Journal of Combinatorial Optimization (2023) 45:132
https://doi.org/10.1007/s10878-023-01045-2

Maximizing the amount of data collected from WSN based on solar-powered UAV in urban environment

Chuanwen Luo[1,2] · Junzhe Hu[1,2] · Yunan Hou[1,2] · Yi Hong[1,2] · Yuqing Zhu[3] · Deying Li[4]

Accepted: 8 May 2023 / Published online: 10 July 2023

Abstract

Unmanned Aerial Vehicle (UAV) plays an increasingly role in data collection from Wireless Sensor Networks (WSNs) with the advantages of its high mobility and flexibility. However, the energy limitation of UAV restricts its application for data collection tasks. To solve the problem, we install solar panel on UAV to acquire energy from sunlight. This paper studies Data Collection Maximization based on Solar-powered UAV (DCMS) problem in urban environment with lots of obstacles, where one UAV equipped with solar panel is used to collect data from WSN. The problem aims at optimizing the flight trajectory of UAV such that the amount of data collected from WSN is maximized. We prove that the problem is NP-hard. To solve the DCMS problem, we first propose three algorithms: Bypass Obstacles during Flight Algorithm (BOFA), Auxiliary Graph Flight Path (AGFP), Construct Flight Plan in data collection Area (CFPA). Their objectives are to bypass the obstacles, to obtain the flight path connecting all data collection areas in WSN, to optimize the flight trajectories of UAV in the data collection areas, respectively. Afterwards, we propose an approximation algorithm called DCMSA to solve the DCMS problem based on BOFA, AGFP, CFPA algorithms. Finally, the proposed algorithm is verified by extensive simulations.

Keywords Wireless Sensor Networks · Solar-powered UAV · Data collection · Trajectory optimization

A preliminary version Luo et al. (2022) of this paper appeared in International Conference on Algorithmic Aspects in Information and Management (AAIM 2022).

✉ Chuanwen Luo
chuanwenluo@bjfu.edu.cn

✉ Yi Hong
hongyi@bjfu.edu.cn

Extended author information available on the last page of the article

1 Introduction

Wireless Sensor Networks(WSNs) are self-organizing, multi-hops distributed networks, where sensors are deployed in the detection area to sense the environment information (Luo et al. 2021b). They have a large of applications, such as environment monitoring, disaster monitoring, intelligent transportation and so on. In traditional WSNs, the data stored in the sensors are transmitted back to the base station through multi-hop routing, which depends on the battery energy carried by sensors. However, since WSNs produce a amount of data as they are widely used in production and life, the traditional way of collecting data will greatly consume the energy of WSNs and reduce their service life. To overcome the above shortcomings, that Unmanned Aerial Vehicles (UAVs) are used as data collectors to collect massive data in WSNs can not be restricted by various ground conditions with the advantages of fast flight speed, flexible flight routes and strong transportation capabilities.

Although UAVs have many advantages for data gathering in WSNs, their battery capacity limitation are considered as the crucial technical challenges for UAVs, and which may make them unable to complete the data collection tasks in many applications. In order to extend the working time of UAVs, many wireless charging technologies are applied to replenish energy for UAVs in much literature, such as solar charging (Thipyopas et al. 2019; Alsharoa et al. 2019), radio frequency (RF) technology (Li et al. 2017), laser charging technology (L. Company 2012; Lahmeri et al. 2020) and so on. However, since laser transmitters and RF chargers are needed to deploy at specific locations in advance, UAVs need to fly at these locations for replenishing energy, which will reduce the efficiency of UAVs and consume a large amount of electric energy. Solar power as nature source can replenish energy for UAVs without energy supply equipment and does not require additional energy and human and material resources. Therefore, in this paper, we consider to use solar power to replenish energy for UAVs when the UAVs are used to collect data from WSN.

However, since the charging efficiency of solar energy is low and the energy limitation initially carried by UAV, we can not gather all data from WSN with the given solar-powered UAV. Specifically, there are many obstacles to present obstructing the communication between the UAV and sensors in urban environment. Therefore, in this paper, we study the Data Collection Maximization based on Solar-powered UAV(DCMS) problem, where the solar-powered UAV is used to collect data from WSN deployed urban environment. In the problem, we not only consider the situation that solar power is used to supplement energy for UAV, but also consider to avoid obstacles when the UAV collects data from sensors during flight. The objective of the problem aims at optimizing the flight trajectory of the solar-powered UAV to maximize the amount of data from sensors in urban environment before the energy of the UAV is exhausted. The contribution of this paper can be summarized as below.

(1) We identify a data collection model of WSN based on solar-powered UAV by considering avoiding obstacles in urban environment, which is called Data Collection Maximization based on Solar-powered UAV(DCMS) problem. Then we prove the problem is NP-hard.

(2) To solve the DCMS problem, we first prove that the optimal horizontal flight speed is a function with respect to the flight altitude to minimize the net energy consumption of UAV by considering solar charging. Then we propose three algorithms: Bypass Obstacles during Flight Algorithm(BOFA), Auxiliary Graph Flight Path (AGFP), Construct Flight Plan in data collection Area (CFPA) to bypass the obstacles, to obtain the flight path connecting all data collection areas in WSN, to optimize the flight trajectories of UAV in the data collection areas, respectively. Afterwards, we propose an approximation algorithm to solve the DCMS problem based on the above algorithms.

(3) The extensive simulations are presented to illustrate the effectiveness of the proposed algorithm for the DCMS problem.

The remainder of this paper is organized as follows. Section 2 introduces related works. Section 3 introduces models and the problem definition. In Sect. 4, we propose an approximation algorithm to solve the DCMS problem. Simulations are shown in Sect. 5. Section 6 concludes this paper.

2 Related works

This section will introduce the relevant research status and put forward the differences of the problems studied in this paper. We classify the study problems into three different types: data collection based on UAV, researches on solar-powered UAV, obstacle avoidance of UAV.

2.1 Data collection based on UAV

In Gong et al. (2018), Gong et al. studied the time minimization problem of UAV by considering both flying and communication of UAV. However, they only investigated the scenario that the UAV collects data from the sensors on a stright line, which is rare in reality and has many limitations. In Liu et al. (2018), the authors designed the flight paths for single UAV and multiple UAVs to maximize the capacity of sensors, but they assumed that the flight paths are fixed. However, in the actual scenario, the UAVs with variable paths will play a higher efficiency in communication in the WSN. In Luo et al. (2020), Luo et al. investigated the maximizing data collection proportion problem to find the trajectory of UAV such that the minimum data collection proportion of collected data to the stored data among all sensors is maximized. In Luo et al. (2021a), Luo et al. designed detailed flight and hover plans for multiple UAVs for data gathering from the WSN. They minimized the maximum flight time of UAVs such that all data in the WSN is collected by the UAVs and transported to the base station. In Sun et al. (2022), Sun et al. presented a new solar-powered fixed-wing UAV-assisted data collection technique, where a fixed-wing UAV harvests solar energy to fly and collect data from smart devices. They optimized the UAV's three-dimensional trajectory to maxmize the minimum of the data upload-ed from any of the smart devices.

2.2 Researches on solar-powered UAV

In Kingry et al. (2018), Kingry et al. presented a prototype quadcopter UAV, which carries a PV cell array. And they reported that the UAV can stay airborne for one to 2 h by harvesting solar power. In Thipyopas et al. (2019), the authors developed a mall solar-powered Unmanned Aerial Vehicle for environmental monitoring application and aimed to achieve continuous flight endurance of 6 h. The results showed that hybrid solar powered UAV weight of 5.5 kg is predicted for 6-h non-stop flight operation from 9AM to 3PM under Thailand weather condition. In Fu et al. (2021), Fu et al. investigated a solar-powered Unmanned Aerial Vehicle system, where UAV collects data from Internet of Things Devices (IoTDs) on the ground and the three-dimensional trajectory is optimized to maximize the total residual energy of the UAV. In Cong et al. (2021), Cong et al. considered a general UAV-enabled wireless communication system, where the fixed-wing UAV with thin-film solar cells is deployed to provide continuous communication services for the ground users.

2.3 Obstacles avoidance of UAV

In Li et al. (2022), Li et al. proposed a dynamic obstacle avoidance path planning strategy for UAV. They optimized the obstacle avoidance effect of the UAV by by changing the UAV turning radius, changing the UAV heading, solving the UAV minimum deviation distance, reducing the UAV obstacle avoidance space. In Zhou et al. (2022), Zhou et al. proposed a trajectory planning scheme and realized the unity of obstacle avoidance and trajectory planning, where the UAV does not deviate from the route after obstacle avoidance, and returns to the scheduled route nearby.

It is clear from previous discussions that there are many researches on UAV-based data collection in WSN and solar-powered UAV and UAV flying in some environment with obstacles. However, few people consider data collection based on solar-powered UAV in WSN in urban environment with obstacles. Inspired by above literatures, in this paper, we study the the data collection maximization problem based on solar-powered UAV by considering obstacles in urban environment, in which we aim to maximize the data collection volume and enable the solar-powered UAV to return to the base station before it runs out of energy. We not only consider the obstacle avoidence method, but also optimize the trajectory of UAV for data gathering from WSN.

3 Models and definition

In this section, we give the models and definition for the problem.

3.1 Network model

In this paper, we consider a WSN deployed at urban environment where many obstacles are located in. For simplicity, we assume that n sensors and m obstacles with known location and size are deployed at a two-dimensional plane area $\mathcal{A} \subseteq \Re^2$. Let $S =$

$\{s_1, s_2, \cdots, s_n\}$ denote the set of sensors, where each sensor $s_i \in S$ stores V_i units of data. We use $O = \{o_1, o_2 \ldots, o_m\}$ to denote the set of the m obstacles in which each obstacle is shaped as a cube. We use a solar-powered UAV u with source node s_0, initial energy E, vertical flight speed v_l, horizontal flying speed v_f and minimum flying altitude H to serve as a mobile collector for gathering data from sensors in WSN, where E is also the energy capacity of UAV.

We use (x_i^s, y_i^s) to represent the coordinates for any $s_i \in S \cup \{s_0\}$. For arbitrary $o_j \in O$, let $p_j^o = (x_j^o, y_j^o, z_j^o)$ denote its centre point and (l_j^o, w_j^o, h_j^o) represent the size of o_j, where l_j^o represents the length of o_j, w_j^o denotes the width of o_j and h_j^o is the height of o_j. As we all known, the UAV can't fly close to the boundary of the obstacle. Therefore, we set a fixed buffer distance d_j^o between UAV and o_j when the UAV meets any $o_j \in O$.

Assume that all sensors have the same communication radius R. For any $s_i \in S$, we let $\Omega(s_i)$ denote the communication area of s_i. The sensor s_i can transmit data to UAV if and only if the UAV is in $\Omega(s_i)$. The data collection area of UAV is a circular area $C(s_i')$ whose radius and center are respectively $R_c = \sqrt{R^2 - H^2}$ and s_i' when u flies at the altitude H. Let $\mathcal{C} = \{C(s_1'), C(s_2'), \cdots, C(s_n')\}$. The UAV can collect data from sensors if and only if there does not exist obstacle between them. If $H \leq h_j^o + d_j^o$, then the UAV must bypass o_j from other three directions: right side, left side and upward side. For simplicity, we assume that the UAV only can fly horizontally and vertically. For any pair of $s_i \in S$ and $s_j \in S$, $\Omega(s_i)$ and $\Omega(s_j)$ are disjoint with each other.

3.2 Communication model

Only when the UAV is located in $\Omega(s_i)$ and there is no obstacles between the UAV and s_i can the UAV collect data from s_i. Therefore, in this paper, we adopt the Free Space Path Loss (FSPL) model between UAV and sensors. Based on Gong et al. (2018) Luo et al. (2021a), the data transmission rate from s_i to u can be described as

$$C_u(s_i) = \begin{cases} \frac{1}{2} W \log_2(1 + \frac{\gamma_0 P_w}{d^\alpha(s_i, u)}), & \text{LoS} \\ 0, & \text{NLoS} \end{cases} \tag{1}$$

where W denotes the channel bandwidth, $d(s_i, u)$ denotes the Euclidean distance between s_i and u, α is the path loss exponent and $2 \leq \alpha < 4$, P_w represents the data transmission power of s_i, $\gamma_0 = \frac{\beta_0}{\sigma^2}$ denotes the reference signal-to-noise ratio in which β_0 represents the channel gain at a reference distance $d_0 = 1m$ and σ^2 is the noise variance.

3.3 Solar energy harvesting model

In this paper, we adopt the solar energy harvesting model in Fu et al. (2021). Ignoring the influence of atmosphere, the power of UAV collected from sunlight when it flies at altitude h can be expressed as

$$P_c(h) = \eta_s A_s G_s(\alpha_s - \beta_s e^{\frac{-h}{\delta_s}}), \tag{2}$$

where η_s represents the energy conversion efficiency, A_s denotes the area of the solar panel, G_s is the average solar radiation, α_s denotes the maximum value of atmospheric transmittance, β_s denotes the atmospheric extinction coefficient, and δ_s is the scale height of the earth.

According to the Eq. (2), we can obtain that the higher the UAV flies, the more power it can charge from the sunlight.

3.4 Propulsion power consumption model

According to the propulsion power consumption model for rotating wing UAV proposed in Wang et al. (2019), we can obtain the consumption power of UAV is a function of flight speed v, and it can be expressed as

$$P(v) = P_0\left(1 + \frac{3v^2}{v_r^2}\right) + P_1\left(\sqrt{1 + \frac{v^4}{4v_0^4}} - \frac{v^2}{2v_0^2}\right)^{\frac{1}{2}} + \frac{d_0 \rho s A v^3}{2}, \tag{3}$$

where P_0 and P_1 are two constants representing the blade profile power and induced power in hovering status, v_r denotes the tip speed of the rotor, v_0 is the mean rotor induced velocity in hover, d_0 represents the fuselage drag ratio, s is the rotor robustness, and ρ denotes the air density in units of kg/m^3, A is rotor disk area.

Based on proof in Wu et al. (2020), we have the equation (3) is convex, i.e., there exists an optimal speed v^* to minimize the value of $P(v)$. And the value of v^* is given as

$$v^* = arg \min_{v \geq 0} P(v), \tag{4}$$

3.5 Definition for the problem

In this subsection, we give the detailed definition of the Data Collection Maximization based on Solar-powered UAV (DCMS) problem as shown in Definition 1, whose objective is to maximize the amount of data collected from WSN by solar-powered UAV with limited initial energy.

In the Definition 1, we use $\Phi(U, Q, T, D_f)$ to denote the feasible flight plan of UAV such that a part of data in WSN are collected and transported to the data center, where U represents the flight tour of UAV, Q denotes the set of hovering points of UAV to collect data from sensors, T is the set of hovering time of UAV at the hovering points in Q, and D_f is the amount of data collected from sensors when the UAV is flying on U.

Definition 1 (**DCMS**) Given a set $S = \{s_1, s_2, \ldots, s_n\}$ of n sensors in which each s_i stores V_i units of data, a base station s_0, a set $O = \{o_1, o_2 \ldots, o_m\}$ of m obstacles with known location and size, a solar-powered UAV u with initial energy E, horizontal flight speed v_f that varies with flight altitude h, vertical flight speed v_l, minimum flight altitude H for horizontal flying, the Data Collection Maximization based on Solar-powered UAV (DCMS) problem aims at finding a flight plan $\Phi(U, Q, T, D_f)$ such that

(1) U starts from and ends at s_0,
(2) the UAV can collect data from s_i when it is within $\Omega(s_i)$ and there is no obstacles between it and s_i,
(3) at any given moment, the energy of the UAV is greater than 0 and less than or equal to E,
(4) for any $o_j \in O$, if o_j is taller than H, then the UAV will by pass it from above or left or right,
(5) for any hovering point $hp_i \in Q$, the UAV collects data from s_i with $t_i^h \in T$ time,
(6) the amount of data collected by the UAV, $D = \Sigma_{hp_i \in Q} t_i^h \cdot C_u(s_i) + D_f$ is maximized.

Formally, problem DCMS can be formulated as

$$\max \ \Sigma_{hp_i \in Q} \frac{1}{2} t_i^h W \log_2 \left(1 + \frac{\gamma_0 P_w}{d^\alpha(s_i, hp_i)}\right) + D_f \tag{5}$$

s.t

$$0 \leq E_e^r \leq E \tag{6}$$

Theorem 1 *The DCMS problem is NP-hard.*

Proof If we set $R = 0$, $E = +\infty$, $V_i = 0$ for each sensor $s_i \in S$, $H = 0$ and $h_j^o = 0$ for any $o_j \in O$, then the DCMS problem can be reduced to the the well-known Traveling Salesman Problem (TSP) since the UAV only needs to visit all sensors for collecting data. Since the TSP problem is a special case of the DCMS problem and the TSP problem has been shown to be NP-hard (?), the DCMS problem is also NP-hard. □

Theorem 2 *Let L and E_{net}^L represent the horizontal flight distance and the net energy consumption flying on L of UAV, respectively. There exists an optimal horizontal flight speed v_f^* to minimize E_{net}^L which is a function with respect to the flight altitude h, i.e., $v_f^* = \varphi(h)$.*

Proof According to the definition, we can obtain

$$E_{net}^L = P(v_f)\frac{L}{v_f} - P_c(h)\frac{L}{v_f}, \tag{7}$$

By substituting $P(v_f)$ and $P_c(h)$ into equation (7), we have

$$E_{net}^L = \left(P_0 \left(1 + \frac{3v_f^2}{v_r^2} \right) + P_1 \left(\sqrt{1 + \frac{v_f^4}{4v_0^4}} - \frac{v_f^2}{2v_0^2} \right)^{\frac{1}{2}} \right. \\ \left. + \frac{d_0 \rho s A v_f^3}{2} - \eta_s A_s G_s \left(\alpha_s - \beta_s e^{\frac{-h}{\delta_s}} \right) \right) \frac{L}{v_f}, \tag{8}$$

Based on Wu et al. (2020), the Eq. (8) can be reduced to

$$E_{net}^L = \left(P_0 \left(1 + \frac{3v_f^2}{v_r^2} \right) + P_1 \frac{v_0}{v_f} + \frac{d_0 \rho s A v_f^3}{2} \right. \\ \left. - \eta_s A_s G_s \left(\alpha_s - \beta_s e^{\frac{-h}{\delta_s}} \right) \right) \frac{L}{v_f}, \tag{9}$$

The first derivative of E_{net}^L with respect to v_f is

$$(E_{net}^L)' = \frac{L}{v_f^2} \left(\frac{3P_0 v_f^2}{v_r^2} - \frac{2P_1 v_0}{v_f} + A d_0 \rho s v_f^3 \right. \\ \left. - P_0 + \eta_s A_s G_s \left(\alpha_s - \beta_s e^{\frac{-h}{\delta_s}} \right) \right), \tag{10}$$

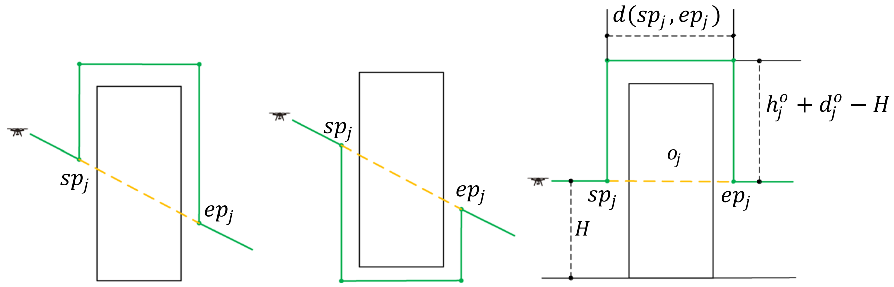
Let $g(v_f) = \frac{3P_0 v_f^2}{v_r^2} - \frac{2P_1 v_0}{v_f} + A d_0 \rho s v_f^3 - P_0 + \eta_s A_s G_s (\alpha_s - \beta_s e^{\frac{-h}{\delta_s}})$, the first derivative of $g(v_f)$ with respect to v_f is

$$g'(v_f) = \frac{6P_0 v_f}{v_r^2} + \frac{2P_1 v_0}{v_f^2} + 3A d_0 \rho s v_f^2, \tag{11}$$

Obviously, when $v_f > 0$, $g'(v_f) > 0$. Therefore, $g(v_f)$ increases strictly monotonically over the interval $(0, +\infty)$. As v_f goes to 0, $g(v_f)$ goes to $-\infty$, and as v_f goes to $+\infty$, $g(v_f)$ goes to $+\infty$. Thus, $g(v_f)$ has a unique zero-point v_f^* on the interval $(0, +\infty)$ that makes $g(v_f)$ minimum. Therefore, E_{net}^L is minimum when $v_f = v_f^*$. Therefore, we can obtain the following function

$$h = \psi(v_f^*) = -\delta_s \ln \left(\frac{\alpha_s}{\beta_s} + \frac{1}{\eta_s A_s G_s \beta_s} \left(\frac{3P_0 (v_f^*)^2}{v_r^2} - \frac{2P_1 v_0}{(v_f^*)} + A d_0 \rho s (v_f^*)^3 - P_0 \right) \right), \tag{12}$$

Let $\chi(v_f) = \frac{3P_0 v_f^2}{v_r^2} - \frac{2P_1 v_0}{v_f} + A d_0 \rho s v_f^3 - P_0$, then $\chi(v_f)$ and $g(v_f)$ have the same monotonicity, and we can rewrite Eq. (12) as



(a) Bypass o_j on the left (b) Bypass o_j on the right (c) Bypass o_j on the upward

Fig. 1 Schematic diagram of obstacle o_j bypassing methods of UAV

$$h = \psi(v_f^*) = -\delta_s \ln\left(\frac{\alpha_s}{\beta_s} + \frac{\chi(v_f)}{\eta_s A_s G_s \beta_s}\right), \tag{13}$$

It is easy to observe that the Eq. (13) is strictly monotonically decreasing in the interval $(0, +\infty)$. According to the existence theorem of the inverse function, we have

$$v_f^* = \varphi(h) = \psi^{-1}(h). \tag{14}$$

□

Therefore, in the following proposed algorithm, we let $v_f = v_f^*$ to minimize the net energy consumption of UAV when the UAV flies at a fixed altitude.

4 Research methods

In this section, we propose an approximation algorithm to solve the DCMS problem. The algorithm consists of four phases.

The first phase to design a algorithm to bypass obstacles. The second phase is first to construct the auxiliary graph based on the given network model. Then we compute an flight path U_c of UAV based on the auxiliary graph to connect all data collection areas in WSN when its energy is enough. Afterwards, for any $s_i \in S$, we compute the two interconnect points st_i and ed_i between U_c and $\Omega(s_i)$. In the third phase, we design the flight trajectory $\Upsilon(U_i, hp_i, t_i^h)$ of UAV in $\Omega(s_i)$ for any $s_i \in S$ such that all data of s_i is collected, where U_i is a path from st_i to ed_i. Based on the first three phases, we can obtain initial flight plan $\Phi(U, Q, T, D_f)$ when the energy of UAV is enough, where $U = \bigcup_{s_i \in S} U_i \cup U_c$.

In the fourth phase, we first compute the energy consumption E_{net} of UAV when the UAV execute the flight plan $\Phi(U, Q, T, D_f)$. Then we compare E_{net} with E. If $E_{net} \le E$, then the algorithm is exit and return the flight plan $\Phi(U, Q, T, D_f)$ and D. Otherwise, the algorithm delete the visited data collection areas on $\Phi(U, Q, T, D_f)$ whose the amount of data collected per unit energy consumption is minimum (Fig. 1).

Algorithm 1 BOFA

Input: UAV u with flight height H, horizontal flight speed v_f^* and vertical flight speed v^*, $p_j^o = (x_j^o, y_j^o, z_j^o)$, (l_j^o, w_j^o, h_j^o);

Output: P_j^o, E_{net}^{bj};

1: Compute the coordinates of corner points $e_{sc}^{lj}, e_{ec}^{lj}, e_{sc}^{rj}, e_{ec}^{rj}, e_{sc}^{uj}$ and e_{ec}^{uj};

2: $P_j^{lo} = sp_j \to e_{sc}^{lj} \to e_{ec}^{lj} \to ep_j$;

3: $P_j^{ro} = sp_j \to e_{sc}^{rj} \to e_{ec}^{rj} \to ep_j$;

4: $P_j^{uo} = sp_j \to e_{sc}^{uj} \to e_{ec}^{uj} \to ep_j$;

5: $E_{net}^{lbj} = \frac{L(P_j^{lo})}{v_f^*} P(v_f^*) - \frac{L(P_j^{lo})}{v_f^*} P_c(H)$;

6: $E_{net}^{rbj} = \frac{L(P_j^{ro})}{v_f^*} P(v_f^*) - \frac{L(P_j^{ro})}{v_f^*} P_c(H)$;

7: $E_{net}^{ubj} = \frac{2d(sp_j, e_{sc}^{uj})}{v^*} P(v^*) + \frac{d(sp_j, ep_j)}{v_f^*}(P(v_f^*) - P_c(h_j^o + d_j^o)) - 2\int_0^{\frac{d(sp_j, e_{sc}^{uj})}{v^*}} P_c(H + v^* t)dt$;

8: $E_{net}^{bj} = min\{E_{net}^{lbj}, E_{net}^{rbj}, E_{net}^{ubj}\}$ and $P_j^o = arc(E_{net}^{bj})$;

4.1 Algorithm to bypass obstacles

In this subsection, we propose an algorithm to bypass obstacles, which is called Bypass Obstacles during Flight Algorithm (BOFA). For any $o_j \in O$, we use sp_j and ep_j to represent the positions where the UAV arrives and leaves o_j, respectively. The BOFA is used to design a path P_j^o to bypass $o_j \in O$, where P_j^o starts from sp_j to ep_j. Let $L(P_j^o)$ be the length of P_j^o. We use E_{net}^{bj} to denote the net energy consumption of UAV flying on P_j^o.

Let e_{sc}^{lj} and e_{ec}^{lj} denote the two corner points on the left of o_j considering buffer distance, respectively. We use e_{sc}^{rj} and e_{ec}^{rj} to represent the two corner points on the right of o_j considering buffer distance, respectively. Let e_{sc}^{uj} and e_{ec}^{uj} denote the two corner points right above sp_j and ep_j, respectively.

Algorithm 2 AGFP

Input: UAV u with flight height H, horizontal flight speed v_f^*, vertical flight speed v^*, $s_0 \cup S$, O, ant number μ, pheromone importance factor ς, total pheromone ϖ, heuristic importance factor ϱ, pheromone volatile factor ξ, maximum number of iterations ϵ;
Output: U_c, $L(U_c)$, $G'(SC, EC, WC)$;

For any pair $s_i, s_j \in S$, compute ip_i^j and ip_j^i;
Compute $SC = \{sc_0, sc_1, ..., sc_n\}$;
Compute $EC = \{(sc_0, sc_1), (sc_0, sc_2), (sc_0, sc_3), ..., (sc_{n-1}, sc_n)\}$
IO_i^j=$\emptyset$;
for any $(sc_i, sc_j) \in EC$ **do**
  for q from 1 to m **do**
    if the edge (ip_i^j, ip_j^i) bypasses o_q, **then**
      $IO_i^j = IO_i^j \cup o_q$, obtain E_{net}^{bq} by executing the Algorithm BOFA;
    end
  end
  if $IO_i^j = \emptyset$ **then**
    $E_{net}^{i,j} = \frac{d(ip_i^j, ip_j^i)}{v_f^*}(P(v_f^*) - P_c(H))$, $WC = WC \cup \{E_{net}^{i,j}\}$;
  else
    Let $k = |IO_i^j|$ and obtain $IO_i'^j = \{o_{l_1}, o_{l_2}, ..., o_{l_k}\}$;
    $E_{net}^{i,j} = (\frac{d(ip_i^j, sp_{l_1})}{v_f^*} + \frac{d(ep_{l_k}, ip_j^i)}{v_f^*} + \Sigma_{q=1}^{k-1}\frac{d(ep_{l_q}, sp_{l_{q+1}})}{v_f^*})(P(v_f^*) - P_c(H)) + \Sigma_{q=1}^{k} E_{net}^{bq}$;
    $WC = WC \cup \{E_{net}^{i,j}\}$;
  end
end
$L(U_c) = +\infty$, for each edge $\in EC$, set the common initial pheromone value t_0;
for $count$ from 1 to ϵ **do**
  for ℓ from 1 to μ **do**
    Randomly initialize the starting position of the ℓ-th ant which is located at one of node $sc_p \in SC$;
    The set of nodes on $G'(SC, EC, WC)$ that hasn't been visited by ℓ-th ant is $NV_\ell = SC$;
    for τ from 1 to $|SC|$ **do**
      $NV_\ell = NV_\ell \setminus \{sc_p\}$;
      Calculate the transition probability Pro on NV_ℓ based on ς, ϱ;
      Update the location sc_p of ℓ-th ant based on Pro;
    end
  end
  for ℓ from 1 to μ **do**
    Compute the circuit C_ℓ and its length $L(C_\ell)$ of the ℓ-th ant;
    if $L(C_\ell) < L(U_c)$ **do**
      $U_c = C_\ell$, $L(U_c) = L(C_\ell)$;
    end
  end
  for each edge $e_i \in EC$ **do**
    Update pheromone value based on ϖ_i and ξ_i;
  end
end

The BOFA algorithm consists of three steps. In the first step, we compute the coordinates of corner points e_{sc}^{lj}, e_{ec}^{lj}, e_{sc}^{rj}, e_{ec}^{rj}, e_{sc}^{uj} and e_{ec}^{uj} based on p_j^o, sp_j and ep_j; In the second step, we obtain the flight paths P_j^{lo}, P_j^{ro} and P_j^{uo} of UAV, where $P_j^{lo} = sp_j \to e_{sc}^{lj} \to e_{ec}^{lj} \to ep_j$, $P_j^{ro} = sp_j \to e_{sc}^{rj} \to e_{ec}^{rj} \to ep_j$ and $P_j^{uo} =$

$sp_j \to e_{sc}^{uj} \to e_{ec}^{uj} \to ep_j$. In the third step, we compute the net energy consumption, E_{net}^{lbj}, E_{net}^{rbj} and E_{net}^{ubj} of UAV when the UAV flies on P_j^{lo}, P_j^{ro} and P_j^{uo}, respectively. Finally, we let $E_{net}^{bj} = min\{E_{net}^{lbj}, E_{net}^{rbj}, E_{net}^{ubj}\}$ and $P_j^o = arc(E_{net}^{bj})$.

4.2 Construct flight path U_c

In this subsection, we propose an algorithm to construct the auxiliary graph $G'(SC, EC, WC)$ and obtain flight path U_c and its length $L(U_c)$ based on G', where SC denotes the node set, EC denotes the edge set and WC represents the set of weight of edges in EC. The algorithm is called Auxiliary Graph Flight Path (AGFP), which consists of the following three steps.

In the first step, we compute the intersection point ip_i^j between $C(s_i')$ and edge (s_i', s_j') and the intersection point ip_j^i between $C(s_j')$ and edge (s_i', s_j').

In the second step, we construct the auxiliary graph $G'(SC, EC, WC)$. We first compute the set $SC = \{sc_0, sc_1, \ldots, sc_n\}$ of points, where each sc_i is a virtual point shrunk from $C(s_i')$ and obtain the set EC that is set of edges made up of any two points in SC, i.e., $EC = \{(sc_0, sc_1), (sc_0, sc_2), \ldots, (sc_{n-1}, sc_n)\}$. Then, for any edge $(sc_i, sc_j) \in EC$, we compute the set IO_i^j of all obstacles passed by the edge (ip_i^j, ip_j^i) and the ordered set $IO_i'^j$ in which all obstacles in IO_i^j are ordered from ip_i^j to ip_j^i. Afterwards, for arbitrary $(sc_i, sc_j) \in EC$, we compute the net energy consumption $E_{net}^{i,j}$ of UAV flying from ip_i^j to ip_j^i based on $IO_i'^j$, and we use $E_{net}^{i,j}$ to denote the weight of the edge (sc_i, sc_j) and $WC = WC \cup \{E_{net}^{i,j}\}$.

In the third step, we use Ant Colony Algorithm(ACA) to construct a hamiltonian circuit U_c on $G'(SC, EC, WC)$. Let ϵ be the number of iterations of the algorithm ACA. We initialize $L(U_c) = +\infty$ and set the common initial pheromone value t_0 for each edge of EC. For any *count* from 1 to ϵ, we repeat executing the following steps. Firstly, we randomly place μ ants on some nodes in SC ($\mu < |SC|$), and let these nodes be the starting positions of ants. Secondly, each ant selects the next node from SC until all ants have visited all nodes on SC. We can obtain an initial hamiltonian circuit C_ℓ and its length $L(C_\ell)$ for the ℓ-th ant. Thirdly, we update pheromone value based on ϖ_i and ξ_i, where ϖ_i and ξ_i are total pheromone and pheromone volatile factor, respectively. Finally, for any ℓ from 1 to μ, we compare $L(C_\ell)$ with $L(U_c)$, if $L(C_\ell) < L(U_c)$, then we set $U_c = C_\ell$ and $L(U_c) = L(C_\ell)$.

Algorithm 3 CFPU

Input: $v_f^*, v^*, H, O, R, s_i, st_i, ed_i, \Gamma$;

Output: $\Upsilon(U_i, hp_i, t_i^h), E_{net}^i$;

Divide $\Omega(s_i)$ into Γ parts based on R, and obtain $\Delta_i = \{\Delta_i^1, \Delta_i^2, ..., \Delta_i^\Gamma\}$;

Compute the ordered sets $O_i^{sx} = \{o_{J1}, o_{J2}, ..., o_{Jk_1}\}$, $O_i^{xe} = \{o_{\aleph_1}, o_{\aleph_2}, ..., o_{\aleph_{k_2}}\}$, $O_i^{se} = \{o_{\iota_1}, o_{\iota_2}, ..., o_{\iota_{k_3}}\}$, and let $X_i = s_i'$, $st_i' = st_i$, $ed_i' = ed_i$.

if $s_i' \in o_{\aleph_1}$ **then**

  if $(l_{\aleph_1}^o + d_{\aleph_1}^o)/2 - |x_i - x_{\aleph_1}^o| \le (w_{\aleph_1}^o + d_{\aleph_1}^o)/2 - |y_i - y_{\aleph_1}^o|$ **then**

    if $x_i \le x_{\aleph_1}$, then $X_i = (x_{\aleph_1} - (l_{\aleph_1}^o + d_{\aleph_1}^o)/2, y_i, H)$;

    if $x_i > x_{\aleph_1}$, then $X_i = (x_{\aleph_1} + (l_{\aleph_1}^o + d_{\aleph_1}^o)/2, y_i, H)$;

  else

    if $y_i \le y_{\aleph_1}$, then $X_i = (x_i, y_{\aleph_1} - (w_{\aleph_1}^o + d_{\aleph_1}^o)/2, H)$;

    if $y_i > y_{\aleph_1}$, then $X_i = (x_i, y_{\aleph_1} + (w_{\aleph_1}^o + d_{\aleph_1}^o)/2, H)$;

  end

end

If $st_i \in o_{J1}$, then obtain P_{J1}^o by executing the BOFA algorithm, $st_i' = P_{J1}^o \cap \Delta_i^\Gamma$;

If $ed_i \in o_{\aleph_{k_2}}$ then obtain $P_{\aleph_{k_2}}^o$ by executing the BOFA algorithm, $ed_i' = P_{\aleph_{k_2}}^o \cap \Delta_i^\Gamma$;

Compute the paths $P_{st_i'}^{X_i}$, $P_{X_i}^{ed_i'}$ and $P_{st_i'}^{ed_i'}$ based on st_i', ed_i', X_i, O_i^{sx}, O_i^{xe} and O_i^{se};

Compute the amount of data $V_{st_i'}^{X_i}$, $V_{X_i}^{ed_i'}$ and $V_{st_i'}^{ed_i'}$ collected on $P_{st_i'}^{X_i}$, $P_{X_i}^{ed_i'}$ and $P_{st_i'}^{ed_i'}$;

if $V_{st_i'}^{ed_i'} < V_i$ **then**

  $\Theta_i = X_i$, $O_i^{s\theta} = O_i^{sx}$, $O_i^{\theta e} = O_i^{xe}$, $P_{st_i'}^{\Theta_i} = P_{st_i'}^{X_i}$, $P_{\Theta_i}^{ed_i'} = P_{X_i}^{ed_i'}$, $V_{st_i'}^{\Theta_i} = V_{st_i'}^{X_i}$, $V_{\Theta_i}^{ed_i'} = V_{X_i}^{ed_i'}$;

  for k from 1 to Γ **do**

    $g_k = (X_i, ed_i) \cap \Delta_i^k$;

    if $g_k \notin O_i^{xe}$ **then**

      Compute O_i^{sg}, O_i^{ge}, $P_{st_i'}^{g_k}$, $P_{g_k}^{ed_i'}$, $V_{st_i'}^{g_k}$, $V_{g_k}^{ed_i'}$;

      If $V_{st_i'}^{g_k} + V_{g_k}^{ed_i'} \ge V_i$, then $\Theta_i = g_k$, $O_i^{s\theta} = O_i^{sg}$, $O_i^{\theta e} = O_i^{ge}$, $P_{st_i'}^{\Theta_i} = P_{st_i'}^{g_k}$, $P_{\Theta_i}^{ed_i'} = P_{g_k}^{ed_i'}$, $V_{st_i'}^{\Theta_i} = V_{st_i'}^{g_k}$, $V_{\Theta_i}^{ed_i'} = V_{g_k}^{ed_i'}$;

    end

  end

  Let $O_i^{s\theta} = \{o_{\hbar_1}, o_{\hbar_2}, ..., o_{\hbar_{k_4}}\}$, $O_i^{\theta e} = \{o_{\partial_1}, o_{\partial_2}, ..., o_{\partial_{k_5}}\}$;

  $NE_{st_i'}^{\Theta_i} = (L(P_{st_i'}^{\Theta_i}) - \sum_{q=1}^{k_4} L(P_{\hbar_q}^o)/v_f^* \cdot (P(v_f^*) - P_c(H)) + \sum_{q=1}^{k_4} E_{net}^{b\hbar_q}$;

  $NE_{\Theta_i}^{ed_i'} = (L(P_{\Theta_i}^{ed_i'}) - \sum_{q=1}^{k_5} L(P_{\partial_q}^o))/v_f^*(P(v_f^*) - P_c(H)) + \sum_{q=1}^{k_5} E_{net}^{b\partial_q}$;

  if $\Theta_i = X_i$ **then**

    $hp_i = X_i$, $t_i^h = \frac{V_i - (V_{st_i'}^{X_i} + V_{X_i}^{ed_i'})}{C_u(s_i)}$, $E_{net}^i = NE_{st_i'}^{X_i} + NE_{X_i}^{ed_i'} + t_i^h(P(0) - P_c(H))$;

  else

    $hp_i = \emptyset$, $t_i^h = 0$, $E_{net}^i = NE_{st_i'}^{\Theta_i} + NE_{\Theta_i}^{ed_i'}$;

  end

  $U_i = P_{st_i'}^{\Theta_i} \cup P_{\Theta_i}^{ed_i'}$;

else

  $U_i = P_{st_i'}^{ed_i'}$, $hp_i = \emptyset$, $t_i^h = 0$;

  $E_{net}^i = (L(P_{st_i'}^{ed_i'}) - \sum_{q=1}^{k_3} L(P_{\iota_q}^o))/v_f^* \cdot (P(v_f^*) - P_c(H)) + \sum_{q=1}^{k_3} E_{net}^{b\iota_q}$;

end

4.3 Construct fight plan $\Upsilon(U_i, hp_i, t_i^h)$ of UAV in $\Omega(s_i)$

In this subsection, we propose an algorithm to construct the flight plan $\Upsilon(U_i, hp_i, t_i^h)$ and compute the net energy consumption E_{net}^i of UAV in $\Omega(s_i)$, which is called Construct Flight Plan in data collection Area (CFPA). The algorithm consists of the following five steps.

In the first step, we divide $\Omega(s_i)$ into Γ parts based on R, and let $\Delta_i = \{\Delta_i^1, \Delta_i^2, \ldots, \Delta_i^\Gamma\}$ denote the set of all hemispherical shells, where for any $\Delta_i^\vartheta \in \Delta_i$, the spherical equation is $(x - x_i)^2 + (y - y_i)^2 + z^2 = (\vartheta \frac{R}{\Gamma})^2 (z \geq 0)$.

In the second step, we compute the ordered sets $O_i^{sx} = \{o_{J1}, o_{J2}, \ldots, o_{Jk_1}\}$, $O_i^{xe} = \{o_{\aleph_1}, o_{\aleph_2}, \ldots, o_{\aleph_{k_2}}\}$, and $O_i^{se} = \{o_{\iota_1}, o_{\iota_2}, \ldots, o_{\iota_{k_3}}\}$, where O_i^{sx}, O_i^{xe} and O_i^{se} represent all obstacles passed by the edge (st_i, s_i') from st_i to s_i', edge (s_i', ed_i) from s_i' to ed_i and edge (st_i, ed_i) from st_i to ed_i, respectively.

In the third step, we initially set $X_i = s_i'$, $st_i' = st_i$, $ed_i' = ed_i$. We judge whether the point s_i' is located in the first obstacle $o_{\aleph_1} \in O_i^{xe}$. If $s_i' \in o_{\aleph_1}$, then we update the coordinates of X_i based on the following situations.

(1) $\frac{l_{\aleph_1}^o + d_{\aleph_1}^o}{2} - |x_i - x_{\aleph_1}^o| \leq \frac{w_{\aleph_1}^o + d_{\aleph_1}^o}{2} - |y_i - y_{\aleph_1}^o|$. If $x_i \leq x_{\aleph_1}$, then $X_i = (x_{\aleph_1} - \frac{l_{\aleph_1}^o + d_{\aleph_1}^o}{2}, y_i, H)$, otherwise, $X_i = (x_{\aleph_1} + \frac{l_{\aleph_1}^o + d_{\aleph_1}^o}{2}, y_i, H)$.

(2) $\frac{l_{\aleph_1}^o + d_{\aleph_1}^o}{2} - |x_i - x_{\aleph_1}^o| > \frac{w_{\aleph_1}^o + d_{\aleph_1}^o}{2} - |y_i - y_{\aleph_1}^o|$. If $y_i \leq y_{\aleph_1}$, then $X_i = (x_i, y_{\aleph_1} - \frac{w_{\aleph_1}^o + d_{\aleph_1}^o}{2}, H)$, otherwise, $X_i = (x_i, y_{\aleph_1} + \frac{w_{\aleph_1}^o + d_{\aleph_1}^o}{2}, H)$.

Afterwards, we update the coordinates of st_i' by determining whether the st_i belongs to $o_{J1} \in O_i^{sx}$. If $st_i \in o_{J1}$, then we compute P_{J1}^o by executing the BOFA algorithm, and obtain $st_i' = P_{J1}^o \cap \Delta_i^\Gamma$. Finally, we update the coordinates of ed_i' by determining whether the ed_i belongs to $o_{\aleph_{k_2}} \in O_i^{xe}$. If $ed_i \in o_{\aleph_{k_2}}$ then we compute $P_{\aleph_{k_2}}^o$ by executing the BOFA algorithm, and obtain $ed_i' = P_{\aleph_{k_2}}^o \cap \Delta_i^\Gamma$.

In the fourth step, we compute the paths $P_{st_i'}^{X_i}$, $P_{X_i}^{ed_i'}$ and $P_{st_i'}^{ed_i'}$ based on st_i', ed_i', X_i, O_i^{sx}, O_i^{xe} and O_i^{se}, and compute the amount of data $V_{st_i'}^{X_i}$, $V_{X_i}^{ed_i'}$ and $V_{st_i'}^{ed_i'}$ collected by UAV during flying on $P_{st_i'}^{X_i}$, $P_{X_i}^{ed_i'}$ and $P_{st_i'}^{ed_i'}$, respectively. Firstly, we compute the avoid obstacle path $P_{\iota_j}^o$ for any $o_{\iota_j} \in O_i^{se}$ by executing the Algorithm BOFA. Then we compute the actual flight path $P_{st_i'}^{ed_i'}$ of UAV from st_i' to ed_i' by considering obstacles in O_i^{se} based on the following four cases.

(1) $st_i = st_i'$ and $ed_i = ed_i'$. Let $P_{st_i'}^{ed_i'} = st_i \to P_{\iota_1}^o \to P_{\iota_2}^o \to \ldots \to P_{\iota_{k_3}}^o \to X_i$.

(2) $st_i = st_i'$ and $ed_i \neq ed_i'$. Let $P_{st_i'}^{ed_i'} = st_i \to P_{\iota_1}^o \to \ldots \to sp_{\iota_{k_3}} \to ed_i'$.

(3) $st_i \neq st_i'$ and $ed_i = ed_i'$. Let $P_{st_i'}^{ed_i'} = st_i \to ep_{\iota_1} \to \ldots \to P_{\iota_{k_3}}^o \to ed_i$.

(4) $st_i \neq st_i'$ and $ed_i \neq ed_i'$. Let $P_{st_i'}^{ed_i'} = st_i \to ep_{\iota_1} \to \ldots \to P_{\iota_{k_3-1}}^o \to sp_{\iota_{k_3}} \to ed_i'$.

Secondly, we compute the avoid obstacle path P^o_{Jj} for any $o_{Jj} \in O^{sx}_i$ by executing the Algorithm BOFA. Then we compute the actual flight path $P^{X_i}_{st'_i}$ of UAV from st'_i to X_i by passing obstacles in O^{sx}_i based on the following four cases.

(1) $st_i = st'_i$ and $X_i = s'_i$. Let $P^{X_i}_{st'_i} = st_i \to P^o_{J1} \to P^o_{J2} \to \ldots \to P^o_{Jk_1} \to X_i$.
(2) $st_i = st'_i$ and $X_i \neq s'_i$. Let $P^{X_i}_{st'_i} = st_i \to P^o_{J1} \to P^o_{J2} \to \ldots \to P^o_{Jk_1}$.
(3) $st_i \neq st'_i$ and $X_i = s'_i$. Let $P^{X_i}_{st'_i} = st'_i \to ep_{J1} \to P^o_{J2} \to \ldots \to P^o_{Jk_1} \to X_i$.
(4) $st_i \neq st'_i$ and $X_i \neq s'_i$. Let $P^{X_i}_{st'_i} = st'_i \to ep_{J1} \to P^o_{J2} \to P^o_{J3} \to \ldots \to P^o_{Jk_1}$.

Thirdly, we compute the avoid obstacle path $P^o_{\aleph_j}$ for any $o_{\aleph_j} \in O^{xe}_i$ by executing the Algorithm BOFA. Then we compute the actual flight path $P^{ed'_i}_{X_i}$ of UAV from X_i to ed'_i by passing obstacles in O^{xe}_i based on the following four cases.

(1) $X_i = s'_i$ and $ed_i = ed'_i$. We have $P^{ed'_i}_{X_i} = X_i \to P^o_{\aleph_1} \to \ldots \to P^o_{\aleph_{k_2}} \to ed_i$.
(2) $X_i \neq s'_i$ and $ed_i = ed'_i$. We have $P^{ed'_i}_{X_i} = X_i \to P^o_{\aleph_2} \to \ldots \to P^o_{\aleph_{k_2}} \to ed_i$.
(3) $X_i = s'_i$ and $ed_i \neq ed'_i$. We have $P^{ed'_i}_{X_i} = X_i \to P^o_{\aleph_1} \to \ldots \to P^o_{\aleph_{k_2-1}} \to sp_{\aleph_{k_2}} \to ed'_i$.
(4) $X_i \neq s'_i$ and $ed_i \neq ed'_i$. Obtain $P^{ed'_i}_{X_i} = X_i \to P^o_{\aleph_2} \to \ldots \to P^o_{\aleph_{k_2-1}} \to sp_{\aleph_{k_2}} \to ed'_i$.

Afterwards, we compute the amount of data $V^{X_i}_{st'_i}$, $V^{ed'_i}_{X_i}$ and $V^{ed'_i}_{st'_i}$ collected by UAV during flying on $P^{X_i}_{st'_i}$, $P^{ed'_i}_{X_i}$ and $P^{ed'_i}_{st'_i}$, respectively. For any $\Delta^{\vartheta}_i \in \Delta_i$, we compute the intersection points $P^{X_i}_{st'_i} \cap \Delta^{\vartheta}_i$ between Δ^{ϑ}_i and $P^{X_i}_{st'_i}$. Let $IS^{X_i}_{st'_i} = \bigcup_{\Delta^{\vartheta}_i \in \Delta_i}(P^{X_i}_{st'_i} \cap \Delta^{\vartheta}_i)$. For arbitrary $\Delta^{\vartheta}_i \in \Delta_i$, we compute the intersection points $P^{ed'_i}_{X_i} \cap \Delta^{\vartheta}_i$ between Δ^{ϑ}_i and $P^{ed'_i}_{X_i}$. Let $IS^{ed'_i}_{X_i} = \bigcup_{\Delta^{\vartheta}_i \in \Delta_i}(P^{ed'_i}_{X_i} \cap \Delta^{\vartheta}_i)$. For any $\Delta^{\vartheta}_i \in \Delta_i$, we compute the intersection points $P^{ed'_i}_{st'_i} \cap \Delta^{\vartheta}_i$ between Δ^{ϑ}_i and $P^{ed'_i}_{st'_i}$. Let $IS^{ed'_i}_{st'_i} = \bigcup_{\Delta^{\vartheta}_i \in \Delta_i}(P^{ed'_i}_{st'_i} \cap \Delta^{\vartheta}_i)$. After that, we obtain the ordered sets $IS'^{X_i}_{st'_i}$, $IS'^{ed'_i}_{X_i}$ and $IS'^{ed'_i}_{st'_i}$ by sorting $IS^{X_i}_{st'_i}$ from st'_i to X_i, $IS^{ed'_i}_{X_i}$ from X_i to ed'_i and $IS^{ed'_i}_{st'_i}$ from st'_i to ed'_i, respectively.

Finally, for any pair of points $is_g, is_{g+1} \in IS'^{X_i}_{st'_i}$, we compute the flight time $t^{sx}_g = \frac{\sqrt{(x_{is_g} - x_{is_{g+1}})^2 + (y_{is_g} - y_{is_{g+1}})^2}}{v^*_f} + \frac{|z_{is_{g+1}} - z_{is_g}|}{v^*}$ of UAV during from is_g to is_{g+1}. Let $V^{X_i}_{st'_i} = \Sigma_{g=1}^{|IS'^{X_i}_{st'_i}|-1} t^{sx}_g C_u(s_i)$. We compute the flight time $t^{xe}_g = \frac{\sqrt{(x_{is_g} - x_{is_{g+1}})^2 + (y_{is_g} - y_{is_{g+1}})^2}}{v^*_f} + \frac{|z_{is_{g+1}} - z_{is_g}|}{v^*}$ of UAV for any pair of points $is_g, is_{g+1} \in IS'^{ed'_i}_{X_i}$, during from is_g to is_{g+1}. Let $V^{ed'_i}_{X_i} = \Sigma_{g=1}^{|IS'^{ed'_i}_{X_i}|-1} t^{xe}_g C_u(s_i)$. For any pair of points $is_g, is_{g+1} \in IS'^{ed'_i}_{st'_i}$,

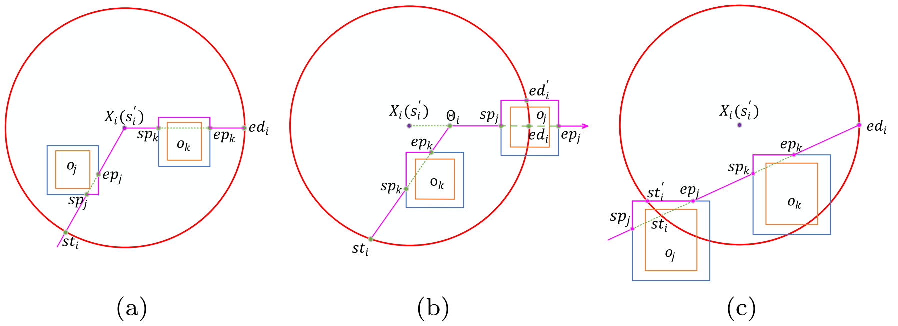


Fig. 2 Schematic diagram of UAV flight trajectory optimization: **a** $V_i > V_{st_i'}^{X_i} + V_{X_i}^{ed_i'}$, UAV flies from st_i to X_i and from X_i to ed_i, and hovers on X_i; **b** $V_{st_i'}^{ed_i'} < V_i \leq V_{st_i'}^{X_i} + V_{X_i}^{ed_i'}$, UAV flies from st_i to Θ_i, and from Θ_i to ed_i'; **c** $V_i \leq V_{st_i'}^{ed_i'}$, UAV flies from st_i' to ed_i directly

we compute the flight time $t_g^{se} = \frac{\sqrt{(x_{is_g} - x_{is_{g+1}})^2 + (y_{is_g} - y_{is_{g+1}})^2}}{v_f^*} + \frac{|z_{is_{g+1}} - z_{is_g}|}{v^*}$ of UAV during from is_g to is_{g+1}. Let $V_{st_i'}^{ed_i'} = \Sigma_{g=1}^{|IS_{st_i'}^{\prime ed_i'}|-1} t_g^{se} C_u(s_i)$.

In the fifth step, we compute $\Upsilon(U_i, hp_i, t_i^h)$ and E_{net}^i by considering the following two scenarios. The optimization result of UAV flight trajectory is shown in Fig. 2.

(1) $V_{st_i'}^{ed_i'} < V_i$. We initially set $\Theta_i = X_i$, $O_i^{s\theta} = O_i^{sx}$, $O_i^{\theta e} = O_i^{xe}$, $P_{st_i'}^{\Theta_i} = P_{st_i'}^{X_i}$, $P_{\Theta_i}^{ed_i'} = P_{X_i}^{ed_i'}$, $V_{st_i'}^{\Theta_i} = V_{st_i'}^{X_i}$, $V_{\Theta_i}^{ed_i'} = V_{X_i}^{ed_i'}$. For any k from 1 to Γ, we repeat executing the following steps.

1) Let $g_k = (X_i, ed_i) \cap \Delta_i^k$. If $g_k \notin O_i^{xe}$, then we compute the obstacles set O_i^{sg} and O_i^{ge} passed by the edge (st_i', g_k) from st_i' to g_k and edge (g_k, ed_i') from g_k to ed_i', respectively, compute the flight paths $P_{st_i'}^{g_k}$ and $P_{g_k}^{ed_i'}$ of UAV from st_i' to g_k and from g_k to ed_i', respectively, and compute the amount of data $V_{st_i'}^{g_k}$ and $V_{g_k}^{ed_i'}$ collected by UAV during flying on $P_{st_i'}^{g_k}$ and $P_{g_k}^{ed_i'}$, respectively.

2) If $V_{st_i'}^{g_k} + V_{g_k}^{ed_i'} \geq V_i$, then we set $\Theta_i = g_k$, $O_i^{s\theta} = O_i^{sg}$, $O_i^{\theta e} = O_i^{ge}$, $P_{st_i'}^{\Theta_i} = P_{st_i'}^{g_k}$, $P_{\Theta_i}^{ed_i'} = P_{g_k}^{ed_i'}$, $V_{st_i'}^{\Theta_i} = V_{st_i'}^{g_k}$, $V_{\Theta_i}^{ed_i'} = V_{g_k}^{ed_i'}$.

Afterwards, we let $O_i^{s\theta} = \{o_{\hbar_1}, o_{\hbar_2}, \ldots, o_{\hbar_{k_4}}\}$ and $O_i^{\theta e} = \{o_{\partial_1}, o_{\partial_2}, \ldots, o_{\partial_{k_5}}\}$ denote the ordered set of obstacles. Then, we compute the net energy consumption $NE_{st_i'}^{\Theta_i} = \frac{L(P_{st_i'}^{\Theta_i}) - \sum_{q=1}^{k_4} L(P_{\hbar_q}^o)}{v_f^*}(P(v_f^*) - P_c(H)) + \sum_{q=1}^{k_4} E_{net}^{b\hbar_q}$ and $NE_{\Theta_i}^{ed_i'} = \frac{L(P_{\Theta_i}^{ed_i'}) - \sum_{q=1}^{k_5} L(P_{\partial_q}^o)}{v_f^*}(P(v_f^*) - P_c(H)) + \sum_{q=1}^{k_5} E_{net}^{b\partial_q}$. Finally, we can obtain $U_i = P_{st'_i}^{\Theta_i} \cup P_{\Theta_i}^{ed'_i}$. If $\Theta_i = X_i$, then $hp_i = X_i$, $t_i^h = \frac{V_i - (V_{st'_i}^{X_i} + V_{X_i}^{ed'_i})}{C_u(s_i)}$, $E_{net}^i = NE_{st'_i}^{X_i} + NE_{X_i}^{ed'_i} + t_i^h (P(0) - P_c(H))$ and U_i is shown in Fig. 2a, otherwise, $hp_i = \emptyset$, $t_i^h = 0$, $E_{net}^i = NE_{st'_i}^{\Theta_i} + NE_{\Theta_i}^{ed'_i}$ and U_i is shown in Fig. 2b.

(2) $V_i \le V_{st'_i}^{ed'_i}$. We set $E_{net}^i = \frac{L(P_{st'_i}^{ed'_i}) - \sum_{q=1}^{k_3} L(P_{\iota_q}^o)}{v_f^*}(P(v_f^*) - P_c(H)) + \sum_{q=1}^{k_3} E_{net}^{b\iota_q}$, $U_i = P_{st'_i}^{ed'_i}$, $hp_i = \emptyset$, and $t_i^h = 0$, where U_i is shown in Fig. 2c.

4.4 Algorithm for the DCMS problem

In this subsection, we propose an approximation algorithm to solve the DCMS problem, which is called DCMSA. The algorithm consists of the following two steps.

Algorithm 4 DCMSA

Input: E, $s_0 \cup S$, v_f^*, v^*, H, O, R;
Output: $\Phi(U, Q, T, D_f)$, D;
Use the AGFP algorithm to obtain virtual graph $G'(SC, EC, WC)$ and U_c based on $s_0 \cup S$ and O and let $U_c = sc_{\rho_0} \to sc_{\rho_1} \to ... \to sc_{\rho_n} \to sc_{\rho_0}$;
Use the CFPU algorithm to obtain $\Upsilon(U_i, hp_i, t_i^h)$ and E_{net}^i for any $s_i \in S$;
$U = \bigcup_{s_i \in S} U_i \cup U_c$, $Q = \bigcup_{s_i \in S} hp_i$, $T = \bigcup_{s_i \in S} t_i^h$;
$E_{net} = \Sigma_{i=1}^n E_{net}^{\rho_i} + \Sigma_{i=0}^{n-1} E_{net}^{\rho_i, \rho_{i+1}} + E_{net}^{\rho_n, \rho_0}$;
Let $S' = \{s_{\rho_0}, s_{\rho_1}, \cdots, s_{\rho_n}\}$ be the ordered set of sensors visited by U from s_0 to s_0;
while $E_{net} > E$ **do**
  for each $s_{\rho_i} \in S' \backslash \{s_{\rho_0}\}$ **do**
    $\Lambda_{\rho_i} = \frac{V_{\rho_i}}{E_{net}^{\rho_{i-1}, \rho_i} + E_{net}^{\rho_i, \rho_{i+1}} + E_{net}^{\rho_i} - E_{net}^{\rho_{i-1}, \rho_{i+1}}}$;
  end
  $\Lambda_{\min} = \min\{\Lambda_{\rho_1}, \Lambda_{\rho_2}, ..., \Lambda_{\rho_{|S'|-1}}\}$, $s_{\rho_x} = arc(\Lambda_{\min})$, $S' = S' \setminus \{s_{\rho_x}\}$;
  $U_c = U_c \setminus \{(sc_{\rho_{i-1}}, sc_{\rho_i}), (sc_{\rho_i}, sc_{\rho_{i+1}})\}$, $U_c = U_c \cup \{(sc_{\rho_{i-1}}, sc_{\rho_{i+1}})\}$;
  Use the CFPU algorithm to obtain $\Upsilon(U_{\rho_{i-1}}, hp_{\rho_{i-1}}, t_{\rho_{i-1}}^h)$, $\Upsilon(U_{\rho_{i+1}}, hp_{\rho_{i+1}}, t_{\rho_{i+1}}^h)$, $E_{net}^{\rho_{i-1}}$, $E_{net}^{\rho_{i+1}}$;
  $U = \bigcup_{s_i \in S' \backslash \{s_{\rho_0}\}} U_i \cup U_c$, $Q = \bigcup_{s_i \in S' \backslash \{s_{\rho_0}\}} hp_i$, $T = \bigcup_{s_i \in S' \backslash \{s_{\rho_0}\}} t_i^h$;
  Let $S' = \{s_{\rho_0}, s_{\rho_1}, \cdots, s_{\rho_{|S'|-1}}\}$ be the ordered set of sensors visited by U from s_0 to s_0, $U_c = sc_{\rho_0} \to sc_{\rho_1} \to ... \to sc_{\rho_{|S'|-1}} \to sc_{\rho_0}$;
  $E_{net} = \Sigma_{i=1}^{|S'|-1} E_{net}^{\rho_i} + \Sigma_{i=0}^{|S'|-2} E_{net}^{\rho_i, \rho_{i+1}} + E_{net}^{\rho_{|S'|-1}, \rho_0}$;
end
$D = \Sigma_{s_i \in S' \backslash \{s_{\rho_0}\}} V_i$;
$D_f = D - \Sigma_{hp_i \in Q} \frac{1}{2} t_i^h W \log_2(1 + \frac{\gamma_0 P_w}{d^\alpha(s_i, hp_i)})$;

In the first step, we compute initial U, Q, T and E_{net}. First of all, we use the AGFP algorithm to obtain a virtual graph $G'(SC, EC, WC)$ and U_c based on $s_0 \cup S$ and O, and let $U_c = sc_{\rho_0} \to sc_{\rho_1} \to \ldots \to sc_{\rho_n} \to sc_{\rho_0}$. Then, for any $s_i \in S$, we use the CFPU algorithm to obtain $\Upsilon(U_i, hp_i, t_i^h)$ and E_{net}^i. Finally, we obtain $U = \bigcup_{s_i \in S} U_i \cup U_c$, $Q = \bigcup_{s_i \in S} hp_i$, $T = \bigcup_{s_i \in S} t_i^h$ and $E_{net} = \Sigma_{i=1}^{n} E_{net}^{\rho_i} + \Sigma_{i=0}^{n-1} E_{net}^{\rho_i,\rho_{i+1}} + E_{net}^{\rho_n,\rho_0}$;

In the second step, we update U, Q, T and E_{net} to obtain the flight plan $\Phi(U, Q, T, D_f)$ and D. Firstly, we let $S' = \{s_{\rho_0}, s_{\rho_1}, \cdots, s_{\rho_n}\}$ be the ordered set of sensors visited by U from s_0 to s_0. Secondly, we repeat executing the following steps when $E_{net} > E$.

1) For each $s_{\rho_i} \in S' \backslash \{s_{\rho_0}\}$, we compute $\Lambda_{\rho_i} = \frac{V_{\rho_i}}{E_{net}^{\rho_{i-1},\rho_i} + E_{net}^{\rho_i,\rho_{i+1}} + E_{net}^{\rho_i} - E_{net}^{\rho_{i-1},\rho_{i+1}}}$.
2) Let $\Lambda_{\min} = \min\{\Lambda_{\rho_1}, \Lambda_{\rho_2}, \ldots, \Lambda_{\rho_{|S'|-1}}\}$, $s_{\rho_x} = arc(\Lambda_{\min})$, $S' = S' \setminus \{s_{\rho_x}\}$.
3) Set $U_c = U_c \setminus \{(sc_{\rho_{i-1}}, sc_{\rho_i}), (sc_{\rho_i}, sc_{\rho_{i+1}})\}$, $U_c = U_c \cup \{(sc_{\rho_{i-1}}, sc_{\rho_{i+1}})\}$.
4) Use the CFPU algorithm to obtain $E_{net}^{\rho_{i-1}}$, $E_{net}^{\rho_{i+1}}$, $\Upsilon(U_{\rho_{i-1}}, hp_{\rho_{i-1}}, t_{\rho_{i-1}}^h)$, $\Upsilon(U_{\rho_{i+1}}, hp_{\rho_{i+1}}, t_{\rho_{i+1}}^h)$.
5) Set $U = \bigcup_{s_i \in S' \backslash \{s_{\rho_0}\}} U_i \cup U_c$, $Q = \bigcup_{s_i \in S' \backslash \{s_{\rho_0}\}} hp_i$, $T = \bigcup_{s_i \in S' \backslash \{s_{\rho_0}\}} t_i^h$.
6) Let $S' = \{s_{\rho_0}, s_{\rho_1}, \cdots, s_{\rho_{|S'|-1}}\}$ be the ordered set of sensors visited by U from s_0 to s_0, $U_c = sc_{\rho_0} \rightarrow sc_{\rho_1} \rightarrow \ldots \rightarrow sc_{\rho_{|S'|-1}} \rightarrow sc_{\rho_0}$ and $E_{net} = \Sigma_{i=1}^{|S'|-1} E_{net}^{\rho_i} + \Sigma_{i=0}^{|S'|-2} E_{net}^{\rho_i,\rho_{i+1}} + E_{net}^{\rho_{|S'|-1},\rho_0}$.

Finally, we can obtain $D = \Sigma_{s_i \in S' \backslash \{s_{\rho_0}\}} V_i$ and $D_f = D - \Sigma_{hp_i \in Q} \frac{1}{2} t_i^h W \log_2(1 + \frac{\gamma_0 P_w}{d^\alpha(s_i, hp_i)})$.

5 Simulation

In this section, we evaluate the performance of the DCMSA algorihtm by extensive simulation experiments on several key performance metrics under different settings. We implement the code using MATLAB 2019b and Java programming.

All results are averaged over 100 random instances. Table 1 gives the values of some constant parameters used in every instance.

5.1 An example for the DCMSA algorithm

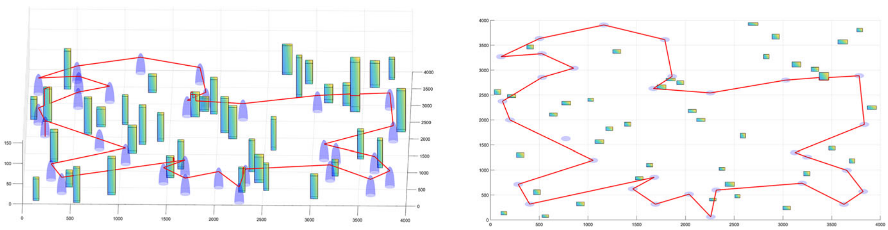
As an example shown in Fig. 3, we set the configurations as $n = 30$, $m = 40$, $R = 50$ m, $H = 40$ m, $E = 200{,}000$ J, $W = 800$ KB/s, $100 \leq V_i \leq 200$ KB for any $s_i \in S$, and the other parameters are shown in Table 1. After executing the DCMSA algorithm for the instance, we can obtain the flight path U in the three dimensional space as shown in Fig. 3a and its top view is shown as Fig. 3b, where the purple zones denote the set of sensors, the cubes represent the set of obstacles, the red lines are the flight paths of UAV.

5.2 Simulations for the DCMSA algorithm

In the following, we evaluate the impact of the different parameter settings on the Data Collection Rate(DCR) that is the proportion of D obtained by DCMSA and $\sum_{s_i \in S} V_i$.

Table 1 Some constant parameters

Notation	Physical meaning	Value
d_j^o	Buffer distance of o_j in m	2
l_j^o	Length of o_j in m	[50,100]
w_j^o	Width of o_j in m	[50,100]
h_j^o	Height of o_j in m	[30,150]
γ_0	Reference SNR at transmission distance 1 m in dB	80
η_s	Energy conversion efficiency	0.4
A_s	Solar panel area in m^2	0.1
G_s	Average solar radiation	1367
α_s	Maximum value of atmospheric transmittance	0.8978
β_s	Atmospheric extinction coefficient	0.2804
δ_s	Scale height of the earth	8000
P_0	Blade power	14.7517
P_1	Induced power	41.5409
v_r	Tip speed of the rotor blade	80
v_0	The average rotor-induced velocity	5.0463
d_0	The fuselage drag ratio	0.5009
ρ	Air density in kg/m^3	1.225
s	Rotor solidity	0.1248
A	Rotor disc area in m^2	0.1256



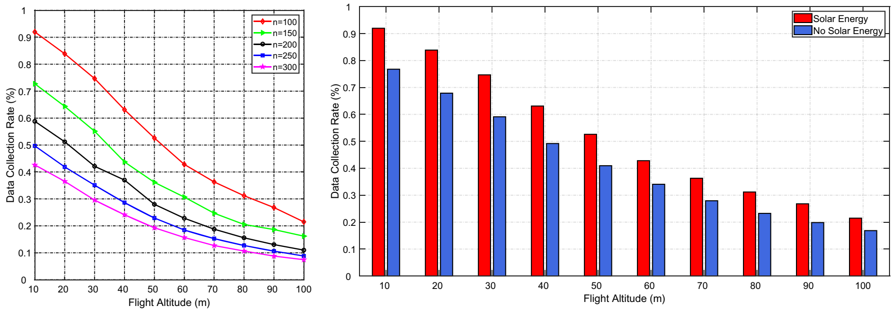
(a) The flight path of UAV in three dimensional space

(b) The top view of the path

Fig. 3 The flight of UAV for a given instance obtained by DCMA algorithm

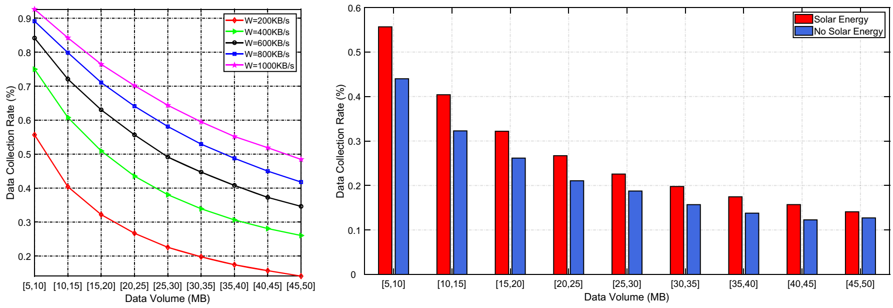
In Fig. 4, we illustrate the performance of the DCMSA algorithm when we set the detection area as 5000 m $\times$ 5000 m, $m = 40$, $R = 100$ m, $W = 800$ KB/s, $\alpha = 2$, $P_w = 10$ W, $20 \le V_i \le 25$ MB for any $s_i \in S$, $E = 700{,}000$ J, n =100, 150, 200, 250, 300 in Fig. 4a and change H from 10 to 100 m. Figure 4a measures the impact of H on the DCR, which shows that the DCR decreases with the increasing of the flight altitude H since the flight time of UAV in the data collection area of each sensor becomes shorter and the data transmission rate decreases as H increases.

At the same time, since the UAV doesn't have enough energy to collect all the data in the WSN, the DCR decreases with the increasing of the number of sensors n.



(a) Effects of n and H (b) Comparison results of DCR

Fig. 4 Simulations by changing H from 10 to 100 m under different n



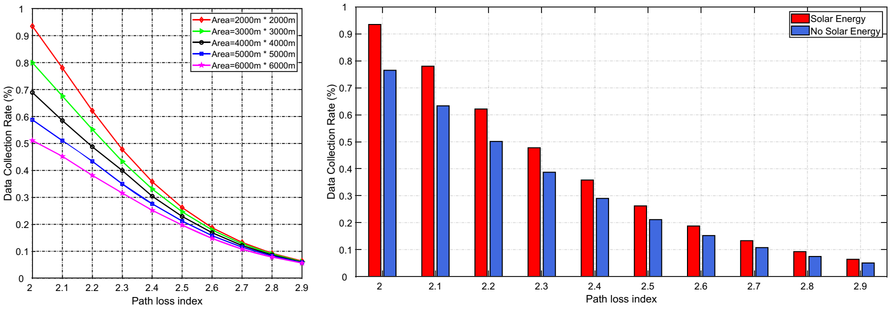
(a) Effects of W and V (b) Comparison results of DCR

Fig. 5 Simulations by changing V_i from [5,10] to [45,50] MB under different W

In Fig. 4b, the DCR when the UAV is charged by solar is obviously better than that without charging, and the average DCR of charging is 9.525% higher than that of non-charging when n =100, as shown in Fig. 4b.

In Fig. 5, we illustrate the performance of the DCMA algorithm when we set the detection area as 5000 m × 5000 m, $n = 100$, $R = 50$ m, $P_w = 10$ W, $m = 40$, $H = 40$ m, $\alpha = 2$, $E = 700{,}000$ J, W =200, 400, 600, 800, 1000 KB/s and change V_i from [5,10] to [45,50] MB for any $s_i \in S$. Figure 5a shows that the DCR decreases with the increasing of the data volume of sensors. This is because the UAV needs to consume more energy on hovering points to collect as the amount of data increases. We also observe that the DCR increases with the increasing of the value of the bandwidth. This is because the efficiency of the data collection is improved, and the flight paths within data collection area of sensors are optimized. The DCR when the UAV is charged by solar is obviously better than that without charging, and the average DCR of charging is 3.744% higher than that of non-charging when $W = 200$ KB/s, as shown in Fig. 5b.

In Fig. 6, we illustrate the performance of the DCMA algorithm as we set $n = 100$, $m = 40$, $R = 50$ m, $H = 40$ m, $W = 1000$ KB/s, $P_w = 10$ W, $20 \le V_i \le 25$ MB for any $s_i \in S$, $E = 600{,}000$ J, $\mathcal{A}$ =2000 × 2000 m, 3000 × 3000 m, 4000 × 4000 m, 5000 × 5000 m, 6000 × 6000 m in Fig. 6a and change α from 2.0 to 2.9. We can find that the DCR decreases with the increasing of α in Fig. 6a since as the data transmission



(a) Effects of $\mathcal{A}$ and α (b) Comparison results of DCR

Fig. 6 Simulations by changing α from 2.0 to 2.9 under different $\mathcal{A}$

rate decreases, it needs more time on the hovering points to collect data, which will consume additional energy. At the same time, since the flight energy consumption of UAV increases as the size of the detection area increases, the DCR decreases with the increasing of the area $\mathcal{A}$. Figure 6b shows that the DCR when the UAV is charged by solar is obviously better than that without charging, and the average DCR of charging is 5.771% higher than that of non-charging when $\mathcal{A}$ =2000 × 2000 m.

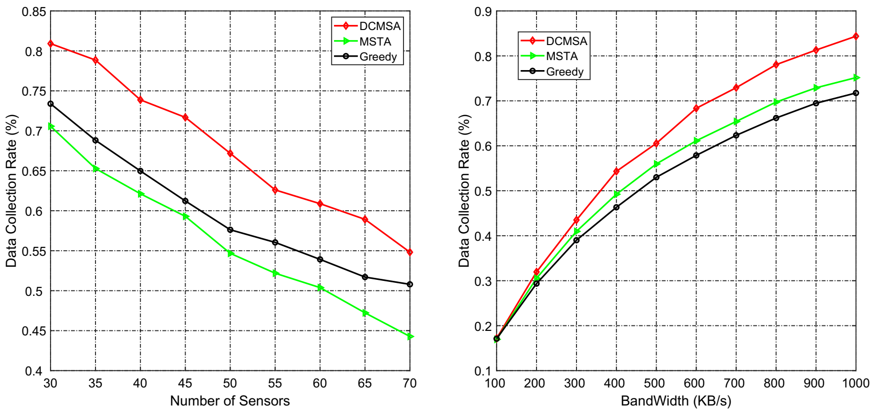
5.3 Performance comparison of different algorithms

In this subsection, we compare the performance of our algorithm DCMSA with other two algorithms MSTA and Greedy to verify the effective of the proposed algorithm.

The MSTA algorithm consists of the following steps: (1) construct a auxiliary graph $G'(SC, EC, WC)$ as shown in Algorithm 2; (2) construct a minimum spanning tree T'_G from $G'(SC, EC, WC)$; (3) obtain the hamiltonian circuit U_c by doubling all edges of T'_G; (3) execute the Algorithms 3 and 4 successively to obtain the flight plan $\Phi(U, Q, T, D_f)$ and D.

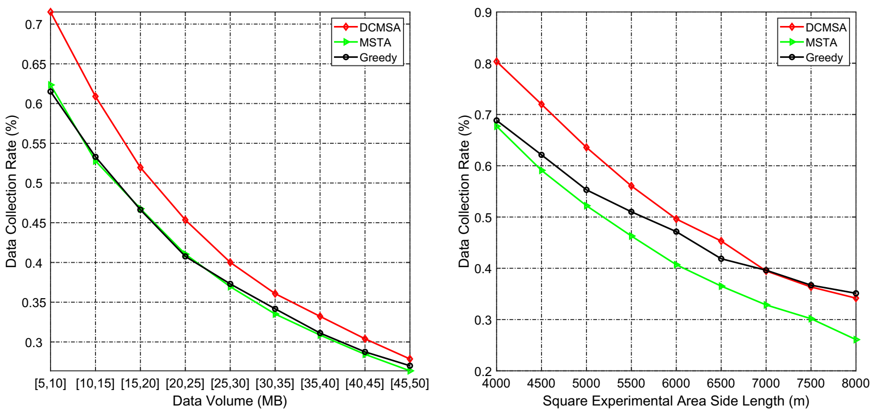
Initially, we set $s_i = s_0$ and $D = 0$, where s_i represents the initial position of UAV. The Greedy algorithm repeats the following steps until the remaining energy of UAV can not arrive the next sensor and return to the base station:(1) compute the amount of data collected per unit of energy $\Lambda_j = \frac{V_j}{E^{i,j}_{net}+E^{j}_{net}+E^{j,0}_{net}}$ for each $s_j \in S$; (2) $\Lambda_j = \max\{\Lambda_k | s_k \in S\}$ and $s_j = arc(\Lambda_j)$; (3) update the energy of UAV as $E = E - (E^{i,j}_{net} + E^{j}_{net})$, $D = D + V_j$, $s_i = s_j$, and $S = S \backslash \{s_j\}$;

In Fig. 7a, we compare the performance of three algorithms when we set the detection area as 4000 m × 4000 m, $R = 50$ m, $P_w = 10$ W, $m = 40$, $H = 40$ m, $W = 1000$ KB/s, $\alpha = 2$, $100 \leq V_i \leq 200$ KB for any $s_i \in S$, $E = 250{,}000$ J and change n from 30 to 70. Figure 7a shows that the proposed algorithm outperforms the other two algorithms. We can find that the average performance of DCMSA algorithm is about 11.522% higher than MSTA algorithm, and about 7.915% higher than Greedy algorithm.



(a) Performance comparison of three algorithms by varying the number of sensors n.

(b) Performance comparison of three algorithms as the bandwidth W grows.



(c) Performance comparison of three algorithms by changing the data volume of sensors V_i.

(d) Performance comparison of three algorithms with the different detection area $\mathcal{A}$.

Fig. 7 Performance comparison of three algorithms by varying n, W, V_i, $\mathcal{A}$

In Fig. 7b, we compare the performance of three algorithms as we set the detection area as 4000 m $\times$ 4000 m, $n = 50$, $R = 50$ m, $P_w = 10$ W, $m = 40$, $H = 40$ m, $\alpha = 2$, $25 \leq V_i \leq 30$ MB for any $s_i \in S$, $E = 500{,}000$ J and change W from 100 to 1000 KB/s. Figure 7b shows that DCMSA algorithm is superior to the other two algorithms. The average performance of DCMSA algorithm is about 5.448% higher than MSTA algorithm, and about 8.006% higher than Greedy algorithm.

In Fig. 7c, we compare the performance of three algorithms when we set the detection area as 4000 m $\times$ 4000 m, $n = 50$, $R = 50$ m, $P_w = 10$ W, $m = 40$, $H = 40$ m, $W = 500$ KB/s, $\alpha = 2$, $E = 350{,}000$ J and change V_i from [5,10] to [45,50] MB for any $s_i \in S$. Figure 7c shows that DCMSA algorithm outperforms the MSTA and Greedy algorithms. We can observe that the average performance of DCMSA algorithm is about 4.247% higher than MSTA algorithm, and about 4.083% higher than Greedy algorithm.

In Fig. 7d, we compare the performance of three algorithms as we set $n = 50$, $R = 50$ m, $P_w = 10$ W, $m = 40$, $H = 40$ m, $W = 800$ KB/s, $\alpha = 2$, $200 \leq V_i \leq 300$ KB for any $s_i \in S$, $E = 300{,}000$ J and change the detection area $\mathcal{A}$ from 4000 m $\times$ 4000 m to 8000 m $\times$ 8000 m. Figure 7d shows that DCMSA algorithm outperforms the other two algorithms. The average performance of DCMSA algorithm is about 9.476% higher than MSTA algorithm, and about 4.352% higher than Greedy algorithm.

6 Conclusion

In this paper, we investigate the Data Collection Maximization based on Solar-powered UAV(DCMS) problem in a wireless sensor network with obstacles in urban environment, which focuses on finding an optimal flight plan to maximize the data collection volume of UAV from WSN and enable the UAV to return to the base station before running out its energy. Then we prove that the DCMS problem is NP-hard. To solve the DCMS problem, we propose the BOFA algorithm to bypass obstacles, the AGFP algorithm to compute the flight path connecting all data collection areas in WSN and the CFPU algorithm to optimize the flight trajectory of UAV in data collection area of each sensor. Finally, we propose an approximation algorithm DCMSA to solve the DCMS problem based on the above proposed three algorithms, and verify the effectiveness of the proposed algorithm with a large of simulations.

Acknowledgements This work is partly supported by National Natural Science Foundation of China under Grant Nos. 62202054, 62002022.

Funding The authors have not disclosed any funding.

Data availability Enquiries about data availability should be directed to the authors.

Declarations

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

References

Alsharoa A, Ghazzai H, Kadri A, Kamal AE (2019) Spatial and temporal management of cellular HetNets with multiple solar powered drones. IEEE Trans Mob Comput 19(4):954–968

Cong J, Li B, Guo X, Zhang R (2021) Energy management strategy based on deep q-network in the solar-powered UAV communications system. In: 2021 IEEE international conference on communications workshops (ICC workshops), pp 1–6

Fu Y, Mei H, Wang K, Yang K (2021) Joint optimization of 3d trajectory and scheduling for solar-powered UAV systems. IEEE Trans Veh Technol 70(4):3972–3977

Gong J, Chang T-H, Shen C, Chen X (2018) Flight time minimization of UAV for data collection over wireless sensor networks. IEEE J Sel Areas Commun 36:1942–1954

Kingry N, Towers L, Liu Y, Zu Y, Wang Y, Staheli B, Katagiri Y, Cook S, Dai R (2018) Design, modeling and control of a solar-powered quadcopter. In: 2018 IEEE international conference on robotics and automation (ICRA), pp 1251–1258
L. Company (2012) Lockheed martin performs first ever outdoor flight test of laser powered UAS
Lahmeri M-A, Kishk MA, Alouini M-S (2020) Stochastic geometry-based analysis of airborne base stations with laser-powered UAVs. IEEE Commun Lett 24(1):173–177
Li KR, See KY, Koh WJ, Zhang JW (2017) Design of 2.45 GHz microwave wireless power transfer system for battery charging applications. In: Progress in electromagnetics research symposium—fall
Li Y, Liu M, Zhang X (2022) Research on dynamic obstacle avoidance path planning strategy of UAV. In: 2022 IEEE 4th international conference on civil aviation safety and information technology (ICCASIT), pp 461–465
Liu S, Wei Z, Guo Z, Yuan X, Feng Z (2018) Performance analysis of UAVs assisted data collection in wireless sensor network. In: 2018 IEEE 87th vehicular technology conference (VTC Spring), pp 1–5
Luo C, Chen W, Li D, Wang Y, Du H, Wu L, Wu W (2020) Optimizing flight trajectory of UAV for efficient data collection in wireless sensor networks. Theor Comput Sci 853:25–42
Luo C, Satpute MN, Li D, Wang Y, Chen W, Wu W (2021a) Fine-grained trajectory optimization of multiple UAVs for efficient data gathering from WSNs. IEEE ACM Trans Netw 29(1):162–175
Luo C, Chen W, Li D, Wang Y, Du H, Wu L, Wu W (2021b) Optimizing flight trajectory of UAV for efficient data collection in wireless sensor networks. Theor Comput Sci 853:25–42
Luo C, Hou Y, Hong Y, Chen Z, Liu N, Li D (2022) Aoi minimizing of wireless rechargeable sensor network based on trajectory optimization of laser-charged UAV. In: Algorithmic aspects in information and management: 16th international conference, AAIM 2022, Guangzhou, China, August 13–14, 2022, Proceedings. Springer, pp 255–267
Sun C, Xiong X, Ni W, Ohtsuki T, Wang X (2022) Max–min fair 3d trajectory planning for solar-powered UAV-assisted data collection. In: 2022 IEEE/CIC international conference on communications in China (ICCC), pp 610–615
Thipyopas C, Sripawadkul V, Warin N (2019) Design and development of a small solar-powered UAV for environmental monitoring application. In: 2019 IEEE Eurasia conference on IOT, communication and engineering (ECICE), pp 316–319
Wang Z, Xu W, Yang D, Lin J (2019) Joint trajectory optimization and user scheduling for rotary-wing UAV-enabled wireless powered communication networks. IEEE Access 7:181369–181380
Wu P, Xiao F, Huang H, Chuan Wang R (2020) Load balance and trajectory design in multi-UAV aided large-scale wireless rechargeable networks. IEEE Trans Veh Technol 69:13756–13767
Zhou Q, Wei Y, He W, Shang S, Fan H, Yin W (2022) Research on obstacle avoidance algorithm of fixed-wing UAV swarms based on improved artificial potential field. In: 2022 international conference on automation, robotics and computer engineering (ICARCE), pp 1–4

Authors and Affiliations

Chuanwen Luo[1,2] · Junzhe Hu[1,2] · Yunan Hou[1,2] · Yi Hong[1,2] · Yuqing Zhu[3] · Deying Li[4]

Junzhe Hu
junzhe_hu@163.com

Yunan Hou
hyn190303107@163.com

Yuqing Zhu
yzhu14@calstatela.edu

Deying Li
deyingli@ruc.edu.cn

[1] School of Information Science and Technology, Beijing Forestry University, Beijing 100083, People's Republic of China

[2] Engineering Research Center for Forestry-oriented Intelligent Information Processing of National, Forestry and Grassland Administration, Beijing 100083, China

[3] Department of Computer Science, California State University at Los Angeles, Los Angeles, CA 90032, USA

[4] School of Information, Renmin University of China, Beijing 100872, People's Republic of China