey

Published online: 25 October 2024

1 Introduction

Wireless sensor networks (WSNs) are actively used in applications such as border and environmental monitoring to ensure the safety of the environment or to detect vehicle density on the roads through environmental monitoring. Optimal coverage rates when continuously monitoring or observing targets in the area of interest (AoI) actually means that too many mobile sensors (sensor nodes) are deployed in the AoI. Also, tracking targets in environmental surveillance means that these targets are patrolled by sensor nodes within certain time periods. Continuous monitoring of the AoI requires the deployment of a large number of mobile sensors, so continuous monitoring of the AoI may not be necessary in some monitoring environments [1]. This is because continuous tracking of targets both increases the cost of deploying additional sensor nodes and reduces the lifetime of the network as the energy of mobile nodes is quickly consumed. If the targets can be tracked within certain time periods, more targets can be tracked with fewer mobile nodes [2]. The advancement of microelectromechanical technology has enabled the regular tracking of targets by mobile nodes. Therefore, the periodic tracking of targets has necessitated the study of sweep coverage (SC) as a new coverage problem [2, 3].

With the rapid development of unmanned aerial vehicle (UAV) technology in the last decade, the number of applications and research in military and civilian fields such as environmental monitoring and exploration, search and rescue, climate monitoring and mapping has increased [4–7]. Considering the portability of UAVs, their ability to move quickly, their ability to fly over a large area at a certain altitude, their long operating time (hovering) and their ability to turn at a minimum angle, UAVs are actively used for environmental monitoring. For example, some areas (targets) in the forest region where there is a high probability of fire outbreak can be monitored by UAVs in certain periods, which enable the monitoring of risky targets. Therefore, by conducting SC in AoI, patrolling by UAVs makes an important contribution to the early detection and control of forest fires [8, 9]. In addition, thanks to the technologies developed as the next-generation mobile communication standard (4G/5G/6G), UAVs can be used as flying base stations [10] to provide downlink and uplink wireless communication services, and this service is called UAV-based communication. In environments where terrestrial ground stations (GSM base stations) cannot be positioned due to unfavorable terrain conditions, UAVs play an important role in establishing coverage areas for ground users [11] or in collecting data from smart devices (downlink) and transmitting it to GSM stations (uplink) [12].

Over the last decade, UAVs have been used more for aerial surveillance of targets than sensor nodes. Since the battery life of UAVs is limited, they perform their task flights as long as their battery power is above the threshold. Therefore, in order to realize effective SC [13] by UAVs in AoI, the targets to be flown should be determined in advance and the optimal path planning should be carried out. When determining the priority targets to be patrolled during path planning, the limited flight time of the UAV should be considered. If a single UAV is

used to regularly track targets distributed over a very large area, effective SC on targets cannot be performed due to the rapidly consumed battery life. However, when multiple UAVs are deployed, effective coverage path planning (CPP) can be performed within each UAV's flight time to prevent battery life from being consumed quickly in complex environments and ensure optimal tracking of targets. CPP is an NP-hard optimization problem that is often related to the traveling salesman problem [14]. The CPP problem can be classified as online and offline based on the knowledge of the targets in the AoI [15]. The offline method [13, 16] assumes that the target locations are known before the flight. Therefore, this method is more suitable for scenarios where the target information in the AoI does not change, i.e., the target distribution environment is certain. The online method [15, 17], on the other hand, is better suited for scenarios in environments where the target information is uncertain, as it considers variable target locations (without prior knowledge).

1.1 Related work

In this section, we review the literature on the UAV-CPP problem, which is relevant to our study, and on the UAV-SC problem, which is the main scope of our study.

1.1.1 The CPP of UAVs

Considering the literature studies, CPP in [18] determines the optimal flight path between departure and arrival points based only on flight environment constraints (obstacles, mountains, military zones, etc.). For the CPP problem, solutions based on the optimization metrics of minimum energy consumption [19], minimum flight time [20] and maximum coverage rate [21] were provided to determine the flight path with UAVs at minimum cost.

Numerous techniques for solving the CPP problem for multiple UAVs have been studied in the literature. These can be categorized into (*i*) studies that represent UAVs in a 3D environment to identify objects and obstacles in the AoI and (*ii*) studies based on collaborative techniques for the CPP of UAVs [22]. In (*i*), studies were conducted on the representation of UAVs in 3D environments based on sampling-based techniques (cell decomposition [23], roadmaps [24], potential fields [25], Voronoi diagram [26]) and artificial intelligence-based techniques (heuristic search techniques [27], brute force search techniques [28], local search techniques [29], artificial neural networks [30]). In (*ii*), studies have been conducted on many collaborative learning methods based on mathematical (linear programming [31], control theory [32], metaheuristic algorithms [33, 34], probabilistic models [35]), bioinspired (evolutionary techniques [36, 37]), machine learning (supervised learning [38], unsupervised learning [39], reinforcement learning [40]) and multi-objective optimization models (convex optimization [41], two-echelon optimization [42]).

Since UAVs are autonomous flying vehicles subject to both kinematic and dynamic constraints, the best solution among all possible options can be achieved

by optimizing the UAVs considering the environment and flight constraints. Therefore, the heuristic search technique is one of the most important techniques that enables the UAV to achieve the optimal CPP by flying to the optimal cost targets between the take-off point and the destination point. Some of the heuristic search methods used for UAV-CPP have been developed from metaheuristic-based optimization algorithms such as Particle Swarm Optimization (PSO), Artificial Bee Colony and Grey Wolf Optimizer (GWO). In [34], a novel hybrid PSO algorithm was proposed to solve the CPP problem in a complicated environment with multiple UAVs. The author developed a hybrid SDPSO by combining the Simulated Annealing Algorithm, which avoids local convergence, with the PSO, which provides a global optimal solution. In [43], a novel hybrid HSGWO-MSOS was proposed that combines the simplified GWO (SGWO) with the modified search for symbiotic organisms (MSOS). In the algorithm, the author effectively combined the exploration and exploitation abilities of wolves and organisms, adapted them to the CPP problem and then performed a convergence analysis guided by the linear difference equations. In [44], a multi-objective evolutionary algorithm (MOEA) was proposed to solve the CPP problem. The author used MOEA for multi-target convergence, which aims to minimize the total distance and radar detection of targets in a terrain continuously monitored by radars. In [45], a differential evolution (DE) algorithm was proposed to solve the CPP as a multi-objective optimization problem in a 3D terrain disaster scenario. The author formulated an objective function in a knee-guided DE algorithm that incorporates the computed path risk and distance according to Bézier theory. In [46], a novel genetic algorithm extended with fuzzy logic (GAF) was proposed to tackle CPP considering multiple objective metrics such as distance, energy efficiency and path curvature. Ntakolia et al. [46]. applied multi-objective optimization to the CPP problem by using the proposed GAF to minimize the flight distance of the UAV, maximize the energy efficiency and ensure the smoothness of the path.

The literature studies presented in this section generally focus on covering the entire AoI and aim to generate CPP by exploring the AoI only once with multiple UAVs. However, in the SC problem, which is the main topic of this paper, the SC task is only performed for distributed targets in the AoI, not for the entire AoI. Therefore, the methods used to solve the UAV-CPP problem do not correspond to the mathematical model of this paper.

1.1.2 The SC of UAVs

Many studies have been conducted in the literature in which effective path planning is achieved by performing SC on distributed targets in the AoI within certain time periods by mobile sensors [2, 47–51] or UAVs [13, 16, 52].

In [2], an algorithm called 2-Partition Sweep Coverage (2-PSC), which considers the coverage time requirements and node locations, was proposed to solve the t, K-sweep coverage problem with a minimum number of mobile sensor nodes. In this study, the t, K-sweep coverage problem was solved by considering K different sweep routes within t sweep periods. The 2-PSC algorithm generates sweep routes by considering the length of the sweep route as well as the coverage requirements,

so that the search for targets is performed by the sensor nodes moving along these routes.

In [49], approximation algorithms for 3 types of sweep cover problems were investigated, namely the partial sweep cover problem (PSC), the price-collecting sweep cover problem (PCSC) and the budgeted sweep cover problem (BSC). In this study, the PSC problem attempts to use a minimum number of mobile sensors to SC on a minimum number of targets. The PCSC problem aims to minimize the cost of mobile sensors, while the BSC problem uses budgeted mobile sensors to perform SC on as many targets as possible.

In [50], the Chargeable Sweep Coverage (CSC) problem is proposed to ensure that the mobile sensors return to their charging stations without running out of energy for effective SC on targets in the AoI, and thus the mobile sensors are continuously recharged and cover the targets within periodic time periods. In their work, they model the CSC problem by creating an unbounded number of candidate paths and including a maximum number of mobile sensors.

In [51], a new problem called Cooperative Sweep Coverage (CSC) is defined for periodic coverage of all targets in AoI by multiple mobile sensors. In their study, they have provided the deployment of multiple sensors on the trajectory to reduce the sweep time or detection delay based on the CSC problem. In addition, based on the Multi-Sink Sweep Coverage (MSSC) problem, they transmit the data collected by mobile sensors with limited storage and battery capacity to the base station at regular intervals. They proposed constant factor approaches called CoCycle and SinkCycle to minimize the sweep time considering both problems.

Although the SC problem has been applied to mobile nodes in WSNs, it has also been adapted in the literature to periodic tracking of targets by UAVs, modeling the UAVs as mobile nodes. When we examined the literature studies, we found that the research related to UAV-SC for periodic tracking (patrolling flight) of targets within the AoI was conducted based on UAV flight constraints, using the departure and arrival points of UAVs as a starting point.

In [13], UAV-SC path planning was investigated, where both synchronous and asynchronous flights are performed based on multiple UAVs to periodically track distributed targets in the AoI. In this study, a weighted targets sweep coverage (WTSC) algorithm was proposed to solve the min-time max-coverage (MTMC) problem and achieve maximum coverage in minimum time. They also established a mathematical model that considers the higher priority of important targets for more effective UAV-SC. However, looking at the objective function in their study, which is calculated with a coefficient based on the weight sums of the targets swept by the UAV, it is clear that this coefficient constrains the further minimization of the objective function.

In [52], a UAV path planning algorithm (PCBGA) based on a greedy strategy and ant colony optimization algorithms was proposed to perform the UAV-SC task in AoI. In their study, they optimized the UAV flight path using the ant colony optimization algorithm to achieve maximum SC rate while minimizing the UAV flight time. However, when calculating the cost function, they minimized the UAV flight time to the target while ensuring the selection of the minimum

weighted target. Therefore, in their study, optimization with this cost function causes in the UAV achieving a low SC rate for the objectives it tracks.

1.1.3 Research gaps

It can be seen that the coefficient of the SC ratio in the objective function defined in the study [13] needs to be improved and that the cost function defined in the study in [52] is not sufficient to select high priority target points. Therefore, we developed a hybrid-based optimization approach to achieve more optimal results through these improvements that we highlighted in [13] and [52], where the same type of SC problem as in our work was studied.

As far as we know, no optimization approach based on Crow Search Algorithm (CSA) and Greedy algorithm (GA) has yet been developed in the literature to solve the SC problem with multiple asynchronously flying UAVs.

1.2 Our contribution

The details of the contribution of this paper can be summarized as follows.

- We consider the UAV-SC problem, where distributed targets in the AoI are periodically tracked by multiple asynchronously flying UAVs to achieve maximum coverage in minimum time.
- We have developed a hybrid WTSC-CSGA approach by combining CSA, a multidimensional and swarm-based metaheuristic algorithm with low time complexity, and a heuristic GA with selected nearest neighbor solution to achieve the single target with minimum cost by generating iterative solutions under multiple targets.
- We create a new multiparameter objective function based on the task completion time (TCT) of multiple UAVs and the coverage rate of the AoI, which is calculated by considering the SC ratio of the UAVs.
- We ensure that critical (important) targets in the AoI are followed with priority by considering the weight values assigned to each target, which indicate the importance of the targets.

The rest of this paper is organized as follows. We outline the formulation of the UAV-SC problem by introducing some preliminary concepts in Sect. 2. Section 3 presents the details of the proposed WTSC-CSGA approach. Section 4 presents the simulated experimental results and discussions. Section 5 contains the conclusions of this study and gives an outlook on planned future work.

2 Problem formulation

In this section, we present a general mathematical model description of the UAV-SC problem by both specifying the SC constraints in the operational environment where multiple UAVs are distributed and formulating the descriptions based on the flight pattern of the UAVs in the environment.

2.1 Distribution environment

Assume that the targets are unevenly distributed in the AoI and that these targets are periodically tracked by multiple UAVs. Considering the following constraints [13], the SC task is performed within the AoI (Fig. 1).

- UAV-SC takes place when each target in the AoI is covered by at most one UAV.
- The take-off and landing routes for multiple UAVs are determined according to the base point P^0.
- The UAV-SC task is completed when the first UAV takes off from the P^0 and the last UAV returns to this P^0.
- To perform UAV coverage path planning (CPP) offline, static targets must be used in the AoI.
- To achieve an effective CPP for multiple UAVs, a very large AoI must be defined as the target environment.

As shown in Fig. 1, N targets are non-uniformly distributed within the AoI and M UAVs are deployed to monitor these targets. By randomly assigning weight values representing the importance of each target, the critical targets are

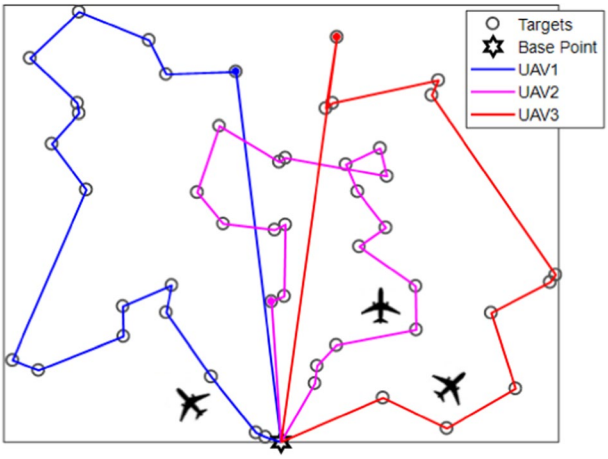


Fig. 1 CPP determined with multiple UAVs for SC

prioritized within the AoI. Therefore, the inclusion of critical targets in the CPP is based on targets with a high weighting.

The TCT for the SC of targets in AoI with multiple UAVs is the time interval between the take-off of the first UAV and the return of the last UAV [13]. Due to the limited battery power of the UAVs, the SC task cannot be performed for all targets distributed in the AoI, so only important targets can be tracked with multiple UAVs. Therefore, considering the limited battery power depending on the target distribution environment, targets with minimum TCT and maximum weight should be

Table 1 List of symbols for parameters

Symbol	Meaning
N	The number of target points
M	The number of UAVs
V	The linear velocity of the UAV
w	The angular velocity of the UAV
P	The set of target points
P^0	Take-off point and destination of the UAV in AoI
R	The detection radius on the ground of the UAV
(x_i, y_i)	The plane coordinate of the i-th target P_i
d_{ij}	The flight (Euclidean) distance from P_i to P_j
θ_{ijp}	The turning angle at P_j of the UAV
$U(k)$	The k-th element of the set of UAVs
$ST(k)$	The swept targets of the k-th UAV
$P_iP_j(k)$	Flight of the k-th UAV from P_i to P_j
$T_{j->p}$	The flight time from P_j to P_p of the UAV
$T_{\text{total}}(k)$	The total flight time of the k-th UAV
$T_{\max}$	The maximum flight time of UAVs
O	The number of UAV operators
T^s	The setup time for UAVs
$T^w(k)$	The waiting time of the k-th UAV
$T^c(k)$	The cumulative flight time of the k-th UAV
SC^{tct}	The task completion time of the SC
W_i	The weight of P_i
$W^{\text{total}}(k)$	The total weight of ST(k)
AoI^{cr}	The coverage rate of the AoI
cost_i	The cost function of P_i
F	The objective function
α	The tuning parameter
$x^{i,\text{iter}+1}$	The new location of the i-th crow at $iter+1$
$m^{j,\text{iter}}$	The best location of the j-th crow at $iter$
$fl^{i,\text{iter}}$	The flight length of the i-th crow at $iter$
$\text{AP}^{j,\text{iter}}$	The awareness probability of the j-th crow at $iter$
$L_{\text{xy}} - U_{\text{xy}}$	Upper and lower plane bound coordinates of the solution space

tracked with multiple UAVs for effective SC. The list of symbols for this study is presented in Table 1.

2.2 UAV flight model

The SC task is performed with linear (straight) and nonlinear (turning) flights between targets in the AoI. For a straight flight between two fixed targets, the linear velocity, represented by the constant V, is used. For turning flights, the angular velocity, represented by the constant w, is used. Assuming that a set of stationary target points $P = \{P_1, P_2, ..., P_N\}$ is non-uniformly (randomly) distributed within the AoI, the UAV performs a straight flight from the i-th target P_i to the j-th target P_j, followed by a turn flight from P_j to the p-th target P_p with an angle θ_{ijp} (Fig. 2) [13]. The UAV flight models to be considered in the SC task are defined as follows:

2.2.1 Definition 1 (*distance of flight*)

In the case that UAV performs a straight flight from P_i to P_j, the flight length between these targets is calculated using the following equation to determine the optimal flight length between the targets [13].

$$d_{\text{ij}} = \sqrt{\left(x_i - x_j\right)^2 + \left(y_i - y_j\right)^2},\ i,j \in \{1,2,...,N\}, \tag{1}$$

where x and y stand for the 2-D (two-dimensional) coordinate plane of the targets P_i and P_j, respectively, and d_{ij} for the flight distance in Euclidean length.

In Fig. 2, after the flight of the UAV from P_i to P_j, the distances of the straight flight paths to P_p or P_q are equal ($d_{\text{jp}} = d_{\text{jq}}$), which assumes that the flight times at these targets are also equal. However, with regard to the UAV's AoI applications, it should be noted that the time consumed during the turning process at point P_j cannot be neglected, since $\theta_{\text{ijp}} < \theta_{\text{ijq}}$ is the turning angle. Even if d_{jp} is equal to d_{jq}, the UAV flight to P_p will take less time if the different turning times due to the different turning angles at P_j are also considered. So when considering UAV flights to targets, accurate flight times should be calculated for real-area applications by considering the turn time based on the turn angle along with the straight flight distance.

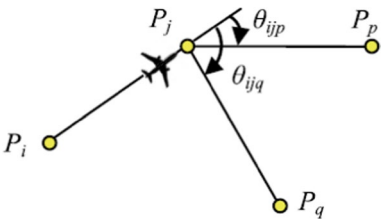


Fig. 2 Diagram of UAV's straight and turning flight at P

2.2.2 Definition 2 (*angle of turning*)

As shown in Fig. 2, the turning angle of the UAV at P_j during a turn from P_j to P_p can be calculated using the cosine law in the following equation [16]. It should be noted that a turning angle between 0 and π can be calculated in this equation.

$$\theta_{\mathrm{ijp}} = \pi - \arccos \frac{\left(\vec{d}_{\mathrm{ij}}\right)^2 + \left(\vec{d}_{\mathrm{jp}}\right)^2 - \left(\vec{d}_{\mathrm{ip}}\right)^2}{2 * \vec{d}_{\mathrm{ij}} * \vec{d}_{\mathrm{jp}}},\ i,j,p \in \{1,2,...,N\}, \tag{2}$$

where the distance vectors d_{ij}, d_{jp} and d_{ip} represent the flight paths from P_i to P_j, P_j to P_p and P_i to P_p, respectively. The turning angle at P_j is also denoted by θ_{ijp} between d_{ij} and d_{jp}.

2.2.3 Definition 3 (*time of flight*)

The flight time refers to the time interval required by the UAV to fly from its current position at P_j to the next position at P_p. The calculation of the flight time therefore consists of two parts. In the first part, the time for the turn flight from P_j to P_p during a turn maneuver and in the second part the time from P_j to P_p during a straight flight are added using the following equation [16] to determine the total flight time from P_j to P_p, which is denoted by $T_{j->p}$

$$T_{j->p} = \frac{\theta_{\mathrm{ijp}}}{w} + \frac{d_{\mathrm{jp}}}{V},\ i,j,p \in \{1,2,...,N\}, \tag{3}$$

where w and V represent the velocity constants of the UAV in turning flight and straight flight, respectively.

The total flight time for the $U(k)$ tracking (sweeping) multiple targets is obtained by calculating the cumulative sum of the flight times at each target tracked by $U(k)$ using the following equation, denoted by $T_{\mathrm{total}}(k)$

$$T_{\mathrm{total}}(k) = \begin{cases} \sum_{i=1}^{N} T_i * 1, & if\ P_i \in \mathrm{ST}(k) \\ \sum_{i=1}^{N} T_i * 0, & \text{otherwise} \end{cases},\quad k \in \{1,2,...,M\}, \tag{4}$$

where $ST(k)$ represents the targets swept by $U(k)$. T_i represents the flight time from any target to P_i.

2.2.4 Definition 4 (*time of waiting*)

In practice, to perform the SC task, each UAV requires a set-up and standby time to get the UAV ready to fly, including connecting the battery and setting the GPS (Global Positioning System) before take-off. When a single operator manages multiple UAVs, it is impossible to prepare all UAVs for flight at the same time, resulting in different pre-flight waiting times for each UAV. In the case where the number of operators is equal to the number of multiple UAVs, each operator is responsible for the flight of a single UAV, and therefore the waiting times of the UAVs correspond to the set-up times. The waiting time determined for $U(k)$ is calculated according to the following equation [13], denoted by $T^w(k)$

$$T^w(k) = T^s * \left(\frac{k}{O}\right), k \in \{1, 2, ..., M\}, \tag{5}$$

where T^s stands for the UAV set-up time and O for the number of UAV operators.

Let us assume, for example, that T^s is 8 min and M is 3. If O is defined as 3, then $(k/O) = 1$ according to Eq. (5) and thus $T^w(k) = T^s$ for $\forall U(k)$. Thus, each operator prepares its UAV for flight within the same waiting time. If O is defined as 1, then $T^w(k)$ is calculated as 8, 16 and 24 min, respectively, so that a single operator prepares each UAV for flight at the end of different waiting times [13].

2.2.5 Definition 5 (*time of cumulative flight*)

The cumulative flight time is the sum of the waiting time required to make $U(k)$ ready for flight and the flight times for the targets it flies to, denoted by $T^c(k)$

$$T^c(k) = T^w(k) + T_{\text{total}}(k),\ k \in \{1, 2, ..., M\}, \tag{6}$$

2.2.6 Definition 6 (*task completion time*)

When determining the TCT for UAV-SC, the flight time of the UAV that last returned to base is taken into account. TCT is therefore the cumulative flight time of the UAV that last returned to base and is calculated based on the maximum of the cumulative flight time of the UAVs, denoted by SC^{tct}

$$\text{SC}^{\text{tct}} = \max\{T^c(k)\},\ k \in \{1, 2, ..., M\}. \tag{7}$$

2.3 Model description of the UAV-SC problem

If we take the above definitions of the UAV flight model as the basis for solving the UAV-SC problem, the straight or turning flight states of the UAV between the targets in the AoI should be determined first. As can be seen in Fig. 2, when the UAV performs a flight from P_i to P_j, the straight flight distance of the UAV between these targets should be calculated based on the Euclidean distance according to Eq. (1). In the case that the UAV performs a flight from P_j to P_p, the turning angle θ_{ijp} should first be calculated according to Eq. (2) and then the straight flight distance according to Eq. (1). So, considering both turning and straight flights of the UAV, the flight time of the UAV from P_j to P_p can be calculated according to Eq. (3). However, if the UAV flies from P_i to P_p, both the flight time of the UAV from P_i to P_j and the flight time from P_j to P_p should be calculated and added according to Eq. (4). Therefore, the total flight time at each $ST(k)$ followed by $\forall U(k)$ should be calculated according to Eq. (4), where $k \in \{1, 2, ..., M\}$. In addition, when solving the UAV-SC problem, the waiting time before launching each UAV should be calculated according to Eq. (5), taking into account the number of operators launching multiple UAVs. For $\forall U(k)$ with departure and arrival point P^0, the cumulative flight time should therefore be calculated by adding the waiting time and the total flight time according to Eq. (6). Consequently, the SC task completion time of multiple UAVs can be calculated based on the maximum cumulative flight time of the UAV that last returned to P^0, as shown in Eq. (7).

3 Method

In this paper, we optimize the UAV-SC problem using the metaheuristic CSA [53] for swarm-based UAV modeling and the heuristic GA [54] for target assignment. In this section, we also present the mathematical models of CSA, GA and the hybrid approach in detail, which we consider as optimization methods in our study.

3.1 Crow Search Algorithm

CSA is a novel metaheuristic algorithm inspired by the intelligent behavior of crows. Crows, a genus of birds, are widely regarded as one of the most intelligent members of the animal kingdom. When they act collectively, they display extraordinary intelligence. They are able to memorize the faces of all living creatures, communicate with each other in sophisticated ways, and store and retrieve their food as the seasons change. Crows observe other birds to locate hidden food and

then steal it. Crows thus use their experience with theft to determine the safest route and take extra measures to prevent their food from being stolen. A swarm of crows shows similarities with the optimization process through its following behavior [53].

- Crows store their surplus food in hidden places in their territory and pick it up when needed.
- They follow each other to reach better food sources.
- When a crow realizes that it is being followed, it retreats to another place nearby to distract the pursuing crow from its hidden food supply.

In optimization, crows act as searchers, the environment is the search space (solution space), each location in the environment corresponds to a solution, the quality of the food source is the objective function, and the best food source in the environment indicates the solution to the problem. Because of these similarities, CSA can be used to solve optimization problems by simulating the intelligent behavior of crows.

Suppose there is a swarm of crows in a d-dimensional environment and its size is N (number of crows in the swarm). The position of the i-th crow at $iter$ iteration in the solution space is represented by a vector $x^{i,\,\mathrm{iter}}(i = 1, 2, ..., N$ and $\mathrm{iter} = 1, 2, ..., \mathrm{iter}_{\max})$, where $x^{i,\,\mathrm{iter}} = \left[x_1^{i,\,\mathrm{iter}}, x_2^{i,\,\mathrm{iter}}, ..., x_d^{i,\,\mathrm{iter}}\right]$ and $iter_{\max}$ is the maximum iteration number. Each crow has a memory in which it remembers the place where it has hidden its food. By memorizing its best position, each crow searches for the best food source in the swarm where it can hide its food and thus reaches its optimal position at $iter$. The optimal position for the i-th crow to hide its food is denoted by $m^{i,\,\mathrm{iter}}$.

Let us assume that at $iter$, the j-th crow wants to visit the location $m^{j,\,\mathrm{iter}}$ where it has hidden its food. At this $iter$, the i-th crow decides to follow the j-th crow to approach $m^{j,\,\mathrm{iter}}$. In this case, as shown in Algorithm 1 of the CSA pseudocode [53], the following two situations can occur.

State 1 The j-th crow does not know that the i-th crow is following it, and so the i-th crow approaches the place where the j-th crow is hiding its food. In this case, the new position of the i-th crow is calculated using Eq. (8).

State 2 The j-th crow knows that the i-th crow is following it. The j-th crow deceives the i-th crow by moving to a random place in the solution space so that its food is not stolen.

Overall, *State 1* and *State 2* can be modeled by the following equation, which calculates the new position of the i-th crow at $iter+1$, denoted by $x^{i,\,\mathrm{iter}+1}$

$$x^{i,\,\mathrm{iter}+1} = \begin{cases} x^{i,\,\mathrm{iter}} + r * fl^{i,\,\mathrm{iter}} * \left(m^{j,\,\mathrm{iter}} - x^{i,\,\mathrm{iter}}\right) & r \geq \mathrm{AP} \\ L_{\mathrm{xy}} + r * \left(U_{\mathrm{xy}} - L_{\mathrm{xy}}\right) & \text{otherwise} \end{cases}, \tag{8}$$

where r is a randomly generated number between 0 and 1, $\mathrm{fl}^{i,\,\mathrm{iter}}$ is the flight length of the crow, AP is the awareness probability of the crow, L_{xy} and U_{xy} are the lower and upper bound coordinates of the solution space, respectively.

If the j-th crow is followed by the i-th crow, the influence of the parameter $\mathrm{fl}^{i,\,\mathrm{iter}}$ on the search for the solution space is taken into account. Small values of this parameter lead to a local search close to $x^{i,\,\mathrm{iter}}$, while large values lead to a global search far away from $x^{i,\,\mathrm{iter}}$. In metaheuristic algorithms, diversification enables the exploration of different regions in the solution space, while intensification facilitates the identification of high-quality solutions in these regions to achieve a balance in the development of search methods [55]. In CSA, the parameter AP controls this balance. Thus, a small value of AP increases the intensity by performing a local search in the solution space, while a large value of AP increases the diversification by performing a global search.

Algorithm 1 CSA

1: Randomly determine the d-dimensional initial positions of the swarm of K crows in the solution space
2: Evaluate the d-dimensional initial positions of each K-swarm.
3: Store the d-dimensional initial positions of each K-swarm optimally in memory
4: **while** $iter \le iter_{\max}$
5: **for** i=1 to K **do**
6: Randomly select any K-swarm to follow (e.g. j)
7: Define an AP between 0 and 1
8: **if** $r_j \ge AP^{j,iter}$ **then**
9: $x^{i,iter+1} = x^{i,iter} + r_i * fl^{i,iter} * \left(m^{j,iter} - x^{i,iter}\right)$
10: **else**
11: Determine a random positions with d-dimension in the solution space for $x^{i,iter+1}$
12: **end if**
13: Check the suitability of the new positions of the i-th swarm in the solution space
14: Update the new position of the i-th swarm according to the optimal value in memory
15: **end for**
16: **end while**

3.2 Greedy algorithm

The GA is a heuristic optimization algorithm that selects the most optimal choice when solving the problem. By selecting the best option at each step, the aim is to achieve an optimal solution quickly without the need for complicated calculations. The time complexity of GA is better than that of other algorithms developed to solve the same problem. Successful results have already been achieved with GA on many problems [56].

In this study, CPP optimization for UAV-SC is performed using GA to select the optimal cost (minimum flight time and maximum weight) from the target points in

each defined problem dimension. Thus, we used GA as a target assignment strategy to determine the optimal costly target for solving the SC problem.

3.3 WTSC-CSGA approximation

In this paper, we present the WTSC-CSGA approach, a hybrid algorithm that combines CSA and GA to determine the $ST(k)$ for $\forall U(k)$, where $k \in \{1, 2, ..., M\}$. We propose this approach as a novel path planning algorithm for efficiently solving the SC problem of UAVs. Since CSA, the basis of our proposed approach, is a metaheuristic optimization algorithm, we first generate iterative solutions with CSA to avoid the problem of frequently falling into optimal local solutions during optimization. Then, we achieve the optimal solution by selecting the solution with the lowest cost among these solutions using the GA heuristic. We perform the optimization of the UAV-SC problem based on the following definitions for WTSC-CSGA:

3.3.1 Definition 7 (*total weight of targets*)

This is the total weight of the targets swept by $\forall U(k)$ during the execution of the SC task, denoted as $W^{\text{total}}(k)$

$$W^{\text{total}}(k)\begin{cases} \sum_{k=1}^{M}\sum_{i=1}^{N} W_i * 1 & if\ P_i \in \text{ST}(k) \\ W_i * 0 & \text{else} \end{cases}, \tag{9}$$

where W_i is the weight of P_i.

3.3.2 Definition 8 (*coverage rate of AoI*)

The coverage rate of the AoI is the ratio of $W^{\text{total}}(k)$, calculated according to Eq. (9), to the sum of the weights of all targets in the AoI, denoted by AoI^{cr}

$$\text{AoI}^{\text{cr}} = \sum_{k=1}^{M} W^{\text{total}}(k) \Big/ \sum_{i=1}^{N} W_i, \tag{10}$$

3.3.3 Definition 9 (*cost function*)

In this study, UAV-SC optimization is performed by considering the flight time and weight of the target, while the targets with minimum cost are determined using WTSC-CSGA. Therefore, we consider targets with maximum weights and minimum flight time when optimizing the target costs. The cost function of the targets is calculated according to the following equation and denoted by cost_i

$$\text{cost}_i = \alpha * T_i + (1-\alpha)/W_i, i \in \{1,2,...,N\} \tag{11}$$

where α is a tuning parameter that reflects the importance of the UAV's flight time at P_i and the weight value of P_i and takes a value in the range from 0 to 1. A high value of this parameter increases the importance of the flight time at P_i, while a low value increases the importance of the weight value of P_i.

3.3.4 Definition 10 (*objective function*)

In WTSC-CSGA, which was developed as a novel metaheuristic-based approximation algorithm, the objective function (fitness) is designed to transform the multi-objective optimization problem into a single-objective optimization problem based on SC^{tct} and AoI^{cr} and is denoted by F

$$F = \alpha * \text{SC}^{\text{tct}} + (1-\alpha)/\text{AoI}^{\text{cr}}. \tag{12}$$

where α is a tuning parameter that assumes a value in the range from 0 to 1 and establishes a balance between SC^{tct} and AoI^{cr}. It is therefore used as an equilibrium coefficient to optimize F.

3.3.5 Definition 11 (constraints of the UAV-SC problem)

In optimizing the UAV-SC problem with WTSC-CSGA in this study, we optimize F to the minimum (Eq. 13) by optimizing SC^{tct} to the minimum and AoI^{cr} to the maximum. This optimization is as follows:

$$\min\ (F) \tag{13}$$

$$\sum_{i=0}^{N}\sum_{j=1}^{N} P_iP_j(k) = 1, k \in \{1,2,...,M\} \tag{14}$$

$$T_{\text{total}}(k) \le T_{\text{max}},\ k \in \{1,2,...,M\} \tag{15}$$

$$\sum_{i=0}^{N} P_iP_j(k) = \sum_{p=0}^{N} P_jP_p(k), j \in \{1,2,...,N\}, k \in \{1,2,...,M\} \tag{16}$$

where T_{max} is the maximum flight time that $\forall U(k)$ can fly. The $P_iP_j(k) = 1$ represents the flight from P_i to P_j with only one UAV (k-th UAV). $P_iP_j(k)$ and $P_jP_p(k)$ represent $U(k)$ flying from P_i to P_j and from P_j to P_p, respectively.

As shown in Eq. (14), the first constraint of the problem is defined such that only one UAV, namely $U(k)$, must fly from P_i to P_j and follow these targets. As shown in Eq. (15), the second constraint of this problem is that the total flight time of $U(k)$ at the minimum cost P_i targets must not exceed the maximum flight time of the UAV. Equation (16) also guarantees that the UAV that flies from P_i to P_j and arrives in P_j has the same $U(k)$ as the UAV that flies from P_j to P_p and leaves P_p. Thus, if both the

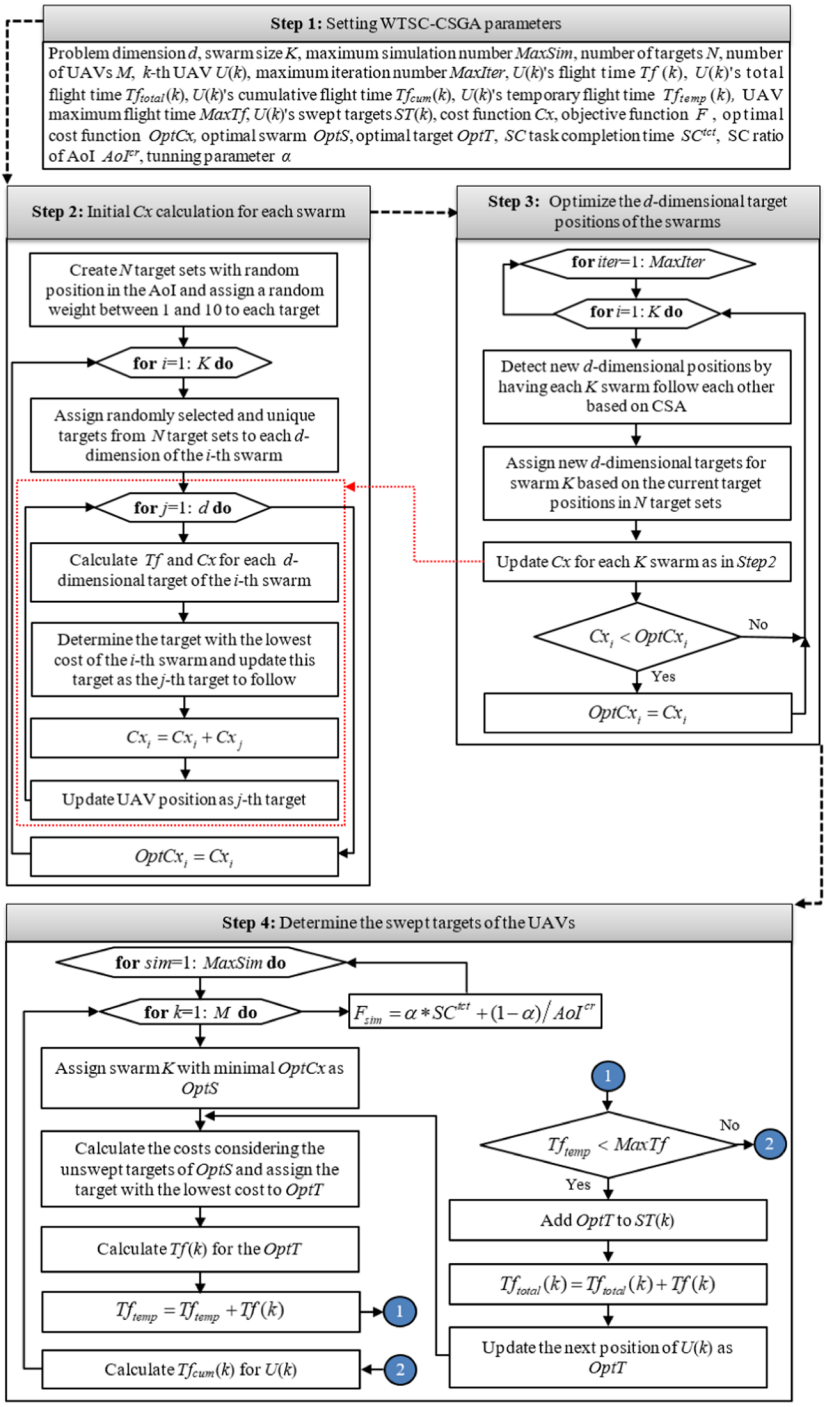


Fig. 3 The flow diagram of the WTSC-CSGA

first and second constraints are met, the ST(k) are determined on the basis of the P_i targets with the minimum cost$_i$, i.e., $P_i \in$ ST(k).

3.3.6 Algorithm descriptions

The WTSC-CSGA approach we developed to solve the UAV-SC problem consists of four steps. Based on these steps, we present the pseudocode of the WTSC-CSGA approach in Algorithm 2 and the flow diagram of this approach in Fig. 3.

Algorithm 2 WTSC-CSGA

Input: It contains the parameters from *Step1* in Fig. 3: d, K, $MaxSim$, N, M, $MaxIter$, $MaxTf$, α
It contains the parameters in Table 1: V, w, P, P^0, O, $T^w(k)$, W_i, $fl^{i,iter}$, $AP^{j,iter}$, $L_{xy} - U_{xy}$

Output: It contains the parameters from *Step1* in Fig. 3: $ST(k)$, SC^{tct}, AoI^{cr}, F

1: Set $ST(k)= \emptyset$, $k \in \{1,2,...,M\}$
2: Set Cx_i=0
3: for i=1 to K **do**
4: **for** j=1 to d **do**
5: **if** $P_iP_j(k)=1$ **then** Calculate the initial Cx_i for each i-th swarm based on *Step2* **end if**
6: **end for**
7: $OptCx_i = Cx_i$
8: end for
9: for $iter$=1 to $MaxIter$ **do**
10: **for** i=1 to K **do**
11: Optimize the target positions of the i-th swarm for the $iter$-th iteration based on *Step3*
12: **if** $Cx_i < OptCx_i$ **then** $OptCx_i = Cx_i$ **end if**
13: **end for**
14: end for
15: for sim=1 to $MaxSim$ **do**
16: **for** k=*1* to M **do**
17: Determine the $ST(k)$ of each $U(k)$ for the sim-th simulation based on *Step4*
18: $Tf_{temp} = Tf_{temp} + Tf(k)$
19: **if** $P_iP_j(k)=1$ *and* $Tf_{temp} \le MaxTf$ **then** (that is, Eq. 14 and Eq. 15 are satisfied)
20: $ST(k) = ST(k) + \{OptT\}$
21: $Tf_{total}(k) = Tf_{total}(k) + Tf(k)$
22: **end if**
23: **end for**
24: Calculate SC^{tct} and AoI^{cr} using Eq. (7) and Eq. (10)
25: $F_{sim} = \alpha * SC^{tct} + (1-\alpha)/AoI^{cr}$
26: end for

Below is a detailed description of the four steps that make up WTSC-CSGA, which we highlight in Algorithm 2 and Fig. 3:

Step 1 First, we define the simulation parameters (Fig. 3) that we consider when solving the UAV-SC problem. Based on these parameters, we run Monte Carlo

simulations for both the operation of CSA and the solution of the UAV-SC problem. We use the parameters d, K, Cx and $OptCx$ in the modeling of CSA, while we use all other parameters in the UAV-SC modeling.

Step 2 Our goal in this step is to first calculate the cost function Cx for each K swarm. To this end, we first generate P target sets with random positions in the AoI modeled as a grid. We assign random weight values between 1 and 10 to each target in the P target sets to ensure that targets with high weights are represented as critical or important. We assign randomly selected targets from the P target sets to each d-dimension of each K swarm, where $d<N$. Then, for each d-dimensional target of each K swarm with the UAV take-off point as the P^0, we calculate the flight time of the UAV to these targets and the cost of these targets. We determine the d-dimensional target of the K swarm with minimal cost using GA and enable the UAV to follow this target and update this target position as the next position of the UAV. By repeating this process for each d-dimensional target of the K swarm, we determine the targets that the UAV will follow in sequence. Moreover, in this process, we apply the first constraint to be considered (Eq. 14) to the UAV-SC problem by ensuring that each d-dimensional target is tracked at most once by the UAV. At the end of this process, by calculating the cumulative total cost of the targets with minimum cost that the UAV follows sequentially, we determine the initial Cx of the K swarm and also determine the initial optimal cost function of the K swarm by assigning Cx to $OptCx$.

Step 3 Our goal in this step is to optimize the d-dimensional target locations of each K swarm based on the P target sets during iteration. To this end, we first use CSA to ensure that each K swarm converges to another randomly selected K swarm. We achieve this convergence by modeling the swarms to follow each other. Each time K swarms converge, new d-dimensional locations are created. However, these newly created positions are different from the positions in the P target sets. This difference poses a problem for UAV-SC, since any change of locations in the P target sets means that the originally randomly generated N targets can no longer be tracked by the UAVs. In the WTSC-CSGA approach that we have developed to solve this problem, we ensure that each K swarm converges to each other, while ensuring that the new d-dimensional locations converge to the existing target locations in the P target sets. At the end of this process, new d-dimensional targets are assigned to the converging K swarm based on the target positions in the P target sets. Therefore, the K swarm needs to be updated by recalculating its Cx as in *Step2*. Since the optimal solution of the K swarm needs to be found and stored in memory, in this step we compare the current Cx and OptCx of each K swarm in the iterations and thus store the optimal solution we have determined in memory. If in the corresponding iteration $Cx<$OptCx for K swarm, then K swarm follows the optimal costly targets and achieves a better solution, which we store in memory with OptCx. Otherwise, by not updating OptCx, we ensure that the current optimal solution for K swarm is maintained.

Step 4 Our goal in this step is to determine the targets that each of the multiple UAVs tracks with SC based on the d-dimensional targets of the swarms with the optimal cost we have determined, and thus optimize F by calculating the task completion time of the multiple UAVs and the SC ratio of the AoI. To this end,

we first determine the number of UAVs that will follow the P target sets and assign to the $OptS$ parameter the swarms with the optimal cost that we determine for each UAV. Since, as mentioned in *Step3*, we store the optimal solution for each K swarm with the OptCx parameter in memory, in this step we achieve K swarms with the optimal cost by determining the minimum of the OptCx computed for each K swarm. Considering the d-dimensional targets that are not swept by SC in the optimal K swarm represented by Opt$S$, and considering the P^0 as the starting point of the UAV's flight, we identify the target with the minimum cost among the d-dimensional targets and assign it to the Opt$T$ parameter. Furthermore, by calculating the UAV's flight time for the Opt$T$ target, we determine the Tf parameter and thus also calculate the Tf_{temp} parameter by considering the cumulative sums of Tf. Considering the maximum flight time of the UAV, which we defined as the second constraint (Eq. 15) in the UAV-SC problem, we decide whether the UAV will follow the Opt$T$ target with SC. That is, if $\text{Tf}_{\text{temp}} < MaxTf$, we determine that the UAV satisfies the MaxTf constraint for the Opt$T$ target and thus the UAV follows this target with SC by adding the Opt$T$ to the ST(k). Since the targets added to ST(k) are tracked by $U(k)$, the targets in $ST(k)$ cannot be tracked a second time by UAVs. Therefore, as mentioned in *Step2*, in this step, we apply the first constraint (Eq. 14) that we considered in the UAV-SC problem. As can be seen in *Step4* of Fig. 3, we calculate the cumulative flight time of $U(k)$ with Tf_{temp} to avoid adding the flight time of the Opt$T$ (Tf) to the cumulative flight time of $U(k)$ (Tf_{total}) when the second constraint is not met. To avoid incorrect calculation of the cumulative flight time of $U(k)$ when the second constraint is not met, we represent the cumulative flight time of $U(k)$ with Tf_{temp} in the second constraint. Once the second constraint is met, we update Tf_{total} and the next position of the UAV based on the Opt$T$ we last added to ST(k). Furthermore, if the second constraint is not met, it is no longer possible to track d-dimensional targets of Opt$S$ with SC, considering the remaining battery time of $U(k)$. In this case, $U(k)$ must complete the SC task and return to base. In this step, as long as the condition $U(k) \leq M$ is met, we calculate the task completion time (SC^{tct}) of multiple UAVs based on the Tf_{total} time of the UAV that last returned to the P^0, and the SC ratio of AoI (AoI^{cr}) based on the SC ratio of each UAV, thus determining F in each Monte Carlo simulation.

3.4 Time complexity analysis of WTSC-CSGA

In this section, we consider the running time of this approach to compute the time complexity of WTSC-CSGA, which we denote by the asymptotic big-O notation. We also compare the time complexity of WTSC-CSGA with WTSC [13], one of the algorithms we consider for performance analysis. Based on the flowchart in Fig. 3, we analyze the time complexity of WTSC-CSGA from 3 aspects: constant, linear and quadratic time:

3.4.1 Constant time

Considering the time complexity of expressions with constant time that do not depend on the number of inputs, we calculate the time complexity as O(1), since the running time of all arithmetic operations and comparisons performed between *Step2* and *Step4* of the WTSC-CSGA approach is the same. Expressions with constant time are therefore not considered in big-O.

3.4.2 Linear time

Considering the time complexity of linear time expressions, which increases in running time as n increases depending on the number of inputs n, we can calculate the time complexity as O(n), since we use a single loop in the cost computations between *Step2* and *Step4* in the WTSC-CSGA approach. Therefore, using multiple single loops with linear time has no effect on the O(n) time complexity.

3.4.3 Quadratic time

Considering the time complexity of quadratic-time expressions whose input number depends on n, we compute the time complexity as $O(n^2)$ since we use interdependent nested loops when computing Tf and Cx of each d-dimensional target for each K swarm between *Step2* and *Step4* in the WTSC-CSGA approach.

In Step2, Step3 and Step4 of the WTSC-CSGA approach, we use nested *for* or *while* loops that contain both constant-time and quadratic-time statements. Since constant-time statements have no effect on big-O, we calculate the running time of each step in the WTSC-CSGA approach as $T(n) = 2n^2$ and determine the time complexity as $O(n^2)$. Thus, if we consider all steps, we calculate the running time of the WTSC-CSGA approach as $T(n) = 6n^2$. Thus, if we consider that the polynomial degree has no effect on the time complexity, we find that the time complexity of the WTSC-CSGA approach is $O(n^2)$. The time complexity of WTSC [13], which we considered in the performance comparison, was calculated as $O(mn^2)$. Since m indicates that a process with linear time is repeated with a single loop, we can calculate the time complexity of WTSC as cubic time $O(n^3)$. Therefore, we find that the WTSC-CSGA approach with a time complexity of $O(n^2)$ is more optimal compared to the WTSC algorithm.

4 Experimental results and discussion

In this section, we conduct experimental studies by performing independent Monte Carlo simulations to evaluate the efficiency of the WTSC-CSGA developed to solve the UAV-SC problem. We use MATLAB R2017a simulator software to evaluate the performance of WTSC-CSGA. We compare the performance of WTSC-CSGA with the WTSC, CycleSplit and G-MSCR algorithms [13]. In developing these algorithms to solve the SC problem with multiple UAVs, WTSC [13] considers power

Table 2 The Monte Carlo simulation parameters

Parameters	Values
AoI	50 km × 50 km
N	50–250
d	50–250
M	5
K	100
V	72 km/h
w	0.1 rad/s
W_i	1–10
$T_{\max}$	90 min
T^s	8 min
α	0.2, 0.4, 0.6, 0.8
$\text{fl}^{i,\text{ iter}}$	1 m
$\text{AP}^{i,\text{ iter}}$	0.05
O	1
MaxIter	50
MaxSim	10
R	200 m

constraints and target weights of the UAVs. CycleSplit [57] considers the minimum period SC problem with multiple mobile sensors along a predefined trajectory. G-MSCR [58] considers the return time constraint with a minimum number of mobile sensors.

4.1 Simulation parameters

In the Monte Carlo simulations performed in this study, we use the parameters given in Table 2. An AoI with a size of 50 km × 50 km (2-D) is created. Grid modeling [10] is used in the AoI to ensure the placement of non-uniformly distributed targets. The grid is placed uniformly with a distance of 100 m between the points. In AoI, the placement of targets is random and the P target sets are set in increments of 50 between 50 and 250. The weight value W_i, which indicates the importance of the location for $\forall P_i$, is randomly assigned between 1 and 10. M is limited to 5 in the AoI. The defined values for MaxSim, the maximum number of Monte Carlo simulations, and MaxIter, the maximum number of iterations per simulation, are 10 and 50, respectively. The constant linear flight speed of the UAV in the AoI, V, is assumed to be 72 km/h, while the angular velocity, w, is 0.1 rad/s and the minimum turning radius is 200 m. In the simulations, it is assumed that the maximum flight time of the UAVs $T_{\max}$ is 90 min and that it requires 8 min for setup (preparation time T^s for flight) before take-off.

The equilibrium coefficient α for SC^{tct} and AoI^{cr} is set to 0.2, 0.4, 0.6 and 0.8, respectively. The number of operators O participating in the flight preparation of

Table 3 The UAV TCTs computed for $N=50$ and the $\forall\alpha$

α	Maximum (min.)	Minimum (min.)	Standard deviation	Mean (min.)
0.2	121.3716	100.9090	4.9977	111.0547
0.4	**111.3630**	**95.4348**	3.9910	**105.3130**
0.6	116.7366	101.3260	4.1839	108.4875
0.8	113.8342	103.3864	**3.3594**	107.4837

The values highlighted in bold show the optimum values from the simulations

Table 4 The optimal simulations where F is minimum for $N=50$ and the $\forall\alpha$

α	SC^{tct} (min.)	AoI^{cr} (%)	F
0.2	100.9090	100.0	20.9818
0.4	**95.4348**	100.0	**38.7739**
0.6	101.3260	100.0	61.1956
0.8	103.3864	100.0	82.9091

The values highlighted in bold show the optimum values from the simulations

each UAV is assumed to be 1. For the specified parameters within the CSA, we assume that the problem size d equals N, the swarm size K equals 100, the flight distance of the swarm $fl^{i,\,iter}$ equals 1 m, and the probability of awareness $AP^{j,\,iter}$ equals 0.05. Assuming that the UAV flies at the optimal altitude of 142 m, the detection radius R on the ground is calculated to be 200 m [10]. If $U(k)$ detects more than one target within the radius R during the flight according to P_i, it assumes that all these targets are swept by $U(k)$.

4.2 WTSC-CSGA performance evaluation

We experimentally investigate the effects of the system parameters in Table 2 on the performance of WTSC-CSGA using 5 different scenarios with 50, 100, 150, 200 and 250 targets. In each scenario, we run 10 Monte Carlo simulations with a maximum of 50 iterations for the parameter α, defined as 0.2, 0.4, 0.6 and 0.8, to optimize F to a minimum with WTSC-CSGA. We analyze the influence of the parameter α on F by calculating the maximum, minimum, mean and standard deviation of the metrics SC^{tct} and AoI^{cr} in these simulations. Thus, we determine the optimal simulation that minimizes F for the parameter that reaches the minimum SC^{tct} and the maximum AoI^{cr} in each scenario. In addition, we analyze the performance of WTSC-CSGA by computing $T_{total}(k)$, $T^c(k)$, the size of $ST(k)$ (the number of targets swept by $U(k)$), the ratio of targets swept to all targets, the total weight of $ST(k)$ and AoI^{cr} in the optimal simulation.

Table 5 Analyzing the optimal simulation where F is the minimum for $N = 50$ and $\alpha = 0.4$

UAV No	(1)**	(2)	(3)	(4)*	(5)
Waiting time (min.)	8	16	24	32	–
Flight time (min.)	87.4348	75.7243	70.3104	45.0137	–
Cumulative time (min.)	95.4348	91.7243	94.3104	77.0137	–
Number of targets swept	20	18	6	6	–
Ratio of total targets swept (%)	100.0				
Total weight in targets swept	121	85	44	31	–
Total weight in all targets	281				
Area coverage ratio (%)	100.0				

*Shows the first UAV number to return to base after the SC task

**Shows the last UAV number

Table 6 The UAV TCTs computed for $N = 100$ and the $\forall\alpha$

α	Maximum (min.)	Minimum (min.)	Standard deviation	Mean (min.)
0.2	128.9459	114.0819	4.9172	122.4668
0.4	129.8060	**109.6192**	5.1962	121.9437
0.6	129.7208	113.0259	4.5874	**120.2056**
0.8	125.0133	115.5127	**2.8971**	125.0133

The values highlighted in bold show the optimum values from the simulations

4.2.1 Study of scenario 1

In this scenario, based on 50 randomly distributed targets, the TCT comparison calculated by WTSC-CSGA is shown in Table 3. The *Maximum*, *Minimum*, *Standard Deviation* and *Mean* of the TCT calculated in the simulations for the parameter $\forall\alpha$ are obtained. The *Standard Deviation* is used to evaluate the stability of the TCT. The simulations show that the parameter α with a value of 0.4 is a decisive parameter for the optimization of this scenario, as a minimum TCT of 95 min is achieved. Therefore, reaching the optimum at the *Maximum*, *Minimum* and *Mean* TCT for $\alpha = 0.4$ is indeed a proof that optimal simulations are performed for this parameter.

If we look at the simulations in Table 4, where F is optimized to the minimum based on SC^{tct} and AoI^{cr}, we see that the UAVs sweep 100% of the AoI in all simulations for $\forall\alpha$. In addition, a simulation for $\alpha = 0.4$ is achieved where F is optimized with a minimum SC^{tct} of 95 min and a maximum AoI^{cr} of 100%. Thus, if we consider the optimization of F with 40% SC^{tct} and 60% AoI^{cr} to the minimum for $\forall\alpha$ separately, we find that $\alpha = 0.4$ is a significant parameter in the optimization process.

In this scenario, the simulation in which F is optimized to a minimum for $\alpha = 0.4$ is analyzed and shown in Table 5. Although the number of multiple UAVs defined in the simulations is 5, the SC is performed with only 4 UAVs and thus all targets are swept by 4 UAVs. If we consider $T_{max} = 90$ as a constraint, we see that the first 32

Table 7 The coverage rates of AoI computed for $N = 100$ and the $\forall\alpha$

α	Maximum (%)	Minimum (%)	Standard deviation	Mean (%)
0.2	100.00	92.24	0.0211	98.28
0.4	**100.00**	96.02	0.0130	**99.21**
0.6	**100.00**	97.53	0.0085	**99.55**
0.8	100.00	96.41	0.0127	98.89

The values highlighted in bold show the optimum values from the simulations

Table 8 The optimal simulations where F is minimum for $N = 100$ and the $\forall\alpha$

α	SC^{tct} (min.)	AoI^{cr} (%)	F
0.2	114.0819	92.24	23.6887
0.4	**109.6192**	**100.0**	**44.4477**
0.6	113.0259	**100.0**	68.2156
0.8	115.5127	97.48	92.6154

The values highlighted in bold show the optimum values from the simulations

min of the cumulative flight time of 77.01 min of the first UAV (UAV No. 4) returning to base is spent on flight preparation (24-min waiting time for the launch of the other three UAVs + 8-min setup time for the launch of the fourth UAV), and the next 45.01 min is used for the flight to a total of 6 targets. It is determined that the last UAV (UAV No. 1) returning to base has completed the SC task by flying over 20 targets during a cumulative flight time of 95.43 min. It is determined that the total weight (281) of targets swept by the 4 UAVs during the SC equals the total weight of all targets in the AoI, resulting in a 100% coverage rate for the AoI.

4.2.2 Study of scenario 2

In this scenario, the TCT comparisons calculated by WTSC-CSGA with $\forall\alpha$ and 100 randomly distributed targets are shown in Table 6. Achieving the *Minimum* TCT for $\alpha = 0.4$, despite calculating the *Mean* TCT, which is minimum for $\alpha = 0.6$, proves that an optimal simulation is performed. Since a minimum TCT of 109 min is achieved in the simulations, it shows that $\alpha = 0.4$ is an important parameter in the optimization in this scenario.

Comparing the *Maximum*, *Minimum*, *Standard Deviation* and *Mean* of the AoI^{cr} calculated for $\forall\alpha$ in Table 7, we see that the optimal values for both the *Maximum* and *Mean* of the AoI^{cr} are obtained for the values 0.4 and 0.6 of the parameter α, respectively. We also identified one or more simulations in which the UAVs flew over all targets with these parameters α, covering the AoI by a maximum of 100%.

Table 9 Analyzing the optimal simulation where F is the minimum for $N = 100$ and $\alpha = 0.4$

UAV No	(1)	(2)	(3)**	(4)	(5)*
Waiting time (min.)	8	16	24	32	40
Flight time (min.)	88.9267	87.6509	85.6192	76.9817	21.8016
Cumulative time (min.)	96.9267	103.6509	109.6192	108.9817	61.8016
Number of targets swept	34	24	22	16	4
Ratio of total targets swept (%)	100.0				
Total weight in targets swept	208	138	130	91	24
Total weight in all targets	591				
Area coverage ratio (%)	100.0				

*Shows the first UAV number to return to base after the SC task

**Shows the last UAV number

Table 10 The UAV TCTs computed for $N = 150$ and the $\forall\alpha$

α	Maximum (min.)	Minimum (min.)	Standard deviation	Mean (min.)
0.2	129.6531	121.9840	2.4083	**126.5984**
0.4	129.9908	**120.4676**	2.6716	127.6688
0.6	129.4937	121.5344	2.2454	127.0378
0.8	129.9385	**120.5467**	2.6092	127.8090

The values highlighted in bold show the optimum values from the simulations

Table 11 The coverage rates of AoI computed for $N = 150$ and the $\forall\alpha$

α	Maximum (%)	Minimum (%)	Standard deviation	Mean (%)
0.2	92.47	80.22	0.0312	88.56
0.4	93.72	87.45	0.0241	**90.25**
0.6	96.15	81.54	0.0519	**90.03**
0.8	**98.01**	83.58	0.0557	89.96

The values highlighted in bold show the optimum values from the simulations

Therefore, the parameter α proves to be crucial in simulations with values of 0.4 and 0.6, as the *Mean* AoI^{cr} is achieved at these values α with the highest rates.

Looking at the simulations where F is optimized to the minimum based on the SC^{tct} and AoI^{cr} for $\forall\alpha$ in Table 8, a simulation for $\alpha = 0.4$ achieves both the minimum SC^{tct} of 109.61 min and the maximum AoI^{cr} of 100.0%. Thus, $\alpha = 0.4$ proves to be a decisive parameter in the optimization of F to the minimum with 40% SC^{tct} and 60% AoI^{cr}.

In this scenario, the simulation in which F is optimized to a minimum for $\alpha = 0.4$ is analyzed and shown in Table 9. In all simulations, the SC task is completed by

Table 12 The optimal simulations where F is minimal for $N = 150$ and the $\forall\alpha$

α	SC^{tct} (min.)	AoI^{cr} (%)	F
0.2	121.9840	85.85	25.3375
0.4	**120.4676**	87.45	48.8731
0.6	121.5344	**95.52**	**73.3394**
0.8	**120.5467**	93.60	96.6510

The values highlighted in bold show the optimum values from the simulations

Table 13 Analyzing the optimal simulation where F is the minimum for $N = 150$ and $\alpha = 0.6$

UAV No	(1)*	(2)	(3)	(4)**	(5)
Waiting time (min.)	8	16	24	32	40
Flight time (min.)	87.6217	86.9435	86.5568	89.5344	70.1386
Cumulative time (min.)	95.6217	102.9435	110.5568	121.5344	110.1386
Number of targets swept	36	31	39	21	16
Ratio of total targets swept (%)	95.33				
Total weight in targets swept	160	204	225	139	82
Total weight in all targets	848				
Area coverage ratio (%)	95.52				

*Shows the first UAV number to return to base after the SC task

**Shows the last UAV number

flying all 5 UAVs. Considering the constraint $T_{max} = 90$, we see that the first 40 min of the cumulative flight time of 61.80 min of the first UAV (UAV No. 5) returning to base is used for flight preparation (32-min waiting time for the launch of the other four UAVs + 8-min setup time for the launch of the fifth UAV), and the next 21.80 min is used for the flight to a total of 4 targets. The last UAV to return to base (UAV No. 3) flew over 22 targets during its 85.61-min flight and completed the SC task in a total of 109.61 min. During the SC task, the total weight of targets overflown by each of the multiple UAVs is 591, which is equal to the total weight of all targets in the AoI, resulting in a 100% coverage rate for the AoI.

4.2.3 Study of scenario 3

The TCT comparisons in this scenario, calculated by WTSC-CSGA based on $\forall\alpha$ for 150 randomly distributed targets, are shown in Table 10. Although only the optimal *Mean* TCT value is achieved for $\alpha = 0.2$, the optimal *Minimum* TCT value is also achieved for $\alpha = 0.4$ and $\alpha = 0.8$. Although the number of simulations with the optimal TCT for $\alpha = 0.2$ is larger, the calculation of the *Minimum* TCT for $\alpha = 0.4$ and $\alpha = 0.8$ only showed that the minimum TCT for these parameters α is actually achieved in two simulations. Thus, reaching 120 min in the minimum TCT optimization proved that the values α of 0.4 and 0.8 are decisive parameters.

Table 14 The UAV TCTs computed for $N = 200$ and the $\forall\alpha$

α	Maximum (min.)	Minimum (min.)	Standard deviation	Mean (min.)
0.2	129.8440	**122.6535**	2.2573	**126.9506**
0.4	129.9416	**122.0447**	2.5428	127.5357
0.6	129.9080	123.8220	**1.7025**	127.8427
0.8	129.8940	**122.3443**	2.3971	127.4670

The values highlighted in bold show the optimum values from the simulations

Table 15 The coverage rates of AoI computed for $N = 200$ and the $\forall\alpha$

α	Maximum (%)	Minimum (%)	Standard deviation	Mean (%)
0.2	**87.29**	67.26	0.0533	78.16
0.4	86.23	76.35	0.0295	80.35
0.6	**87.36**	71.39	0.0390	**81.09**
0.8	85.49	75.04	0.0282	81.06

The values highlighted in bold show the optimum values from the simulations

Table 16 The optimal simulations where F is minimal for $N = 200$ and the $\forall\alpha$

α	SC^{tct} (min.)	AoI^{cr} (%)	F
0.2	123.5024	**87.29**	25.6169
0.4	**122.0447**	86.23	49.5137
0.6	123.8220	77.54	74.8091
0.8	**122.3443**	79.19	98.1279

The values highlighted in bold show the optimum values from the simulations

When analyzing the *Maximum*, *Minimum*, *Standard Deviation* and *Mean* AoI^{cr} value for $\forall\alpha$ in Table 11, we find that even when calculating the optimal *Mean* AoI^{cr} value for $\alpha = 0.4$ and $\alpha = 0.6$, only the *Maximum* AoI^{cr} value for $\alpha = 0.8$ is achieved, which is an optimal simulation for this parameter. Thus, $\alpha = 0.8$ proves to be a crucial parameter for optimizing the AoI^{cr} to the maximum.

Looking at the simulations in which the minimum SC^{tct} and maximum AoI^{cr} for $\forall\alpha$ are shown in Table 12, it is clear that there are several simulations in which F was optimized to a minimum. The minimum SC^{tct} is achieved for $\alpha = 0.4$ and $\alpha = 0.8$, while the maximum AoI^{cr} is achieved for $\alpha = 0.6$. The SC^{tct} for $\alpha = 0.4$ and $\alpha = 0.8$ seem to converge, while only the AoI^{cr} for $\alpha = 0.6$ is clearly recognized as optimal at 95.52%. Thus, it is obvious that $\alpha = 0.6$ plays a crucial role in the minimization of F, which is calculated with 60% of the minimum SC^{tct} and 40% of the maximum AoI^{cr}.

Table 17 Analyzing the optimal simulation where F is the minimum for $N=200$ and $\alpha = 0.2$

UAV No	(1)*	(2)	(3)	(4)	(5)**
Waiting time (min.)	8	16	24	32	40
Flight time (min.)	89.1048	89.3212	88.3340	86.5278	83.5024
Cumulative time (min.)	97.1048	105.3212	112.3340	118.5278	123.5024
Number of targets swept	37	40	34	27	21
Ratio of total targets swept (%)	79.50				
Total weight in targets swept	242	217	222	153	107
Total weight in all targets	1078				
Area coverage ratio (%)	87.29				

*Shows the first UAV number to return to base after the SC task

**Shows the last UAV number

Table 18 Analyzing the optimal simulation where F is the minimum for $N=200$ and $\alpha = 0.4$

UAV No	(1)*	(2)	(3)	(4)	(5)**
Waiting time (min.)	8	16	24	32	40
Flight time (min.)	89.6008	87.8365	89.8439	85.9521	82.0447
Cumulative time (min.)	97.6008	103.8365	113.8439	117.9521	122.0447
Number of targets swept	42	33	35	31	24
Ratio of total targets flown (%)	82.50				
Total weight in targets swept	265	171	146	164	112
Total weight in all targets	995				
Area coverage ratio (%)	86.23				

*Shows the first UAV number to return to base after the SC task

**Shows the last UAV number

In this scenario, the simulation in which F is optimized to a minimum for $\alpha = 0.6$ is analyzed and shown in Table 13. The SC task is performed in the simulations with all 5 UAVs. Considering the $T_{\max} = 90$ constraint, it is found that the first UAV (UAV No. 1) returning to base performs flight preparation in the first 8 min of the cumulative flight time of 95.62 min and flies to 36 targets in the remaining 87.62 min. It is noted that the last UAV to arrive at the base (UAV No. 4) flies 21 targets in its 89.53-min flight, giving a cumulative time of 121.53 min for the SC task. It is found that the total weight of targets swept by the 5 UAVs during the SC is 810, which corresponds to a coverage of 95.52% of the AoI. If we look at the number of targets swept by the 5 UAVs, we find that 143 targets or 95.33% of all targets in the AoI are covered by the UAVs.

4.2.4 Study of scenario 4

In this scenario, the TCT comparisons computed by WTSC-CSGA based on $\forall\alpha$ for 200 randomly distributed targets are shown in Table 14. Although $\alpha = 0.4$ and

Table 19 The UAV TCTs computed for $N=250$ and the $\forall\alpha$

α	Maximum (min.)	Minimum (min.)	Standard deviation	Mean (min.)
0.2	129.9057	125.6071	1.2050	128.6567
0.4	129.9271	**120.7597**	2.4655	**127.2179**
0.6	129.9794	124.3186	1.3756	128.5122
0.8	129.6166	121.6618	2.2810	**127.8527**

The values highlighted in bold show the optimum values from the simulations

Table 20 The coverage rates of AoI computed for $N=250$ and the $\forall\alpha$

α	Maximum (%)	Minimum (%)	Standard deviation	Mean (%)
0.2	76.61	64.50	0.0359	71.48
0.4	**78.61**	66.22	0.0331	71.20
0.6	76.71	66.96	0.0285	**72.22**
0.8	74.60	68.09	0.0184	**72.09**

The values highlighted in bold show the optimum values from the simulations

$\alpha = 0.8$ only achieved the optimal *Minimum* TCT, $\alpha = 0.2$ achieved both the optimal *Minimum* TCT and the optimal *Mean* TCT. This shows that optimal simulations were performed for $\alpha = 0.2$. Thus, $\alpha = 0.2$ proves to be a decisive parameter in optimizing the TCT in the SC to a minimum.

A comparison of the *Maximum*, *Minimum*, *Standard Deviation* and *Mean* AoI^{cr} computed for $\forall\alpha$ in Table 15 shows that the optimal simulation is achieved only at the *Maximum* AoI^{cr} for $\alpha = 0.2$, while for $\alpha = 0.6$ the optimal simulations are achieved at both the *Maximum* and the *Mean* AoI^{cr}. It can therefore be observed that the optimum values are more likely to be achieved in the simulations for $\alpha = 0.6$ and that this parameter is decisive for optimizing the AoI^{cr} to the maximum.

When examining the simulations in which the minimum SC^{tct} and the maximum AoI^{cr} for $\forall\alpha$ were determined in Table 16, two simulations are identified in which F was optimized to the minimum for $\alpha = 0.2$ and $\alpha = 0.4$. The minimum SC^{tct} is reached in 122 min for $\alpha = 0.4$ and $\alpha = 0.8$, while the maximum AoI^{cr} is achieved with 87.29% for $\alpha = 0.2$. Thus, considering the convergence of these metrics for $\forall\alpha$, it is concluded that $\alpha = 0.2$ and $\alpha = 0.4$ are crucial in optimizing F to the minimum.

The simulation that minimizes F for $\alpha = 0.2$ is analyzed and shown in Table 17. Taking the $T_{\max} = 90$ constraint into account, it turns out that the last UAV (UAV No. 5) that returned to base flew over 21 targets during its 83.50-min flight, completing the SC task in a total of 123.50 min. The total number of targets overflown by 5 UAVs during the SC is 941, which corresponds to a coverage of 87.29% of the AoI. If we also look at the number of targets swept, it becomes clear that the UAVs overflew 159 targets, which corresponds to 79.5% of the 200 targets in the AoI.

Table 21 Optimal simulations where F is minimum for $N=250$ and the $\forall\alpha$

α	SC^{tct} (min.)	AoI^{cr} (%)	F
0.2	125.6071	71.76	26.2362
0.4	**120.7597**	**72.59**	49.1305
0.6	124.3186	68.39	75.1760
0.8	121.6618	68.09	97.6232

The values highlighted in bold show the optimum values from the simulations

Table 22 Analyzing the optimal simulation where F is the minimum for $N=250$ and $\alpha = 0.4$

UAV No	(1)*	(2)	(3)	(4)**	(5)
Waiting time (min.)	8	16	24	32	40
Flight time (min.)	89.7033	89.0885	88.7619	88.7597	80.0402
Cumulative time (min.)	97.7033	105.0885	112.7619	120.7597	120.0402
Number of targets swept	46	38	27	32	33
Ratio of total targets swept (%)	70.40				
Total weight in targets swept	289	240	148	182	203
Total weight in all targets	1463				
Area coverage ratio (%)	72.59				

*Shows the first UAV number to return to base after the SC task

**Shows the last UAV number

The simulation analysis for optimizing F to a minimum for $\alpha = 0.4$ is shown in Table 18. Considering the $T_{max} = 90$ constraint, it can be seen that the last UAV (UAV No. 5) that returned to base flew over 24 targets during its 82.04-min flight and completed the SC task in a total time of 122.04 min. The 5 UAVs flew over targets with a total weight of 858 during the SC, which corresponds to an AoI coverage of 86.23%. If we also look at the number of targets overflown by the 5 UAVs, 165 targets were swept, which is 82.50% of the 200 targets in the AoI.

4.2.5 Study of scenario 5

In this scenario, the TCT comparisons computed for WTSC-CSGA with 250 randomly distributed targets are shown in Table 19. Although only the optimal *Mean* TCT for $\alpha = 0.8$ is achieved, the calculation of both *Minimum* and the *Mean* TCT for $\alpha = 0.4$ shows that a larger number of optimal simulations have been achieved. Thus, achieving a minimum of 120 min in the optimization of TCT is evidence that $\alpha = 0.4$ is a crucial parameter.

The *Maximum*, *Minimum*, *Standard Deviation* and *Mean* AoI^{cr} computed for $\forall\alpha$ are shown in Table 20. Although the optimal *Mean* AoI^{cr} is achieved for $\alpha = 0.6$ and $\alpha = 0.8$, the calculation of optimal *Maximum* AoI^{cr} for $\alpha = 0.4$ shows the

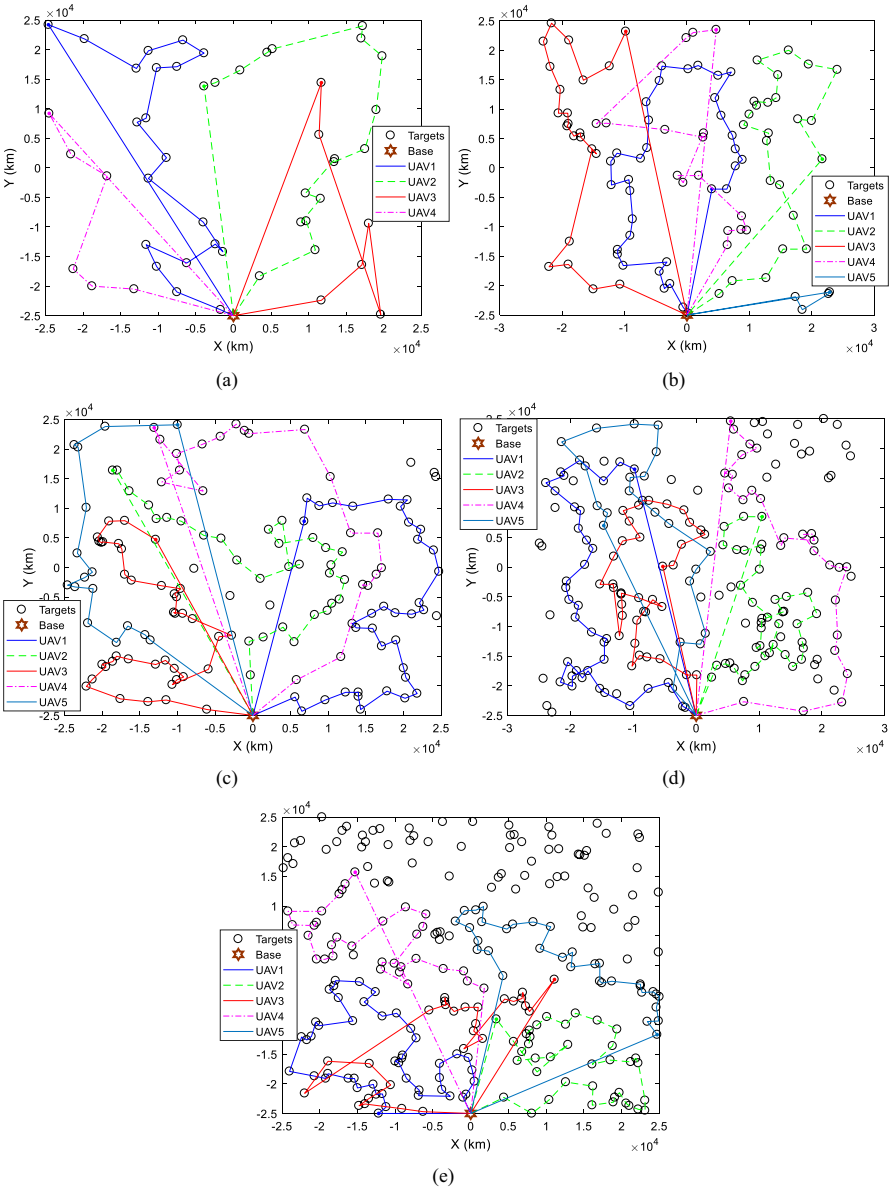


Fig. 4 CPPs of multiple UAVs by WTSC-CSGA for **a** $\alpha = 0.4$ in *Scenario 1*, **b** $\alpha = 0.4$ in *Scenario 2*, **c** $\alpha = 0.6$ in *Scenario 3*, **d** $\alpha = 0.2$ in *Scenario 4*, **e** $\alpha = 0.4$ in *Scenario 5*

achievement of optimal simulation. It can be concluded that $\alpha = 0.4$ is a crucial parameter in optimizing AoI^{cr} to the maximum.

When examining the simulations in Table 21, where F is optimized based on the minimum SC^{tct} and maximum AoI^{cr} for $\forall\alpha$, it is found that F is optimized to the minimum with an SC^{tct} of 120 min and a AoI^{cr} of 72.59% in $\alpha = 0.4$. It can be

Table 23 Simulation parameters for WTSC, CycleSplit and G-MSCR

Parameters	Values
AoI	50 km × 50 km
N	50–250
M	5
α	0.1, 0.3, 0.5, 0.7, 0.9
W_i	1–10
V	72 km/h
w	0.1 rad/s
$T_{\max}$	90 min
T^s	8 min

concluded that $\alpha = 0.4$ is a crucial parameter when optimizing F based on 40% SC^{tct} and 60% AoI^{cr} to the minimum.

We analyze the simulation that optimizes F to a minimum for $\alpha = 0.4$, as shown in Table 22. Considering the $T_{\max} = 90$ constraint, it can be seen that the first UAV (UAV No. 1) that returned to base flew over 46 targets during its 89.70-min flight, completing the SC task in a total of 97.70 min. The last UAV to return to base (UAV No. 4) flew over 32 targets during its 88.75-min flight, completing the SC task in a total of 120.75 min. It can be noted that the total weight of targets swept by the 5 UAVs during the SC is 1062, covering 72.59% of the AoI. If we also look at the number of targets swept by the UAVs, we can see that a total of 176 targets were swept, representing 70.4% of all targets in the AoI.

4.2.6 CPP of the scenarios

In the simulations performed with the α parameters to optimize F to the minimum in each scenario study, the CPPs of ST(k) determined for $\forall U(k), k \in \{1, 2, ..., M\}$ are shown in the diagrams in Fig. 4. The paths planned for the $\forall U(k)$ are shown as different colored linear lines in each of the diagrams, while targets that are unevenly distributed by the UAVs are shown as circles. In addition, the last target swept by each UAV is shown in these diagrams with a filled circle in the color of the respective UAV.

For the CPPs of the targets in the AoI, we find from Monte Carlo simulations that the SC task is completed based on 4 UAVs in *Scenario 1* and 5 UAVs in the other scenarios. In this simulation, we found that in *Scenario 1* (Fig. 4a) all 50 targets for $\alpha = 0.4$ which we determined to be optimal and in *Scenario 2* (Fig. 4b) all 100 targets for $\alpha = 0.4$ which we determined to be optimal were swept, resulting in an SC rate of %100 in both scenarios. In *Scenario 3* (Fig. 4c), for $\alpha = 0.6$, which we determined to be optimal, we found that by sweeping 143 out of 150 targets, an SC rate of 95.52% was achieved, which corresponds to the weight sum of the targets. In *Scenario 4* (Fig. 4d), for $\alpha = 0.2$, which we determined to be optimal, we found that by sweeping 200 out of 159 targets, an SC rate of 87.29% was achieved, which corresponds to the weight sum of the targets. In *Scenario 5* (Fig. 4e), for $\alpha = 0.4$, which

Table 24 Comparison of SC^{tct} and WTSC-CSGA performance ratio in the scenarios based on the number of targets

N	SC^{tct}(min.)				Performance ratio (%)		
	WTSC-CSGA	WTSC	CycleSplit	G-MSCR	WTSC	CycleSplit	G-MSCR
50	**95.43**	110	118	122	13.24	19.12	21.77
100	**109.61**	114	130	120	3.85	15.68	8.65
150	120.46	**113**	126	128	-6.60	4.39	5.89
200	122.04	**114**	126	130	-7.05	3.14	6.12
250	120.75	**114**	121	129	-5.92	0.20	6.39

The values highlighted in bold show the optimum values from the simulations

Table 25 Comparison of AoI^{cr} and WTSC-CSGA performance ratio in the scenarios based on the number of targets

N	AoI^{cr}(%)				Performance ratio (%)		
	WTSC-CSGA	WTSC	CycleSplit	G-MSCR	WTSC	CycleSplit	G-MSCR
50	**100.00**	**100.00**	**100.00**	96.00	0	0	4.16
100	**100.00**	91.00	84.00	88.00	9.89	19.04	13.63
150	**95.52**	74.00	70.00	69.00	29.08	36.45	38.43
200	**87.29**	65.00	58.00	64.00	34.29	50.50	36.39
250	**72.59**	62.00	50.00	59.00	17.08	45.18	23.03

The values highlighted in bold show the optimum values from the simulations

we determined to be optimal, we found that by sweeping 250 out of 176 targets, an SC rate of 72.59% was achieved, which corresponds to the weight sum of the targets. Therefore, as can be seen in these diagrams, we determine the CPP of the targets followed by each UAV launched asynchronously by a single operator in the SC task with optimal simulations.

4.3 Performance comparison

In this section, we compare WTSC-CSGA with the WTSC, CycleSplit and G-MSCR algorithms considering the SC^{tct} and AoI^{cr} metrics obtained in the simulations of the optimum achieved by WTSC-CSGA in each scenario. In this comparison, we present the parameters [13] used for the simulations of WTSC, CycleSplit and G-MSCR in Table 23. In the parameters we consider, the AoI is defined as a square area of 50 km$\times$50 km, the targets are randomly distributed in the AoI and the number of targets varies in steps of 50 between 50 and 250. In addition, α, which is used as a tuning parameter for the WTSC algorithm, is set between 0.1 and 0.9 and each target is assigned weight values between 1 and 10, which are randomly distributed in the AoI. We note that the number of asynchronously flying UAVs in AoI is defined as 5,

each of the UAVs flies at a constant linear velocity of 72 km/h during the SC task, the angular velocity of the UAVs is 0.1 rad/s, the maximum flight duration of the UAVs is 90 min, and the pre-flight setup time for each UAV is 8 min. In this paper, based on these basic parameters in Table 23, we ensure that both the characteristics of the environment in which the targets are distributed and the flight parameters of multiple UAVs are the same, thus perform similar modeling for WTSC-CSGA.

Tables 24 and 25 show the comparison of the performance ratio of WTSC-CSGA compared to the other algorithms along with the SC^{tct} and AoI^{cr} changes of the algorithms. In the performance comparison, we analyze the effect of changing the number of targets in AoI on SC^{tct} and AoI^{cr}.

In all simulations performed by WTSC-CSGA based on $\forall\alpha$ for $N=50$ (see Table 5), the SC task was completed with only 4 UAVs on all targets with a success rate of 100%. However, in comparison, the other algorithms [13] required 5 UAVs to complete the SC task. According to Table 24, the SC^{tct} calculated with 4 UAVs for WTSC-CSGA is 95.43 min. This corresponds to a minimum improvement of 13.24% and a maximum improvement of 21.77% compared to the other algorithms. Looking at the flight time of multiple UAVs from a theoretical perspective, it becomes clear that the fewer UAVs are used in the AoI, the greater the number of targets each UAV must fly over to accomplish the SC task. This leads to an increase in the duration of UAV flights for each target. However, the WTSC-CSGA algorithm achieves a lower SC^{tct} compared to the other algorithms, although it uses fewer UAVs in *Scenario 1*. Thus, it is clear that WTSC-CSGA provides better performance considering the SC^{tct} achieved in *Scenario 1*.

Based on *Scenario 2* with $N=100$, the optimal simulation of WTSC-CSGA achieved 109 min for SC^{tct}, resulting in a performance improvement of 3.85% to 15.68% compared to other algorithms. Furthermore, a AoI^{cr} of 100% was achieved in the optimal simulation by ensuring that all targets were swept by 5 UAVs, resulting in a performance improvement of 9.89% to 19.04% compared to other algorithms.

Based on *Scenario 3* with $N=150$, the optimal simulation with WTSC-CSGA reached 120 min for SC^{tct}. A 6.60% lower performance was achieved compared to the WTSC algorithm, while a minimum performance of 4.39% and a maximum performance of 5.89% were achieved compared to the other algorithms. Considering the AoI^{cr} of 95.52% in the optimal simulation, a performance improvement of 29.08% to 38.43% was achieved compared to the other algorithms.

Based on *Scenario 4* with $N=200$, the SC^{tct} in the optimal simulation of WTSC-CSGA was achieved in 122 min. Compared to the WTSC algorithm, the performance was 7.05% lower, while compared to the other algorithms, a performance improvement between 3.14 and 6.12% was achieved. Considering the AoI^{cr} of 87.29% in the optimal simulation, a performance improvement of 34.29% to 50.5% was achieved compared to the other algorithms.

Based on *Scenario 5* with $N=250$, the optimal simulation of WTSC-CSGA reached 120 min for the SC^{tct}. Compared to the WTSC algorithm, the performance decreased by 5.92%. However, compared to the other algorithms, there was an improvement in performance between 0.20 and 6.39%. Considering the AoI^{cr} of 72.59% in the optimal simulation, a performance improvement between 17.08% and 45.18% was achieved compared to the other algorithms..

When varying the number of targets between 150 and 250, it is noted that the SC^{tct} performance of the WTSC-CSGA is lower compared to the WTSC algorithm. However, it is clear that the AoI^{cr} performance is higher than that of the WTSC algorithm, showing an improvement between 17.08% and 34.29%. This is due to the fact that the WTSC algorithm performs the SC task for fewer targets, resulting in a lower SC^{tct} and therefore a lower AoI^{cr}. In addition, the CycleSplit and G-MSCR algorithms have both high SC^{tct} and low AoI^{cr} compared to WTSC-CSGA. Thus, the higher SC^{tct} value of WTSC-CSGA compared to WTSC proves that the SC task is actually performed effectively by multiple UAVs on more targets, resulting in a higher AoI^{cr} value.

4.4 Impact of UAV Energy Consumption on the SC problem

In this section, we partially explore the impact of UAV energy consumption on the SC problem. When we consider the cumulative flight time $T_{\text{total}}(k)$ of each kth UAV at the targets where the SC task is performed, we see that one of the parameters that determines the energy consumption of multiple UAVs is the SC^{tct}, which is calculated with the maximum $T_{\text{total}}(k)$. The energy consumption of multiple UAVs changes in direct proportion to the parameter SC^{tct}. The minimum flight time of each UAV at the targets where each UAV performs the SC task proves that all UAVs will return to the P^0 in the minimum $T_{\text{total}}(k)$ and thus the SC^{tct} will also be minimum. As can be seen from Table 24, the minimum of SC^{tct} in the simulations of N between 150 and 250 in the WTSC algorithm does not ensure that AoI^{cr} (Table 25) is optimal. However, the maximum of AoI^{cr} in the simulations of N between 50 and 250 in our WTSC-CSGA approach does not always require the minimum of SC^{tct} (except for N between 50 and 100). This is because a lower SC^{tct} indicates that multiple UAVs perform SCs on fewer targets, and therefore a lower AoI^{cr} is calculated. Thus, optimizing SC^{tct} to a minimum does not lead to optimizing AoI^{cr} to a maximum, even if it optimizes the UAV's energy consumption to a minimum. Therefore, we cannot linearly investigate the impact of the UAV's energy consumption on the UAV-SC problem.

5 Conclusion

In this paper, we present a novel solution to the SC problem with asynchronous flying multiple UAVs based on the offline path planning method. Our approach is to develop a hybrid of metaheuristic CSA and heuristic GA. In this approach, we develop a novel model that optimizes F to an efficient CPP in AoI based on the metrics of minimum SC^{tct} and maximum AoI^{cr}. The performance results obtained from the scenario studies based on the parameter N show a significant improvement of up to 21% and 50% for SC^{tct} and AoI^{cr}, respectively. In *Scenario 1*, only 4 UAVs are used to complete the SC task on 100% of the targets in the AoI by WTSC-CSGA. The result shows that the targets are swept with fewer UAVs compared to the other algorithms. Looking at the performance of SC^{tct} in each scenario, the reason for the higher performance of WTSC

between *Scenario 3* and *Scenario 5* is that it performs the SC task on fewer targets and thus achieves lower SC^{tct}. However, since WTSC-CSGA performed the SC tasks on more targets between *Scenario 2* and *Scenario 5*, it achieved a relatively higher SC^{tct} compared to WTSC and thus an optimal AoI^{cr}. The performance results show that the proposed WTSC-CSGA algorithm outperforms the WTSC, CycleSplit and G-MSCR algorithms in completing the SC task for targets in the AoI and achieves an effective CPP.

In this study, we only focus on offline path planning without considering location changes within the AoI, including environment and targets. Therefore, when solving the SC problem, we use multiple asynchronously flying UAVs to track the targets. In future research, online path planning that considers real-time AoI and target updates will be investigated as a new strategy. We are also planning the scope of future research as follows:

- We aim to solve the UAV-SC problem with the improved Crow Search Algorithm based on multiple UAVs flying simultaneously and contribute to the literature by achieving better performance than our proposed approach.
- We aim to contribute to the literature by comparing the performance of different swarm-based metaheuristic algorithms for solving the UAV-SC problem considering multiple synchronously and asynchronously flying UAVs.

Author contributions As a single author, the entire manuscript was written by the author.

Data availability No datasets were generated or analyzed during the current study.

Declarations

Conflict of interest The authors declare no competing interests.

References

1. Hasan Ali N, Özdağ R (2022) Coverage analysis and a new metaheuristic approach using the Elfes Probabilistic detection model in wireless sensor networks. Measurement 200:111627. https://doi.org/10.1016/j.measurement.2022.111627
2. Liu C, Du H (2021) t, K -sweep coverage with mobile sensor nodes in wireless sensor networks. IEEE Internet Things J 8(18):13888–13899. https://doi.org/10.1109/JIOT.2021.3070062
3. Gao X, Chen Z, Pan J, Wu F, Chen G (2020) Energy efficient scheduling algorithms for sweep coverage in mobile sensor networks. IEEE Trans on Mobile Comput 19(6):1332–1345. https://doi.org/10.1109/TMC.2019.2910074
4. J. Wubben, J. P. Matos-Carvalho, D. Pedro, S. Tomic, and C. T. Calafate (2024), Empirical evaluation of multi UAV coverage path planning for aerial surveying. In: Presented at the 20th International Conference on Distributed Computing in Smart Systems and the Internet of Things (DCOSS-IoT), Abu Dhabi, United Arab Emirates: IEEE
5. Lin C, Han G, Xu T, Martínez-García M (2021) Energy-optimal data collection for unmanned aerial vehicle-aided ındustrial wireless sensor network-based agricultural monitoring system: a clustering compressed sampling approach. IEEE Trans Ind Inf 17(6):4411–4420

6. Xie R, Meng Z, Wang L, Li H, Wang K, Wu Z (2021) Unmanned aerial vehicle path planning algorithm based on deep reinforcement learning in large-scale and dynamic environments. IEEE Access 9:24884–24900
7. Wen H (2024) Route planning for UAVs maritime search and rescue considering the targets moving situation. Ocean Eng 310:118623
8. Harikumar K, Senthilnath J, Sundaram S (2019) Multi-UAV oxyrrhis marina-inspired search and dynamic formation control for forest firefighting. IEEE Trans Automat Sci Eng 16(2):863–873. https://doi.org/10.1109/TASE.2018.2867614
9. Filist S et al (2024) An unmanned aerial vehicle autonomous flight trajectory planning method and algorithm for the early detection of the ignition source during fire monitoring. Int J Remote Sens 12(45):4178–4197. https://doi.org/10.1080/01431161.2024.2358451
10. Ozdag R (2022) Multi-metric optimization with a new metaheuristic approach developed for 3D deployment of multiple drone-BSs. Peer-to-Peer Netw Appl 15(3):1535–1561. https://doi.org/10.1007/s12083-022-01298-4
11. Liu Y, Xie J, Xing C, Xie S, Luo X (2024) Self-organization of UAV networks for maximizing minimum throughput of ground users. IEEE Trans Veh Technol 73(8):11743–11755. https://doi.org/10.1109/TVT.2024.3369020
12. Amhaz A, Elhattab M, Sharafeddine S, Assi C (2024) UAV-assisted cooperative downlink NOMA: deployment and resource allocation. IEEE Trans Veh Technol 73(9):1–14. https://doi.org/10.1109/TVT.2024.3386839
13. Li J, Xiong Y, She J, Wu M (2020) A path planning method for sweep coverage with multiple UAVs. IEEE Internet Things J 7(9):8967–8978. https://doi.org/10.1109/JIOT.2020.2999083
14. Gharehchopogh FS, Abdollahzadeh B (2022) An efficient harris hawk optimization algorithm for solving the travelling salesman problem. Cluster Comput 25(3):1981–2005. https://doi.org/10.1007/s10586-021-03304-5
15. Hu W et al (2023) Multi-UAV coverage path planning: a distributed online cooperation method. IEEE Trans Veh Technol 72(9):1–14. https://doi.org/10.1109/TVT.2023.3266817
16. Wang H, Du H (2022) Time sensitive sweep coverage with minimum UAVs. Theoret Comput Sci 928:197–209. https://doi.org/10.1016/j.tcs.2022.06.025
17. Tang J, Liang Y, Li K (2024) Dynamic scene path planning of uavs based on deep reinforcement learning. Drones 8(2):60. https://doi.org/10.3390/drones8020060
18. Qadir Z, Zafar MH, Moosavi SKR, Le KN, Mahmud MAP (2022) Autonomous UAV path-planning optimization using metaheuristic approach for predisaster assessment. IEEE Internet Things J 9(14):12505–12514. https://doi.org/10.1109/JIOT.2021.3137331
19. Popescu D, Dragana C, Stoican F, Ichim L, Stamatescu G (2018) A collaborative UAV-WSN network for monitoring large areas. Sensors 18(12):4202. https://doi.org/10.3390/s18124202
20. Tutsoy O, Asadi D, Ahmadi K, Nabavi-Chashmi SY, Iqbal J (2024) Minimum distance and minimum time optimal path planning with bioinspired machine learning algorithms for faulty unmanned air vehicles. IEEE Trans Intell Transport Syst 25(8):9069–9077. https://doi.org/10.1109/TITS.2024.3367769
21. Ahmadi SM, Kebriaei H, Moradi H (2018) Constrained coverage path planning: evolutionary and classical approaches. Robotica 36(6):904–924. https://doi.org/10.1017/S0263574718000139
22. Aggarwal S, Kumar N (2020) Path planning techniques for unmanned aerial vehicles: a review, solutions, and challenges. Comput Commun 149:270–299. https://doi.org/10.1016/j.comcom.2019.10.014
23. Datsko D, Nekovar F, Penicka R, Saska M (2024) Energy-aware multi-UAV coverage mission planning with optimal speed of flight. IEEE Robot Autom Lett 9(3):2893–2900. https://doi.org/10.1109/LRA.2024.3358581
24. D. Zhang, Y. Xu, and X. Yao (2018) An Improved Path Planning Algorithm for Unmanned Aerial Vehicle Based on RRT-Connect. In: 2018 37th Chinese Control Conference (CCC), Wuhan: IEEE, pp 4854–4858. https://doi.org/10.23919/ChiCC.2018.8483405
25. Liu M, Zhang H, Yang J, Zhang T, Zhang C, Bo L (2024) A path planning algorithm for three-dimensional collision avoidance based on potential field and B-spline boundary curve. Aerosp Sci Technol 144:108763. https://doi.org/10.1016/j.ast.2023.108763
26. Kim M-J, Kang TY, Ryoo C-K (2024) Real-time path planning for unmanned aerial vehicles based on compensated voronoi diagram. Int J Aeronaut Space Sci. https://doi.org/10.1007/s42405-024-00771-z

27. J. Liu, X. Wang, B. Bai, and H. Dai (2018) Age-optimal trajectory planning for UAV-assisted data collection. In: IEEE INFOCOM 2018 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), Honolulu, HI: IEEE, pp 553–558. https://doi.org/10.1109/INFCOMW.2018.8406973
28. Athira KA, Yalavarthi R, Saisandeep T, Harshith KS, Sha A (2024) ACO-DTSP algorithm: optimizing UAV swarm routes with workload constraints. Proc Comput Sci 235:163–172. https://doi.org/10.1016/j.procs.2024.04.019
29. Perazzo P, Sorbelli FB, Conti M, Dini G, Pinotti CM (2017) Drone path planning for secure positioning and secure position verification. IEEE Trans on Mobile Comput 16(9):2478–2493. https://doi.org/10.1109/TMC.2016.2627552
30. Y. Zhang, Y. Zhang, Z. Liu, Z. Yu, and Y. Qu (2018) Line-of-Sight Path Following Control on UAV with Sideslip Estimation and Compensation. In: 2018 37th Chinese Control Conference (CCC), Wuhan: IEEE, pp 4711–4716. https://doi.org/10.23919/ChiCC.2018.8483606
31. J. De Waen, H. T. Dinh, M. H. Cruz Torres, and T. Holvoet (2017) Scalable multirotor UAV trajectory planning using mixed integer linear programming. In: 2017 European Conference on Mobile Robots (ECMR), Paris: IEEE, pp 1–6. https://doi.org/10.1109/ECMR.2017.8098706
32. Kyriakis P, Moustris G (2019) Terrain following for fixed-wing unmanned aerial vehicles using feedback equivalence. IEEE Control Syst Lett 3(1):150–155. https://doi.org/10.1109/LCSYS.2018.2854239
33. Pehlivanoglu YV, Pehlivanoglu P (2021) An enhanced genetic algorithm for path planning of autonomous UAV in target coverage problems. Appl Soft Comput 112:107796. https://doi.org/10.1016/j.asoc.2021.107796
34. Yu Z, Si Z, Li X, Wang D, Song H (2022) A novel hybrid particle swarm optimization algorithm for path planning of UAVs. IEEE Internet Things J 9(22):22547–22558. https://doi.org/10.1109/JIOT.2022.3182798
35. X. Li and J. Chen (2017) An Efficient Framework for Target Search with Cooperative UAVs in a FANET. In: 2017 IEEE International Symposium on Parallel and Distributed Processing with Applications and 2017 IEEE International Conference on Ubiquitous Computing and Communications (ISPA/IUCC), Guangzhou: IEEE, pp 306–313. https://doi.org/10.1109/ISPA/IUCC.2017.00051
36. Yang Q, Yoo S-J (2018) Optimal UAV path planning: sensing data acquisition over IoT sensor networks using multi-objective bio-inspired algorithms. IEEE Access 6:13671–13684. https://doi.org/10.1109/ACCESS.2018.2812896
37. Aljalaud F, Kurdi H, Youcef-Toumi K (2023) Bio-inspired multi-UAV path planning heuristics: a review. Mathematics 11(10):2356. https://doi.org/10.3390/math11102356
38. Wu Z, Li J, Zuo J, Li S (2018) Path planning of UAVs based on collision probability and kalman filter. IEEE Access 6:34237–34245. https://doi.org/10.1109/ACCESS.2018.2817648
39. Farmani N, Sun L, Pack DJ (2017) A scalable multitarget tracking system for cooperative unmanned aerial vehicles. IEEE Trans Aerosp Electron Syst 53(4):1947–1961. https://doi.org/10.1109/TAES.2017.2677746
40. Puente-Castro A, Rivero D, Pazos A, Fernandez-Blanco E (2022) UAV swarm path planning with reinforcement learning for field prospecting. Appl Intell 52(12):14101–14118. https://doi.org/10.1007/s10489-022-03254-4
41. Zeng Y, Xu X, Zhang R (2018) Trajectory design for completion time minimization in UAV-enabled multicasting. IEEE Trans Wireless Commun 17(4):2233–2246. https://doi.org/10.1109/TWC.2018.2790401
42. Z. Luo, Z. Liu, J. Shi, Q. Wang, T. Zhou, and Y. Liu (2018) The Mathematical Modeling of the Two-Echelon Ground Vehicle and Its Mounted Unmanned Aerial Vehicle Cooperated Routing Problem. In: 2018 IEEE Intelligent Vehicles Symposium (IV), Changshu: IEEE, pp 1163–1170. https://doi.org/10.1109/IVS.2018.8500391
43. Qu C, Gai W, Zhang J, Zhong M (2020) A novel hybrid grey wolf optimizer algorithm for unmanned aerial vehicle (UAV) path planning. Knowl-Based Syst 194:105530. https://doi.org/10.1016/j.knosys.2020.105530
44. Dasdemir E, Köksalan M, Tezcaner Öztürk D (2020) A flexible reference point-based multi-objective evolutionary algorithm: an application to the UAV route planning problem. Comput Op Res 114:104811. https://doi.org/10.1016/j.cor.2019.104811

45. Yu X, Li C, Yen GG (2021) A knee-guided differential evolution algorithm for unmanned aerial vehicle path planning in disaster management. Appl Soft Comput 98:106857. https://doi.org/10.1016/j.asoc.2020.106857
46. C. Ntakolia, K. S. Platanitis, G. P. Kladis, C. Skliros, and A. D. Zagorianos (2022) A Genetic Algorithm enhanced with Fuzzy-Logic for multi-objective Unmanned Aircraft Vehicle path planning missions. In: 2022 International Conference on Unmanned Aircraft Systems (ICUAS), Dubrovnik, Croatia: IEEE, pp 114–123. https://doi.org/10.1109/ICUAS54217.2022.9836068
47. Nie Z, Du H (2021) An approximation algorithm for general energy restricted sweep coverage problem. Theoret Comput Sci 864:70–79. https://doi.org/10.1016/j.tcs.2021.02.028
48. Liang W, Zhang Z (2022) Approximation algorithm for prize-collecting sweep cover with base stations. Theoret Comput Sci 929:1–10. https://doi.org/10.1016/j.tcs.2022.06.026
49. Liang W, Zhang Z, Du D-Z (2023) A unified approach to approximate partial, prize-collecting, and budgeted sweep cover problems. Optim Lett. https://doi.org/10.1007/s11590-023-02008-6
50. Liang D, Feng B, Liao X (2024) A path planning method for chargeable sweep coverage with multiple charging stations. IEEE Access 12:34931–34941. https://doi.org/10.1109/ACCESS.2024.3373543
51. Gao X, Fan J, Wu F, Chen G (2022) Cooperative sweep coverage problem with mobile sensors. IEEE Trans Mobile Comput 21(2):480–494. https://doi.org/10.1109/TMC.2020.3008348
52. Jia Y et al (2022) The UAV path coverage algorithm based on the greedy strategy and ant colony optimization. Electronics 11(17):2667. https://doi.org/10.3390/electronics11172667
53. Askarzadeh A (2016) A novel metaheuristic method for solving constrained engineering optimization problems: crow search algorithm. Comput Struct 169:1–12. https://doi.org/10.1016/j.compstruc.2016.03.001
54. Xiang D, Lin H, Ouyang J, Huang D (2022) Combined improved A* and greedy algorithm for path planning of multi-objective mobile robot. Sci Rep 12(1):13273. https://doi.org/10.1038/s41598-022-17684-0
55. Castello Rosa ADF, Pereira FH (2024) An intensification approach based on fitness landscape characteristics for job shop scheduling problem. J Comb Optim 47(5):77. https://doi.org/10.1007/s10878-024-01176-0
56. Lian J, Wang P, Li G (2023) A greedy algorithm based compensation circuit for optimizing the output statistics of APUF. Microelectron J 131:105636. https://doi.org/10.1016/j.mejo.2022.105636
57. Gao X, Fan J, Wu F, Chen G (2018) Approximation algorithms for sweep coverage problem with multiple mobile sensors. IEEE/ACM Trans Netw 26(2):990–1003. https://doi.org/10.1109/TNET.2018.2815630
58. C. Liu, H. Du, and Q. Ye(2016) Sweep Coverage with Return Time Constraint. In: 2016 IEEE Global Communications Conference (GLOBECOM), Washington, DC, USA: IEEE, pp 1–6. https://doi.org/10.1109/GLOCOM.2016.7842310

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.