Energy-Efficient UAV Flight Planning for a General PoI-Visiting Problem with a Practical Energy Model

Jianping Huang*, Feng Shan*, Runqun Xiong*, Yuchao Shao†, Junzhou Luo*
* School of Computer Science and Engineering, Southeast University, Jiangsu, Nanjing 210096, China.
† School of Cyber Science and Engineering, Southeast University, Jiangsu, Nanjing 210096, China.
{huangjp1997, shanfeng, rxiong, yuchaoshao, jluo}@seu.edu.cn

***Abstract*—Unmanned aerial vehicles (UAVs) are being widely exploited for various applications, *e.g.,* traverse to collect data from ground sensors, patrol to monitor key facilities, move to aid mobile edge computing. We summarize these UAV applications and formulate an abstract problem, namely the *general waypoint-based PoI-visiting problem*, aiming at minimizing flight energy consumption, which is critical due to its limited onboard storage capacity. In our problem, we pay special attention to the energy consumption for turning and switching operations on flight planning, which is usually ignored in the literature but plays an important role in practical UAV flights. We propose a novel method that uses specially designed graph parts to model the turning and switching cost and thus transfer the problem into a classic graph problem, *i.e.,* traveling salesman problem, which can be efficiently solved. Finally, we evaluate our proposed algorithm by simulations. The results show it costs less than 107% of the optimal minimum energy consumption for small scale problem and costs only half as much energy as a naive algorithm for large scale problem.**

***Index Terms*—Unmanned Aerial Vehicle, Energy Efficient, Path Planning, Graph Theory**

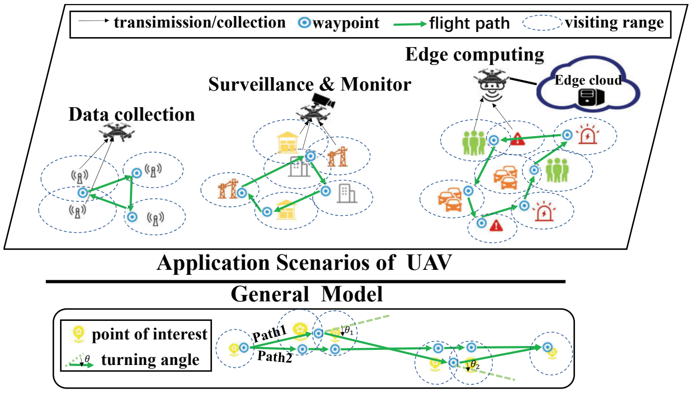


Fig. 1. The application scenarios and the general problem. A general problem is formulated from various application scenarios, such as data collection, surveillance and monitor, UAV-aided edge computing.

I. Introduction

Unmanned aerial vehicles (UAVs) are becoming increasingly popular because they are more and more affordable. They are being exploited widespreadly for various applications, *e.g.,* traverse to collect data from ground sensors [1], patrol to monitor key ground facilities [2], move to aid ground mobile edge computing [3]. Compared with traditional ground robots [4], which have to avoid countless obstacles or otherwise restricted to given routes (road or rail), UAVs are more flexible and mobile.

However, due to the limited energy storage capacity onboard, the energy consumption of UAV directly affects its flight endurance. Ahmed *et al.* [5] conclude that, for a typical commercial UAV, the flight energy consumption accounts for the most proportion than any other operations, such as wireless transmission. There are three conventional flight energy consumption models in the literature: the distance-related model [5]–[8] where the energy consumption is proportional to the distance covered, the duration-related model [1], [9], [10] where the energy consumption is proportional to the duration of the flight, and the speed-related model [11]–[13] where the energy consumption is related to the flight speed. However, all models simplify UAV flight operations and do not reflect energy consumption accurately.

To reveal a more practical and accurate energy consumption model for a UAV, we conduct a set of real-world experiments. We disclosed that, in addition to covered distance, varying speed and making turns at stopping points also affect energy consumption. Our energy model hence is distinct from most existing researches, which gives us inspiration and motivation to study the energy-efficient UAV flight planning.

We also discover that automatic flight planning is mostly implemented by waypoints, for both community-supported open-source UAVs [14] and commercial closed-source UAVs [15]. In other words, modern rotor-wing UAVs plan their route by a serial of waypoints, which is a location when UAVs stop and make turns. Therefore, waypoints divide the flight route into a serial of straight lines, where a UAV accelerates, maintains the cruise speed, and decelerates. Because of the additional energy consumption caused by acceleration/deceleration, fewer waypoints result in less cost. By investigating some popular UAV applications, such as data collection, facility patrol, UAV-aided edge computing and so on, we formulate a general waypoint-based flight planning problem where a UAV is planned to fly and visit a set of Points of Interests (PoIs). A UAV visits a PoI by visiting any point within its range, where ranges of PoIs may overlap. Even when a visiting point is within the range of two PoIs, a UAV visits (motionlessly) PoIs at this point one by one, with switching time between two visits. Such points are called waypoints for UAV flight planning, and we refer to such problem as the *waypoint-based PoI-visiting* problem.

Such a generalized problem matches various application scenarios as shown in Fig. 1. (1) In the data collection scenario, a UAV traverses all sensors to collect data and each ground sensor has a transmission range, within which the UAV can receive the sensed data. The transmission ranges may overlap with each other, while in the common range, the UAV has to switch from receiving one sensor to another. (2) In the surveillance and monitor scenario, a UAV patrols a set of key facilities with a camera and each facility has a visibility range, only within which the onboard optical equipment can sense useful data. The visibility ranges may overlap when two facilities are close, but the UAV must switch by rotating the camera from one facility to another. (3) In the UAV-aided edge computing scenario, a UAV equipped with a power computation unit is dispatched to aid ground device computation and each device has a computation offloading range, within which the UAV can receive the offloaded computation tasks. Such offloading ranges may overlap with each other, while the UAV has to receive offloading tasks sequentially.

Although much work [1], [2], [16]–[18] has studied these popular UAV applications, most of them view the ground PoI as a single "point" rather than a range, failing to model general problems in real-world scenarios. Moreover, some existing researches investigated the UAV flight planning for special application scenarios only, *e.g.,* the algorithm proposed in [1] is for collecting data, the algorithm in [2] is designed for monitoring in severe environment, and the algorithm in [16] is intended for automating CSI map construction, so it is uneasy to extend these algorithms to other application scenarios.

This paper adopts a more practical flight energy model, especially considering the energy consumption of turning and switching, to study the *waypoint-based PoI-visiting* problem, which is quite challenging to solve. The readers can sense some challenges from two example paths illustrated in Fig. 1. Compared with Path 2, Path 1 consists of fewer waypoints, consuming less energy by the definition of waypoint. However, Path 1 is longer than Path 2, which consumes more energy. Besides, Path 1 costs more turning energy and switching energy than Path 2. Because the turning angle of Path 1 is larger than that of Path 2, and the waypoints of Path 1 are in the common range of two PoIs. In a summary, a two-fold tradeoff needs to be handled for any UAV flight planning algorithm. On the one hand, there is a tradeoff between waypoints and flight distance. The UAV usually has to detour to visit these waypoints in common range due to the small range of the overlap, increasing the cost of covered distance. On the other hand, there is a tradeoff between waypoints, turning angle, and the number of switching. Similarly, since the waypoints in the overlap are more restricted, the route is usually winding which implies the larger angle of turns. Meanwhile, more waypoints in the overlap cause more frequent switching. As mentioned above, the energy consumption of turning and switching is a large proportion of the total.

As the challenges stated above, it is onerous to find a straightforward solution to the *waypoint-based PoI-visiting* problem. Since its partial goal is to minimize the covered distance of a tour, it is natural to think about the classic traveling salesman problem (TSP) and its variants, aiming to find a minimum-cost cycle on a graph. However, one important missing step is how to embed the turning cost (proportion to the turn angle) and switching cost into a graph.

Therefore, the contributions of this paper are summarized as follows.

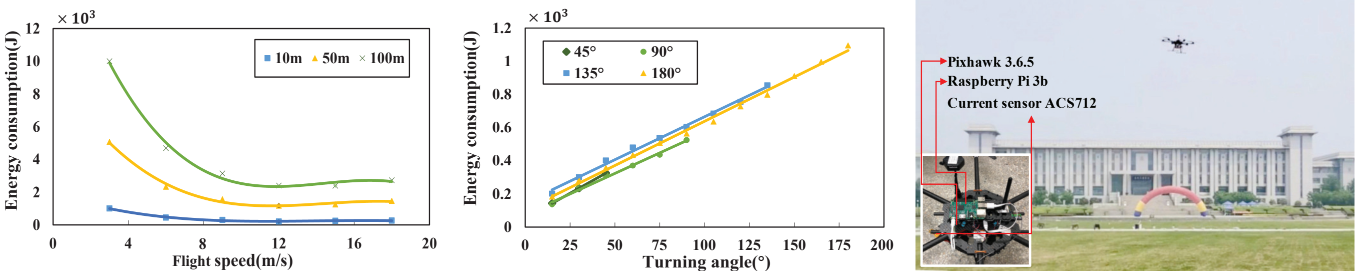
- We devise a set of real-world experiments to develop a more practical flight energy model for rotor-wing UAVs. In this model, turning at a waypoint and switching between PoIs also cost energy. This model is distinct from most existing studies.
- We formulate a general problem that is suitable for various UAV application scenarios, *e.g.,* data collection, surveillance and monitor, and UAV-aided edge computing. This problem is to design a waypoint-based flight planning to minimize the UAV energy consumption according to our energy model.
- We propose a novel approach that uses regular polygons to model the turning cost and virtual split to model the switching cost, converting the problem into a classic graph problem, which can be solved efficiently by an existing solution.
- We conduct simulations to evaluate the performance of the proposed algorithm. The results show it performs near the optimal solution, within 107% of the minimum energy consumption for small scale problems, and costs only half the energy by a naive algorithm for large scale problems.

The rest of the paper is organized as follows. Section II surveys related work. Our motivation is presented in Section III. And Section IV shows the system model and problem formulation. Then a solution is given in Section V to the problem. The elaborate simulations are introduced in Section VI. Finally, Section VII concludes the paper.

II. Related Work

A. UAV flight energy model

As stated above, there are three types of energy consumption models commonly used for the UAV flight: the distance-related model, the duration-related model, and the speed-related model. Ahmed *et al.* [5] obtain the distance-related model to assign energy efficient trajectories for a fleet of UAVs. Liu *et al.* [6] propose a UAV distance-related model to optimize UAV communication coverage, connectivity and energy consumption. Xiong *et al.* [7] leverage distance coverage model to solve UAV efficient-energy problem by a dynamic programming approach. Huang *et al.* [8] propose a general control and monitoring platform for cooperative UAS according to the distance-related model. There are also extensive efforts that adopt a duration-related model. Gong *et al.* [1] focus on a duration-related mode and use a UAV



(a) Flight speed and energy consumption.

(b) Turning angle and energy consumption.

(c) Our UAV during a flight test.

Fig. 2. A practical energy consumption model based on our real-world flight test. In (a), there is an optimal speed to minimize the energy consumption for a fixed distance. (b) illustrates that energy consumption is a linear function about the angle of turn. (c) is a photo of our UAV during a flight test.

to collect data from a set of ground sensors with flight time minimization. Mozaffari *et al.* [9] use a duration-related model to address the problem that how to efficiently deploy multiple UAVs to collect data from Internet of Things (IoT) devices. Many researches indicate that the speed-related model is more practical. Zeng *et al.* [19] developed a worthy speed-related energy model to minimize the total UAV energy consumption. Morbidi *et al.* [12] leverage a brushless DC motor to get the speed-related model, determining minimum-energy paths for UAV. While some speed-related models assumed the flight power is proportional to either the linear [20] [21] or the square [22] [23] of flight speed, which are both too simple.

B. UAV flight planning

Furthermore, plentiful researches provide valuable ideas for UAV flight planning. Bouzid *et al.* [24] propose a flight planning algorithm to dispatch UAVs to detect specific locations in disaster relief missions. Yang *et al.* [18] design a multi-object bionic flight planning algorithm to find a well-performed path for UAV to visit key locations. Gong *et al.* [1] use UAVs for data collection over sensor networks with flight time minimization, formulating flight planning as a dynamic programming problem. Liu *et al.* [25] leverage flight planning based on a deep learning approach for UAV to collect data from mobile crowdsensing. Vallejo *et al.* [2] plan a path for UAV by characterizing critical points in a catastrophe scenario. To the best of our knowledge, most studies simply the real-world scenarios, *i.e.,* regard the visiting point as a single point, or assume the visiting ranges never overlap.

III. Motivation

To obtain a more practical and accurate UAV energy consumption model, we conduct a series of real-world experiments. In this section, we present the detailed experiment settings and results. The experiment results show that varying flight speed and making turns also affect energy cost, which motivates the study of this paper.

In the flight tests, our equipment is a 3.8kg Model X4108 hexacopter rotor-wing UAV with a 1000mAh battery capacity. This UAV is with autopilot Pixhawk 3.6.5, connecting to a companion computing device, Raspberry Pi 3b single-board computer (RPi). A current module ACS712 is installed onboard to detect the real-time battery current, which is read by the RPi via an I^{2}C communication protocol. The real-time voltage value is read by the RPi through USB via the MAVLink communication protocol. It is easy to calculate the real-time power consumption given the real-time current and voltage. By MAVLink protocol, the companion RPi sends control commands to UAV automatically via the USB link. In our flight tests, the basic action sequence of UAV is to takeoff first, then to accelerate to the desired speed (cruise speed) and fly at this speed, next to decelerate to 0 for hovering or making turns, and finally to prepare for the next round of flight. A photo of our UAV during a flight test can be found in Fig. 2(c).

Test 1: the relationship between flight speed and flight energy power. In this test, we let the UAV fly straightly at a fixed distance of 10m, 50m, 100m respectively at various cruise speeds. We test 6 speeds for each distance, and the relationship between UAV energy consumption E and flight speed v is shown in Fig.2(a). An optimal speed v^* for a certain distance can be found to minimize the energy consumption.

Test 2: the relationship between turning angles and flight energy consumption. In this test, we command the UAV to make a turn at an angle of 45°, 90°, 135°, and 180° respectively. The energy consumption is illustrated in Fig.2(b). It is easy to see that the angle θ and the UAV energy consumption E are related nearly through function $E = 5.3316\theta + 104.65$ in our settings.

Hereby, we obtain a more practical and accurate UAV flight energy model, disclosing the effect of flight speed and turning is non-negligible. This result motivates us to redefine UAV flight energy consumption composition, which will be described in detail in the next section.

IV. System model and problem formulation

A. System model

Assume there are n PoIs randomly located within a rectangle region, denoted as PoI $i, i = 1, 2, \cdots, n$. A PoI can be a wireless sensor, a facility, or a mobile device. Each PoI has a range, *e.g.,* the transmission range of a wireless sensor, the visibility range of a key facility, the computation offloading range of a mobile device. The area covered by the range of

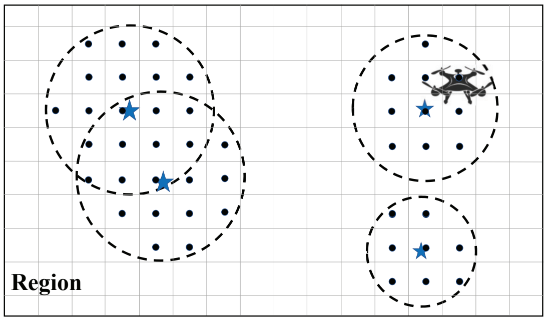


Fig. 3. The relationships for ranges, grids and waypoints within the region. A set of square grids is used to rasterize the region. PoIs are marked by blue stars, of which the ranges are colored in black dashed circles, while the candidate waypoints are marked by black dots.

PoI i is denoted as R_i, which is usually a circle and its radius is allowed to vary for different PoIs. Note that R_i may overlap with another range area R_j.

There is a base station within the region, and a UAV is dispatched periodically from this station to tour and visit all PoIs. PoI i is visited by the UAV if any point within area R_i is visited. Without the loss of generality, assume the base station is denoted as PoI 0. In each tour, the UAV flies at a fixed altitude, and follows a waypoint-based route. Assume there are m waypoint on the route, denoted as Waypoint $j, j = 1, 2, \cdots, m$. Obviously, at Waypoint x, the UAV can visit any PoI i if R_i contains Waypoint x. Let D_x represent the set of PoIs that the UAV visits at Waypoint x. If $|D_x| > 1$, it means the UAV visits (motionlessly) more than one PoI at Waypoint x. Since every PoI must be visited, we have *all PoIs visiting constraint*,

$$|\cup_x D_x| = n + 1. \tag{1}$$

We want to plan a route that consists of a serial of waypoints, and ensure all PoIs are visited. Since a waypoint that visits a certain PoI can be anywhere inside its range, we divide the region by small grid to reduce the searching space of the planning algorithm. Assume a $N \times M$ size region is rasterized into a $g \times g$ size grid. Thus area R_i of PoI i may cover a set of grid centers, and let the set be denoted as S_i. Any Waypoint x is assumed to be at a grid center, so the grid centers are called the *waypoint candidates*. Because at least one waypoint is required to be within R_i, set S_i is also called the *candidate waypoint set* for PoI i. In this way, Waypoint x, where PoI i is visited, *i.e.,* $i \in D_x$, must be in the candidate waypoint set of PoI i, *i.e.,* $x \in S_i$, so we have the *range visiting constraint*,

$$\forall i \in D_x \Rightarrow x \in S_i \qquad \forall x. \tag{2}$$

An illustration of the relationship of ranges, grids, and waypoints is in Fig. 3.

Now planning a route is simplified to choose waypoints from all the waypoint candidates. Assume x and y are two arbitrary waypoint candidates of the region. Let w_{xy} indicts whether straight direct path e_{xy}, from Waypoint x to Waypoint y, is included in the route,

$$w_{xy} = \begin{cases} 1, & e_{xy} \text{ is in the route,} \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

Then, when $|D_x| > 0$, there must be a path to and from x, while $|D_x| = 0$, certainly there is no such path. We have the following *route connecting constraint*,

$$\sum_{\forall x} w_{xy} = \sum_{\forall y} w_{xy} = \begin{cases} 1, & |D_x| > 0, \\ 0, & |D_x| = 0. \end{cases} \tag{4}$$

Our goal is to plan a route such that the UAV energy consumption is minimized.

B. Energy consumption model

We now model the energy consumption according to our real-world experiments in the previous section.

Let x and y be two arbitrary waypoint candidates. Flying straightly from x to y, the UAV first accelerates to the cruise speed and then flies at this speed until it decelerates to 0 to visit at Waypoint y.

Definition 1 (Straight flight energy consumption)**.** *The energy consumption for a straight flight on path e_{xy}, from Waypoint x to y, is defined as $E(e_{xy})$, which is related to acceleration/deceleration and cruise speed.*

Our proposed method can handle arbitrary energy function $E(e_{xy})$. In our simulations, we set $E(e_{xy}) = c_1|e_{xy}| + C_1$, where c_1 is the energy consumption ratio proportional to the distance, and C_1 is the energy consumption related to acceleration/deceleration. We denote the total energy consumption related to straight flight as E_C on a tour formulated as

$$E_C \quad = \sum_{\forall x, \forall y} E(e_{xy}) w_{xy}. \tag{5}$$

Let x, y and z be three arbitrary waypoint candidates that are not on a straight line. Between straight paths, e_{xy} and e_{yz}, the UAV has to make a turn at y. Denote the angle of turn as q_{xyz}, which is determined by the locations of x, y and z.

Definition 2 (Turning energy consumption)**.** *The energy consumption for a UAV to make a turn at Waypoint y, from x to z, is defined to be $E(q_{xyz})$, where q_{xyz} is the heading angle changed from path e_{xy} to e_{yz}.*

Energy function $E(q_{xyz})$ is the energy consumption related to make turns by angle q_{xyz}. Our proposed method can handle arbitrary energy function $E(q_{xyz})$. In our simulations, we set $E(q_{xyz}) = c_2 q_{xyz} + C_2$, where c_2 and C_2 are the constant factors that related to a special UAV. The total energy consumption for all turns is denoted as E_T, which can be calculated as

$$E_T \quad = \sum_{\forall x, \forall y, \forall z} E(q_{xyz}) w_{xy} w_{yz}. \tag{6}$$

If D_x contains more than one PoI, the UAV has to switch visiting at Waypoint x. Additional energy consumption occurs because the UAV is required to hover to setout the switch, *e.g.,* establishing connection to a mobile device or sensor, rotating the optical camera from one direction to another, and the cost is called switching energy consumption.

Definition 3 (Switching energy consumption)**.** *The energy consumption for a UAV to switch between PoIs at Waypoint x is defined to be $E(D_x)$, which is related to the number of PoIs in D_x.*

If $|D_x| = 1$, there is no switch needed, we set the switching energy to be proportional to $|D_x|-1$, *i.e.,* $E(D_x) = c_3(|D_x|-1)$, where c_3 is the constant factor that related to a special UAV. So the total energy consumption of switching between PoIs is aggregated by all waypoints, denoted as E_S, which can be computed as

$$E_S = \sum_{\forall x} E(D_x). \tag{7}$$

After explicitly decomposing the total energy consumption, we now recompose it by summing the costs of three parts discussed above, simply marked as E_{ALL}.

$$E_{ALL} \quad = \quad E_C + E_T + E_S. \tag{8}$$

C. Problem formulation

Given the model described above, we are ready to define this problem.

Definition 4 (P1)**.** *Given a set of PoIs and models mentioned above, the waypoint-based PoI-visiting problem is to find a route for UAV to minimize the total energy consumption in Eq. (8), while the* range visiting constraint *Eq. (2), the* all PoIs visiting constraint *Eq. (1), the* route connecting constraint *Eq. (3) and Eq. (4) are satisfied.*

V. The model of turning and switching on graph

Problem P1 seeks a flight route, starting from PoI 0 and ending at PoI 0, selecting a serial of waypoints to visit each PoI with the minimum total energy consumption in Eq. (8). This problem seems to have deep roots in classic graph problems, such as the generalized traveling salesman problem (GTSP). In GTSP, there is a set of cities and some subsets of these cities, where a salesman must visit every subset by one of its city with the shortest tour and ultimately return to the starting city. In other words, if we view energy cost as edge weight, then a tour must be discovered in a complete weighted graph to cover all subsets and the sum of weights of the tour edges is the minimum.

However, one important missing step is how to embed the turning cost, which is proportional to the turning angle, and the switching cost, which is proportional to the number of switching, into a graph. More specially, there are two most significant difficulties with this problem: 1) the *turning energy consumption* in Definition 2 is unable to be intuitively reflected by any graph element such as the edge weight. For example in Fig. 4(a), at Waypoint y, its *turning energy consumption* is related to the turning angle, so its last waypoint, x, and its next waypoint, z, directly affect the turning cost on the route. Both two waypoints are not determined when the graph is constructed, so it is arduous to represent the cost of making turns on the graph by edge weight. And 2) the *switching energy consumption* in Definition 3 can not be easily represented in the graph model either. For example in Fig. 5(a), Waypoint x_1 is in the overlap of PoI 1 and PoI 2, so the UAV can either visit (motionlessly) the two PoIs sequentially with switching cost at x_1, or visits one PoI only without switching cost, which can not be determined when the graph is constructed, so it is hard to be modeled by edge weight. Therefore, how to convert the energy consumption of straight flight, the energy consumption of turning, and the energy consumption of PoI-switching, into a unified form (edge weight) on the graph model is our main task. In the following subsections, we propose a novel approach to solve this problem ingeniously.

We start with a simple case, *i.e.,* modeling energy cost of straight flight by graph $G_1(\mathbb{S}_1, \mathbb{E}_1, \mathbb{W}_1)$. The vertex set $\mathbb{S}_1$ is defined to encompass all waypoint candidates, $\mathbb{S}_1 = S_1 \cup S_2 \cup \cdots \cup S_n$. The edge set $\mathbb{E}_1$ includes e_{xy} if Waypoint x and y are candidates of two different PoIs. In order to model the energy cost of straight flight on e_{xy}, we directly set the edge weight $W(e_{xy}) = E(e_{xy})$. And $\mathbb{W}_1$ includes all weights. Hence, graph $G_1(\mathbb{S}_1, \mathbb{E}_1, \mathbb{W}_1)$ is generated.

A. Modeling energy cost of making turns

This subsection improves graph G_1 to include the energy cost of making turns, and generates graph G_2.

According to our practical energy consumption model, we have the turning energy consumption $E(\theta) = c_2\theta + C_2$, where θ is the heading angle changed, and c_2 and C_2 are factors that related to a special UAV. Since energy C_2 is constant for any turn, it can be added to the weight of edges directly, $W(e_{xy}) = E(e_{xy}) + C_2, \forall x, y$. Now we focus on modeling the proportional energy to angles $c_2\theta$.

The core idea is that we approximate the infinitely precise turning angle into a set of finite options, and use edge weight to represent the energy cost of making turns. To simplify the illustration and make is easier to calculate, we utilize regular octagons in Fig. 4(b) to replace original waypoints in Fig. 4(a). It is also feasible to choose other regular polygons, like hexagon, decagon and so on. Then we evenly divide the infinite 360° turning angle into 8 ranges to present 8 heading directions of a UAV, where each corner corresponds to one direction range of 45°. Hence when a UAV makes a turn, the change of its heading direction is demonstrated by two kinds of corners on the octagon, *i.e.,* the arrival corner and departure corner. In this way, the turning is proportional to the path length between the two corners, as shown in Fig. 4(b). Furthermore, during a straight flight path, the heading direction of the UAV is determined by the departure corner and stays changeless until arrives at the next octagon. For this reason, the reachable target corners of octagons are within the direction range of the departure corner.

Formally, we convert graph $G_1 = (\mathbb{S}_1, \mathbb{E}_1, \mathbb{W}_1)$ into $G_2 = (\mathbb{S}_2, \mathbb{E}_2, \mathbb{W}_2)$. Any vertex (waypoint) $x \in \mathbb{S}_1$, is expanded into a regular octagon, denoted as O_x, with 8

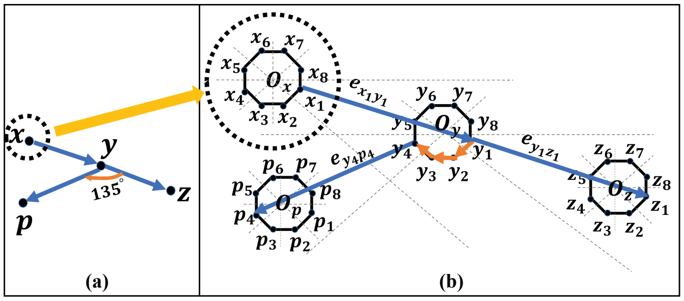


Fig. 4. An example of modeling energy cost of making turns. In (a), according to our real-world experiments, making turns also costs UAV energy, which is proportional to turning angles. This part of energy consumption is hard to be represented in a graph. In (b), we utilize regular octagons to replace original waypoints, evenly dividing the infinite 360° turning angle into 8 ranges to present 8 heading directions of a UAV, where each corner of ocatgon corresponds to 45° direction range.

vertices, indexed as $x_1, x_2, \cdots, x_8$, each representing one 45° direction. Here we let $x_8 = x_0$ for loop purpose. And the set $\mathbb{S}_2$ includes all vertices of octagons. Between every two adjacent vertices on O_x, such as x_i and x_j, there is a directed edge, $e_{x_ix_j} \in \mathbb{E}_2, i = 1, 2, \cdots, 8$, whose weight is set to $W(e_{x_ix_j}) = 45°c_2$, representing the cost of a 45° turn. For straight flight, such as from O_x to O_y, the UAV keeps its heading direction unchange, which means the range of departure corner x_i on O_x must cover O_y, and the arrival corner y_i on O_y has the same direction index as x_i. We can ascertain the direction index of O_y with the respect to O_x, defined as d_{xy}. Therefore, for any edge $e_{xy} \in \mathbb{E}_1$, we create an edge $e_{x_iy_i} \in \mathbb{E}_2$, where $i = d_{xy}$. Note that O_x is an infinitesimal regular octagon without physical significance, so the weight of $e_{x_iy_i}$ is equal to the that of original edge e_{xy}, *i.e.*, $W(e_{x_iy_i}) = W(e_{xy})$. Let $\mathbb{W}_2$ covers all weights. The pseudo code is in Algorithm `ModelingTurns`.

```
Algorithm 1: ModelingTurns ($G_1(\mathbb{S}_1, \mathbb{E}_1, \mathbb{W}_1)$)
1  for each $x \in \mathbb{S}_1$ do
2      $\mathbb{S}_2 = \mathbb{S}_2 \cup \{x_1, x_2, \cdots, x_8\}$;
3      $\mathbb{E}_2 = \mathbb{E}_2 \cup \{e_{x_1x_2}, e_{x_2x_3}, \cdots, e_{x_8x_1}\}$;
4      $\mathbb{W}_2 = \mathbb{W}_2 \cup \{W(e_{x_ix_j}) = 45°c_2\}, \forall i \neq j$;
5  end
6  for each $e_{xy} \in \mathbb{E}_1$ do
7      $\mathbb{E}_2 = \mathbb{E}_2 \cup e_{x_iy_i}$, where $i = d_{xy}$;
8      $\mathbb{W}_2 = \mathbb{W}_2 \cup W(e_{x_iy_j}) = W(e_{xy})$;
9  end
10 return $G_2(\mathbb{S}_2, \mathbb{E}_2, \mathbb{W}_2)$
```

An example is given in Fig. 4(b) to help the readers to get a better sense on how our modeling works. There are four waypoints x, y, z and p, thus we have four octagons O_x, O_y, O_z and O_p. First, we check path $x \to y \to z$. Assume the UAV starts at direction 1 of O_x, *i.e.*, x_1, since O_y is within the direction index of x_1, there is an edge $e_{x_1y_1}$, according to Line 7 of Algorithm `ModelingTurns`. Then the UAV moves along $e_{x_1y_1}$ to reach y_1. Next the UAV restarts at y_1, through edge $e_{y_1z_1}$ to arrive at z_1 by the same logic. Note that there is no energy cost of turns on this straight path, and the energy consumed on acceleration/deceleration is modeled on the weight of two edges. Second, we check path $x \to y \to p$, where the UAV has to make a turn at y. Similarly, assume the UAV starts at x_1 and arrives at y_1 through edge $e_{x_1y_1}$. Subsequently, since O_p is outside the direction index of y_1, at first the UAV chooses the octagonal edges, *i.e.*, $e_{y_1y_2}$,$e_{y_2y_3}$ and $e_{y_3y_4}$, to make three 45° turns to arrive at y_4 according to the algorithm. Then it directly reaches the target p_4 through edge $e_{y_4p_4}$. In this case, the energy cost of making three 45° turns can be calculated by the three-edges weight of octagon. As a conclusion, our modeling of turns works correctly, the larger the angle of turning, the more sum of edge weight on the octagon.

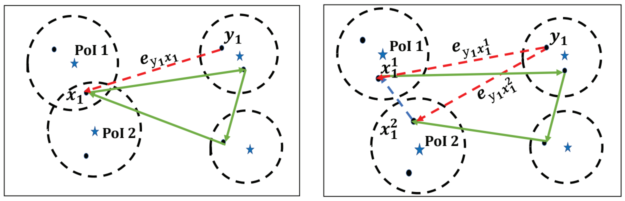
B. Modeling energy cost of switching PoI-visiting

In this subsection, we pay special attention to the switching energy cost of PoI-visiting. By the problem definition, PoI i can be visited if vertex $x_k \in \mathbb{S}_2$ is included in the path, $\forall x \in S_i, k = 1, 2, \cdots, 8$. However, if $D_x = \{i, j\}$, there is the switching cost at x, $E(D_x) = c_3(|D_x| - 1)$, which must be reflected by the path.

Our core idea is that, for any vertex in the common set of multiple PoIs, we split it into virtual vertex copies, one for each PoI. And assign the cost of switching to the weight of a connecting edge between any two copied vertices. This idea is inspired by [26]. Consequently, we convert graph $G_2 = (\mathbb{S}_2, \mathbb{E}_2, \mathbb{W}_2)$ into $G_3 = (\mathbb{S}_3, \mathbb{E}_3, \mathbb{W}_3)$. For any vertex $x_i \in \mathbb{S}_2$, set $V(x_i) = \{k | x \in S_k\}$ is all the PoIs that has x_i in its range. So any vertex $x_i \in \mathbb{S}_2$ is converted to $|V(x_i)|$ copies. And if $|V(x_i)| > 1$, we connect every two copies with an edge weighted c_3, representing one switching cost between involved PoIs. We present these steps formally in Algorithm `ModelingSwitch`.

```
Algorithm 2: ModelingSwitch ($G_2(\mathbb{S}_2, \mathbb{E}_2, \mathbb{W}_2)$)
1  for each $x_i \in \mathbb{S}_2$ and each $p \in \{k | x \in S_k\}$ do
2      $\mathbb{S}_3 = \mathbb{S}_3 \cup \{x_i^p\}$;
3      for each $q \in \{k | x \in S_k\}$ and $p \neq q$ do
4          $\mathbb{E}_3 = \mathbb{E}_3 \cup \{e_{x_i^p x_i^q}\}$;
5          $\mathbb{W}_3 = \mathbb{W}_3 \cup \{W(e_{x_i^p x_i^q}) = c_3\}$;
6      end
7  end
8  for each $e_{x_ix_{i+1}} \in \mathbb{E}_2$ and $p \in \{k | x \in S_k\}$ do
9      $\mathbb{E}_3 = \mathbb{E}_3 \cup \{e_{x_i^p x_{i+1}^p}\}$;
10     $\mathbb{W}_3 = \mathbb{W}_3 \cup \{W(e_{x_i^p x_{i+1}^p}) = W(e_{x_ix_{i+1}})\}$;
11 end
12 for each $e_{x_iy_i} \in \mathbb{E}_2$ do
13     for each $p \in \{k | x \in S_k\}$ and $q \in \{k | y \in S_k\}$ do
14         $\mathbb{E}_3 = \mathbb{E}_3 \cup \{e_{x_i^p y_i^q}\}$;
15         $\mathbb{W}_3 = \mathbb{W}_3 \cup \{W(e_{x_i^p y_i^q}) = W(e_{x_iy_i})\}$;
16     end
17 end
18 return $G_3(\mathbb{S}_3, \mathbb{E}_3, \mathbb{W}_3)$
```

An example is given in Fig. 5 to clarify some key steps in Algorithm `ModelingSwitch`. Assume Waypoint x_1 lo-



(a) (b)

Fig. 5. An example of modeling energy cost of switching PoI-visiting. In (a), switching visiting between PoIs also costs energy, which is proportional to the number of switching. It is hard to be represented in a graph. In (b), we separate the overlapping ranges by splitting waypoints into virtual copies, and assign the energy consumption of one switch to the weight of a connecting edge between two copied vertices .

cates in the overlapping range of PoI 1 and PoI 2. Algorithm ModelingSwitch, first separating the overlapping ranges by making two copied of x_1, *i.e.,* x_1^1 and x_1^2, and assigning them to PoI 1 and PoI 2 one for each, as displayed in Fig. 5(b). Then an edge is added to connect the two copied vertex (the blue dashed line in (b)), whose weight equals to the energy cost of one switching between the two PoIs. Moreover, every edge with an endpoint inside the overlapping range, such as edge $e_{y_1x_1}$, is replaced by two new edges (red dashed lines in (b)), $e_{y_1x_1^1}$ and $e_{y_1x_1^2}$, with the same weight as that of $e_{y_1x_1}$. Hereby, the switching cost is now reflected on the graph, and the switching cost can be easily modeled on the graph by summing up the weights of all involved edges.

C. Redefinition of the problem by a graph model

Since the turning cost and the switching cost are modeled by Algorithm ModelingTurns and ModelingSwitch in the previous subsections, we can redefine P1 by the generated graph $G_3(\mathbb{S}_3, \mathbb{E}_3, \mathbb{W}_3)$. Then we define a new directed weighted graph $D(\mathbb{S}, \mathbb{E}, \mathbb{W}, S')$, where $S' = \{S_1', S_2', \cdots S_n'\}$, and $S_i' = \{x_k^i | \forall x, k\}, i = 1, 2, \cdots, n$, is vertex set for PoI i. Note that, we have $S_i' \cap S_j' = \varnothing, \forall i \neq j$ by ModelingSwitch. Besides, the vertex set $\mathbb{S} = \mathbb{S}_3$, the edge set $\mathbb{E} = \mathbb{E}_3$, the weight set $\mathbb{W} = \mathbb{W}_3$.

Definition 5 (P2)**.** *Given a directed weighted graph $D = (\mathbb{S}, \mathbb{E}, \mathbb{W}, S')$, the* waypoint-based PoI-visiting *problem is to find a feasible tour in D to visit each subset S_i' at least once, while the sum of weights of all chosen edges is minimum.*

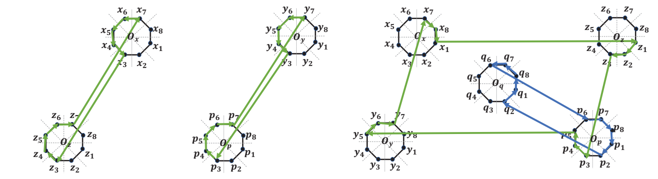
The set $\mathbb{E} = \{< v_i, v_j >\}$ includes all directed edges. The weight of directed edge $< v_i, v_j > \in \mathbb{E}$ is defined as c_{ij}, and $w_{ij}^{'}$ represents whether $< v_i, v_j >$ is in the flight route,

$$w_{ij}^{'} = \begin{cases} 1, & \text{edge } < v_i, v_j > \text{ is in the route,} \\ 0, & \text{otherwise.} \end{cases} \tag{9}$$

D. Mathematics formulation

This subsection describes the details of solving P2. Now we formulate P2 with the objective function and constraints:

$$Min \sum_{v_i, v_j \in \mathbb{S}, <v_i, v_j> \in \mathbb{E}} c_{ij} w_{ij}^{'} \tag{10}$$


x_6 x_7 x_5 o_x x_8 x_4 x_1 x_3 x_2
y_6 y_7 y_5 o_y y_8 y_4 y_1 y_3 y_2
z_6 z_7 z_5 o_z z_8 z_4 z_1 z_3 z_2
p_6 p_7 p_5 o_p p_8 p_4 p_1 p_3 p_2
q_6 q_7 q_5 o_q q_8 q_4 q_1 q_3 q_2

(a) (b)

Fig. 6. Two practical cases of subloop in a flight route.

s.t.

$$\left.\begin{array}{l} \sum_{v_i \in S_i, v_j \notin S_j} \sum_{<v_i, v_j> \in \mathbb{E}} w_{ij}^{'} \geq 1 \\ \sum_{v_i \notin S_i, v_j \in S_i} \sum_{<v_i, v_j> \in \mathbb{E}} w_{ij}^{'} \geq 1 \end{array}\right\} for\ all\ sets\ S_i \tag{11}$$

$$\sum_{v_i \in \mathbb{S}, <v_i, v_j> \in \mathbb{E}} w_{ij}^{'} - \sum_{v_k \in \mathbb{S}, <v_j, v_k> \in \mathbb{E}} w_{jk}^{'} = 0,$$
$$for\ all\ vertices\ v_j \in \mathbb{S} \tag{12}$$

$$\sum_{i \in G} \sum_{v_i \in S_i} \sum_{j \notin G} \sum_{v_j \in S_j, <v_i, v_j> \in \mathbb{E}} w_{ij}^{'} \geq 1,$$
$$for\ all\ sets\ G\ which\ are\ subsets\ of\ the\ collection\ of\ set\ \mathbb{S},$$
$$2 \leq |G| \leq n - 2 \tag{13}$$

$$w_{ij}^{'} \in \{0, 1\}\ for\ all < v_i, v_j > \in \mathbb{E} \tag{14}$$

1) Objective function: Our goal is to find a cycle to visit each subset at least once on the graph with the minimum sum of weights of all visiting-edges, denoted as the objective function in Eq. (10).

2) Constraints and transformation to GTSP: We intend to transform the problem P2 into the GTSP, defined as

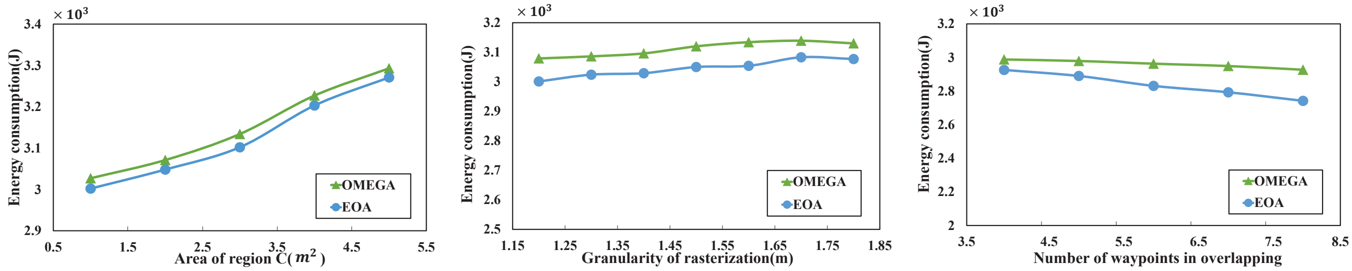
Definition 6 (GTSP)**.** *[16] Given a complete weighted graph $G = (V, E, w)$ on n vertices and a partition of V into m sets $P_V = \{V_1, ..., V_m\}$, where $V_i \cap V_j = \emptyset$ for all $i \neq j$ and $U_{i=1}^m V_i = V_j$, find a cycle in G that contains exactly one vertex from each set $V_i, i \in 1, ..., m$ and has minimum length.*

Three constraints are imposed to make the problem in Definition 5 equivalent to GTSP as follows.

- **Subset coverage**. The UAV has to visit each PoI at least once, which means in-edge and out-edge both necessarily exist in each subset, so Eq. (11) unfolds this constraint.
- **Tour continuity**. Each waypoint of the tour has the same in-degree as the out-degree to keep the tour continuous. We use Eq. (12) to guarantee the continuity.
- **Subloop avoidance**. As shown in Fig. 6(a), the tour is impracticable due to the possible subloops. Hence, the constraint as Eq. (13) is crucial to avoid this case.

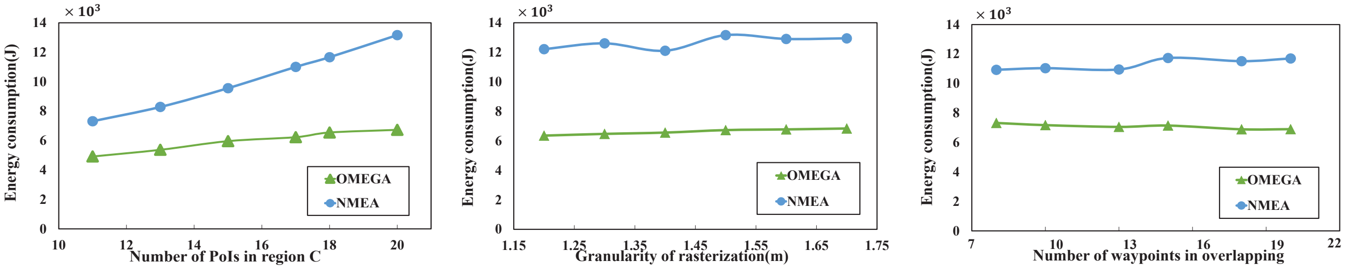
However, our solution is not rigorous enough. As shown in Fig. 6(b), a tour complies the three constraints but is infeasible (a subloop is marked in blue). To fix this little bug, we modify the constraint in Eq. (11) as follows:

$$\left.\begin{array}{l} \sum_{v_i \in S_i, v_j \notin S_j} \sum_{<v_i, v_j> \in \mathbb{E}} w_{ij}^{'} = 1 \\ \sum_{v_i \notin S_i, v_j \in S_i} \sum_{<v_i, v_j> \in \mathbb{E}} w_{ij}^{'} = 1 \end{array}\right\} \text{for all sets } S_i \tag{15}$$



(a) (b) (c)

Fig. 7. Algorithm performance comparison between OMEGA and EOA.



(a) (b) (c)

Fig. 8. Algorithm performance comparison between OMEGA and NMEA.

where the UAV visits each subset once and only once.

According to the modeling of making turns, the edges are formed only between two vertices in the same-index range. To get closer to GTSP, we add an extra operation to make the graph a complete one: for any two vertices without connection, assign them an edge, whose weight is equals to the sum of cost from one vertex to another. For example, two vertices, e.g., v_i and v_j, belong to different regular octagons without connection, so we create edge $< v_i, v_j >$, and its weight is the sum of cost from v_i to v_j.

3) Large neighborhood search based algorithm usage: Now we have converted the original problem to GTSP and there always exists a feasible tour, then we can expediently address our problem by referring to the Large Neighborhood Search based algorithm [27].

VI. Simulation

In this section, we conduct simulations to evaluate the performance of our flight planning algorithm, the Optimization of Minimum-Energy by Graph Algorithm (OMEGA). We use a brute-force method, the Enumerated Optimal Algorithm (EOA), to search for optimal solutions. We use the running time of EOA as a criterion to distinguish small scale problem from large scale problem, *i.e.,* if an optimal solution can be computed by EOA within 500s, it is called a small scale problem, otherwise, it is a large scale problem. For large scale problems, the Naive Minimum-Energy Algorithm (NMEA) is used for comparison.

A. Simulation settings

Based on our energy model of UAV flight introduced in the previous section, we set the straight cost to $25J$ each meter, the turning cost to $7.64J$ each degree, and the switching cost to $1200J$ each time. PoI number n, region size N, grid size g, and the number of candidate waypoint m are variables to be assigned different values to evaluate their impacts on algorithm performance. Then we run 50 times and take the average result of the 50 instances.

B. Results and discussion

1) Algorithm comparison between OMEGA and EOA: When the problem scale is small, we follow EOA to get the optimal result by enumerating all possible paths. The comparison between our OMEGA and EOA is shown in Fig. 7. As shown in Fig. 7(a), for both OMEGA and EOA, the lower the distribution density of PoIs, the more the UAV energy consumption. Because with a fixed number of PoIs, the larger region leads to the longer distance, indicating the UAV needs more energy to visit PoIs. In Fig. 7(b), as the granularity of rasterization gets coarser, *i.e.,* enlarge the grid, the energy consumption increasing. Enlarging the grid makes fewer candidate waypoints, so the route has to choose these non-optimal waypoints, leading to greater energy consumption. In Fig. 7(c), we can see that the number of candidate waypoints in overlap increases while the energy cost is less. This is because more candidate waypoints in overlap which means more optional routes for UAVs, more likely to get one that has the minimum energy cost. Obviously, the varying tendencies of the two curves are similar generally in these subfigures. To be specific, the performance of our OMEGA is close to that of EOA on the whole, whose error is no more than 107%.

2) Algorithm comparison between OMEGA and NMEA: When the problem scale is large, it is impossible to enumerate all routes due to the enormous time complexity. Hence we verify the efficiency of OMEGA by comparing it with the NMEA where only the covered flight distance is considered. The comparison between OMEGA and NMEA is shown in Fig. 8. From Fig. 8(a) and 8(b), we can draw a similar conclusion as Fig. 7(a) and 7(b) respectively: the lower the distribution density of PoIs or the coarser the rasterization of the region, the more energy consumption. However, as shown in Fig. 8(c), the curve of the NMEA implies that energy consumption is more even if we increase the number of overlapping waypoints. The explanation of this case is that NMEA always neglects the cost of both turning and switching, and prefers to select the overlapping waypoints to minimize energy cost. Whereas, from a global perspective, since the cost of turning and switching is a non-negligible part of the total cost, it may not be the best if flight distance is considered merely. In short, compared with NMEA, the high efficiency of our OMEGA is evident by saving nearly 50% of energy consumption.

VII. Conclusion

In this paper, we study the *waypoint-based PoI-visiting problem* for UAVs. With the investigation of previous related work, most existing flight models simplify the UAV energy consumption, motivating us to build a more practical and accurate one by a set of real-world experiments. Then we formulate a general problem with our energy model to match more application scenarios of UAVs. To address this problem, we propose a novel graph-based energy-efficient approach, utilizing a well-studied classic solution of GTSP to find a tour with the minimum cost. We conduct simulations by comparing with the best and the naive baseline respectively, to evaluate the performance of OMEGA. The final result shows that OMEGA is excellent-performance within 107% of the best which enumerates all possibilities, and nearly 50% of energy compared to the naive which considers covered distance only.

Acknowledgment

This work was supported by the National Key Research and Development Program of China Grant 2017YFB1003000, National Natural Science Foundation of China Grants 62072101, 61632008, 61972086, 62072102, and 61872079, the Fundamental Research Funds for the Central Universities, Jiangsu Provincial Key Laboratory of Network and Information Security Grant BM2003201, Key Laboratory of Computer Network and Information Integration of the Ministry of Education of China Grant 93K-9, and Collaborative Innovation Center of Novel Software Technology and Industrialization.

References

[1] J. Gong, T. Chang, C. Shen, and X. Chen, "Flight time minimization of UAV for data collection over wireless sensor networks," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 9, pp. 1942–1954, 2018.

[2] D. Vallejo, J. J. Castro-Schez, C. Glez-Morcillo, and J. Albusac, "Multi-agent architecture for information retrieval and intelligent monitoring by UAVs in known environments affected by catastrophes," *Eng. Appl. Artif. Intell.*, vol. 87, 2020.

[3] Q. Hu, Y. Cai, G. Yu, Z. Qin, M. Zhao, and G. Y. Li, "Joint offloading and trajectory design for UAV-enabled mobile edge computing systems," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 1879–1892, 2019.

[4] H. Huang and A. V. Savkin, "Viable path planning for data collection robots in a sensing field with obstacles," *Comput. Commun.*, vol. 111, pp. 84–96, 2017.

[5] S. Ahmed, A. Mohamed, K. A. Harras, M. Kholief, and S. Mesbah, "Energy efficient path planning techniques for UAV-based systems with space discretization," in *IEEE Wireless Communications and Networking Conference, WCNC 2016, Doha, Qatar, April 3-6, 2016*. IEEE, 2016, pp. 1–6.

[6] C. H. Liu, Z. Chen, J. Tang, J. Xu, and C. Piao, "Energy-efficient UAV control for effective and fair communication coverage: A deep reinforcement learning approach," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 9, pp. 2059–2070, 2018.

[7] R. Xiong and F. Shan, "Dronetank: Planning UAVs' flights and sensors' data transmission under energy constraints," *Sensors*, vol. 18, no. 9, p. 2913, 2018.

[8] Z. Huang, W. Wu, F. Shan, Y. Bian, K. Lu, Z. Li, J. Wang, and J. Wang, "Couas: Enable cooperation for unmanned aerial systems," *ACM Trans. Sens. Networks*, vol. 16, no. 3, pp. 24:1–24:19, 2020.

[9] M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, "Mobile unmanned aerial vehicles (UAVs) for energy-efficient internet of things communications," *IEEE Trans. Wirel. Commun.*, vol. 16, no. 11, pp. 7574–7589, 2017.

[10] A. Rahmati, X. He, I. Güvenç, and H. Dai, "Dynamic mobility-aware interference avoidance for aerial base stations in cognitive radio networks," in *2019 IEEE Conference on Computer Communications, INFOCOM 2019, Paris, France, April 29 - May 2, 2019*. IEEE, 2019, pp. 595–603.

[11] K. Dorling, J. Heinrichs, G. G. Messier, and S. Magierowski, "Vehicle routing problems for drone delivery," *IEEE Trans. Syst. Man Cybern. Syst.*, vol. 47, no. 1, pp. 70–85, 2017.

[12] F. Morbidi, R. Cano, and D. Lara, "Minimum-energy path generation for a quadrotor UAV," in *2016 IEEE International Conference on Robotics and Automation, ICRA 2016, Stockholm, Sweden, May 16-21, 2016*, D. Kragic, A. Bicchi, and A. D. Luca, Eds. IEEE, 2016, pp. 1492–1498.

[13] F. Shan, J. Luo, R. Xiong, W. Wu, and J. Li, "Looking before crossing: An optimal algorithm to minimize UAV energy by speed scheduling with a practical flight energy model," in *39th IEEE Conference on Computer Communications, INFOCOM 2020, Toronto, ON, Canada, July 6-9, 2020*. IEEE, 2020, pp. 1758–1767.

[14] "Mission Planning - Copter documentation," https://ardupilot.org/copter/docs/common-mission-planning.html, accessed Aug. 10, 2020.

[15] "Onboard SDK - DJI Developer," https://developer.dji.com/cn/onboard-sdk, accessed Aug. 10, 2020.

[16] S. Piao, Z. Ba, L. Su, D. Koutsonikolas, S. Li, and K. Ren, "Automating CSI measurement with UAVs: from problem formulation to energy-optimal solution," in *2019 IEEE Conference on Computer Communications, INFOCOM 2019, Paris, France, April 29 - May 2, 2019*. IEEE, 2019, pp. 2404–2412.

[17] C. Wang, F. Ma, J. Yan, D. De, and S. K. Das, "Efficient aerial data collection with UAV in large-scale wireless sensor networks," *IJDSN*, vol. 11, pp. 286 080:1–286 080:19, 2015.

[18] Q. Yang and S. Yoo, "Optimal UAV path planning: Sensing data acquisition over iot sensor networks using multi-objective bio-inspired algorithms," *IEEE Access*, vol. 6, pp. 13 671–13 684, 2018.

[19] Y. Zeng, J. Xu, and R. Zhang, "Energy minimization for wireless communication with rotary-wing UAV," *IEEE Trans. Wirel. Commun.*, vol. 18, no. 4, pp. 2329–2345, 2019.

[20] X. Cao, J. Xu, and R. Zhang, "Mobile edge computing for cellular-connected UAV: computation offloading and trajectory optimization," in *19th IEEE International Workshop on Signal Processing Advances in Wireless Communications, SPAWC 2018, Kalamata, Greece, June 25-28, 2018*. IEEE, 2018, pp. 1–5.

[21] T. Zhang, Y. Xu, J. Loo, D. Yang, and L. Xiao, "Joint computation and communication design for UAV-assisted mobile edge computing in iot," *IEEE Trans. Ind. Informatics*, vol. 16, no. 8, pp. 5505–5516, 2020.

[22] J. Zhang, L. Zhou, Q. Tang, E. C. H. Ngai, X. Hu, H. Zhao, and J. Wei, "Stochastic computation offloading and trajectory scheduling for UAV-assisted mobile edge computing," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 3688–3699, 2019.

[23] H. Guo and J. Liu, "UAV-enhanced intelligent offloading for internet of things at the edge," *IEEE Trans. Ind. Informatics*, vol. 16, no. 4, pp. 2737–2746, 2020.

[24] Y. Bouzid, Y. Bestaoui, and H. Siguerdidjane, "Quadrotor-UAV optimal coverage path planning in cluttered environment with a limited onboard energy," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2017, Vancouver, BC, Canada, September 24-28, 2017*. IEEE, 2017, pp. 979–984.

[25] C. H. Liu, Z. Chen, and Y. Zhan, "Energy-efficient distributed mobile crowd sensing: A deep learning approach," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 6, pp. 1262–1276, 2019.

[26] C. Noon and J. Bean, "An efficient transformation of the generalized traveling salesman problem," *INFOR. Information Systems and Operational Research*, vol. 31, 02 1993.

[27] S. L. Smith and F. Imeson, "GLNS: an effective large neighborhood search heuristic for the generalized traveling salesman problem," *Comput. Oper. Res.*, vol. 87, pp. 1–19, 2017.