Looking before Crossing: An Optimal Algorithm to Minimize UAV Energy by Speed Scheduling with a Practical Flight Energy Model

Feng Shan, Junzhou Luo, Runqun Xiong, Wenjia Wu, Jiashuo Li
School of Computer Science and Engineering, Southeast University, Nanjing, China
{shanfeng, jluo, rxiong, wjwu, jiashuoli}@seu.edu.cn

***Abstract*—Unmanned aerial vehicles (UAVs) are being widely used in wireless communication, *e.g.,* collecting data from ground nodes (GNs), where energy is critical. Existing works combine speed scheduling, *i.e.,* the controlling of speed, with trajectory design for UAVs, making it complicated to solve while loses focus on the fundamental nature of speed scheduling. We focus on speed scheduling by considering straight line flights, with applications in monitoring power transmission lines, roads, water/oil/gas pipes and rivers/coasts. By real-world flight tests, we disclose a speed-related flight energy consumption model, distinct from typical distance-related or duration-related models. Based on such a practical energy model, we develop the *looking before crossing (virtual rooms)* algorithm, where virtual rooms on the time-distance diagram represent the spatio-temporal constraint of GNs in wireless transmission. This algorithm is proved to be optimal in solving the offline problem, where all information is known before scheduling. For the online problem, *i.e.,* GN information is not unavailable unless flies close, we propose an offline-inspired online heuristic. Simulation shows its performance is near the offline optimal. Our study on the practical flight energy model and speed scheduling sheds light on a new research direction on UAV-aided wireless communication.**

***Index Terms*—Unmanned Aerial Vehicle, Energy Efficient, Speed Scheduling, Practical Energy Model, Looking Before Crossing, Offline Optimal Algorithm**

I. Introduction

Unmanned aerial vehicles (UAVs) are being widely adopted in wireless communication research community [1]–[6], as wireless base stations [3], wireless relays [4], and used in edge computing [5], data collection [6]. Wireless sensors and Internet of Things (IoT) devices are widely deployed for various monitoring purposes nowadays. Data collection from such ground nodes (GNs) using UAVs is one of the most important applications, because UAVs can fly close to establish line-of-sight energy-efficient data communications. There are already tremendous research efforts in this direction [6]–[14].

Energy consumption of UAVs is one of the key issues, because there is normally limited energy supply on board. Hence, it is important to carefully manage energy consumption and also important to model the flight energy consumption. Most existing works consider two classic UAV energy models, *e.g.,* the distance-related energy consumption model [7], [8], [12] and the duration-related model [6], [9], [15]. In the distance-related model, the energy consumption is assumed to be proportional to the distance that UAV covers; while in the duration-related model, it is assumed to be proportional to the duration of flight. However, both models simplify the UAV energy consumption and do not accurately reflect the energy consumption during flight for indepth investigations.

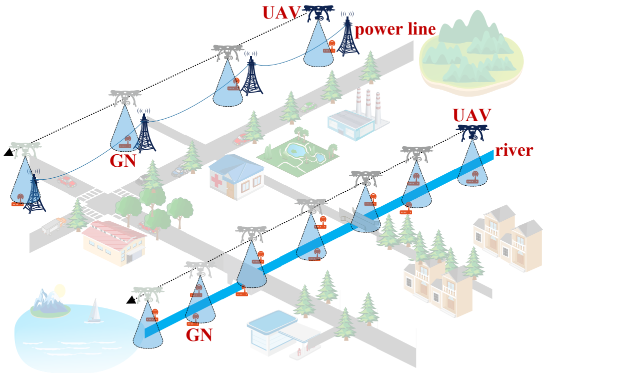


Fig. 1. The application scenarios. A UAV is dispatched to collect data from a set of wireless sensors or IoT devices (GNs) deployed along a straight line, such as a power transmission line, a road, a water/oil/gas pipe or a river/coast.

Therefore, we conduct a set of real-world flight tests using a multi-rotor UAV. In our on-site flight tests, a practical speed-related energy consumption model is disclosed. In this model, the flight power is related to the flight speed: the UAV has a particular speed at which its power consumption is lowest; at both higher and lower speeds it consumes more power. Details will be discussed in the next section. Such model has been verified by a most recent theoretical analysis work on energy model of rotary-wing UAVs [11]. As a result, we conclude that our speed-related model is more practical and more general than most existing energy consumption models used for wireless communication.

The most related work is by Zeng *et al.* [11]. After proposing a theoretical energy model, they study the UAV energy consumption minimization problem for collecting data from GNs deployed in a given area. Their formalized problem is difficult to solve, because besides the speed scheduling, the UAV flight trajectory also needs to be determined. As a result, they obtain a heuristic solution with uncertain difference towards the optimal solution. Instead of an area, we focus on deploying GNs along a straight line, which is with lots of applications as well, for example, monitoring power transmission lines, roads, water/oil/gas pipes or river/coast. See Fig. 1 as an illustration of the application scenarios. Besides, we consider only the flight energy consumption, and ignore energy consumption for wireless data transmission. This is because, according to our real-world flight tests, the power for our UAV hover is around $400W$, while a typical LoRa/Wi-Fi wireless communication module's power consumption is $100mW$ [16] / $300mW$ [17]. The wireless transmission consumption is a thousand times smaller, thus it is ignored to let our problem focus on tracing the fundamental nature of speed scheduling.

The readers can sense such fundamental nature of speed scheduling from the following challenges to our problem.

1) The UAV flight energy must be minimized to collection all data: on the one hand, if the UAV flies at a slower speed, it has enough time to collect all data however slower speed consumes more energy according to our practical energy model; on the other hand, a faster-speed flight may reduce the flight energy consumption but may cause GN data not being completely collected since there may be not enough time in transmission range.
2) The transmission ranges of GNs may overlap each other and the UAV collects data from one GN at a time. As a result, every GN competes for UAV time to deliver its own data. Moreover, each GN has a different amount of data to transmit and a different transmission range size, such competition is rather complicated.

The contributions of this paper are summarized as follows.

- We adopt a practical speed-related energy consumption model based on our real-world flight tests, and the results show that the UAV has a particular speed in which its power consumption is lowest; at both higher and lower speeds it consumes more power. This model is distinct to most existing works that assumes either the distance-related or duration-related model.
- We propose an optimal algorithm named *looking before crossing (virtual rooms)* to solve the offline problem. Since each GN has spatial limits transmission range and temporal requirements on transmission time, we propose *virtual rooms* to represent these spatio-temporal constraints on a time-distance diagram. A trajectory crossing these virtual rooms can be uniquely mapped to a solution. It is proved that by looking before crossing, an optimal solution can be determined.
- We propose a heuristic inspired by the offline algorithm to solve the online problem where GN information is not unavailable unless the UAV flies close. Simulation results show its performance is quite close to the performance of the offline optimal solution, the online energy consumption is within 102% of the offline consumption. Our study on the speed scheduling and practical flight energy model shed light on a new research direction on UAV-aided wireless communication.

The rest of the paper is organized as follows. Section II introduces the flight energy model. Section III presents the system model and the problem formulation. The *crossing-the-rooms* problem and its optimal properties are introduced in Section IV. Special cases and the general problem are studied in Section V and Section VI, respectively. Then, the online policy and simulations are described in section VII. Finally, section VIII concludes the paper.

II. Flight Energy Consumption Model in Data Collection

In this section, we first introduce two classic flight energy consumption models and study the related works in the direction of UAV-aided wireless communciation. Then, we present a practical speed-related model.

A. Distance-related and duration-related flight energy models

In the distance-related flight energy consumption model, the energy consumption is assumed to be proportional to the distance a UAV covers. Liu *et al.* [8] adopt this model and consider the UAV as a airborne base stations to serve ground user devices within a given region. A team of UAVs must move according to the data collection demands while stay connected. Piao *et al.* [7] study an indoor CSI measurement problem in which a drone has to fly to each of the measurement point in a given floor of a building. The energy of the drone depend on the distance covered and the number of turns made. Xiong *et al.* [12] assume that both distance and turn number affect the UAV energy consumption, so they design a dynamic programming approach to reduce the number of turns and total flight distance while guarantee to fly close to GNs to collect data.

In the duration-related flight energy consumption model, the energy consumption is assumed to be proportional to the duration of flight. Mozaffari *et al.* [9] investigate the mobility of UAVs to collect data from ground IoT devices. The flight power is assumed to be constant, therefore the energy consumption is related to flight duration. Then, a 3D trajectory planning problem is solved. Rahmati *et al.* [15] adopt this model and study the problem to find good locations so as to better relay the data from the designated sources to destinations. Gong *et al.* [6] focus on the UAV flight time minimization problem for data collection over a linearly deployed GNs. Their proposed algorithms imply that the energy consumption solely related to the flight time.

B. A practical speed-related flight energy model

However, both models simplify the UAV energy consumption. We want to answer the following question: how does the flight speed relate to the energy consumption?

Therefore, we perform a set of real-world on-site flight tests to understand the relationship between the flight speed and the power consumption. In our flight tests, we use a 2 kgs hexacopter drone with setting similar to the flight tests conducted in [18]. Specifically, a flight controller Pixhawk 3.6.5 is installed on this hexacopter drone, which is connected to a companion computing device, Raspberry Pi 3b single-board computer (RPi). By MAVLink protocol, the Pixhawk controller keeps sending UAV battery voltage information

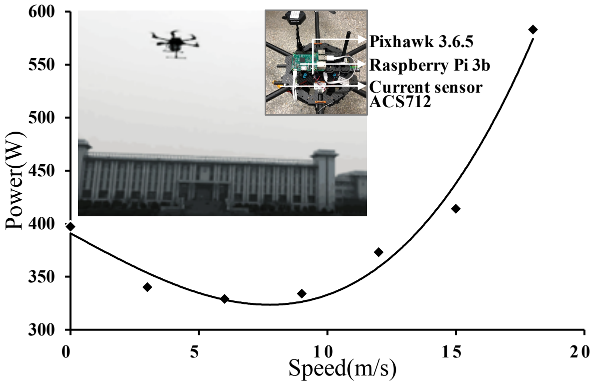


Fig. 2. A practical speed-related energy consumption model based on our real-world flight test. The power consumption is related to the UAV speed via a convex function, *e.g.,* it first decreases and then increases as speed increases.

to the RPi. We further install a current module ACS712 to monitor real-time UAV battery current values, which are read by the RPi via an I^{2}C communication protocol. With both the voltage and current values collected, it is easy to compute the power consumption of the UAV.

In our flight tests, we let the drone fly along a straight line for no longer than 1000m, and we vary the speed from 0m/s (hover) to 18m/s with step 3m/s. Fig. 2 shows a UAV during our flight test. Flight test on each speed is repeated 10 times and the mean value is used to alleviate anomalies from individual trials. The experiment results are presented in Fig. 2, which provides a comprehensive understanding on the relationship between flight speed and the UAV power consumption.

The results clearly show that neither the distance-related energy model nor duration-related energy model holds. The flight power is a convex function of the flight speed, more specifically, the flight power of UAV first decreases and then increases as flight speed increases. We also notice that our findings are consistent with measurement results of pioneer researchers [19]. Most recently, such model has been verified by a theoretical analysis [11]. A similar energy model is discovered by measurement in [**?**] as well. As a conclusion, the speed-related energy consumption model is more practical and more general than other existing models used for wireless communication.

III. System Model and Problem Formulation

We assume a set of GNs, *e.g.,* wireless sensors and IoT devices, are unevenly distributed along a straight line, such as a power transmission line, a road, a water/oil/gas pipe or a river/coast, to perform monitoring task and sensing data. There are n GNs and they are indexed according to their locations, as GN $i, i = 1, 2, \cdots, n$. A UAV flies over such straight line to collect sensed data from these GNs. The UAV is assumed to fly at a fixed height starting from an initial position towards a destination position. The UAV can fly slow or fly fast, but it can never fly back. We use an axis to represent such straight line flight path, as in Fig. 3. Without loss of generality, we assume initial position is at the original and d on the axis represents distance d to the initial position.

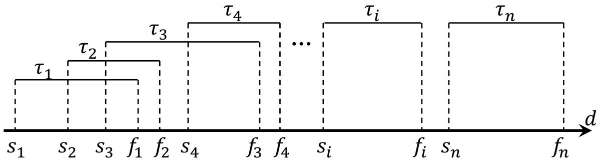


Fig. 3. Each GN i requires a minimum time τ_i for data transmission to the UAV and the UAV collects data from one GN at a time. The problem is to determine the speed of UAV, such that each GN i has enough time to upload data within its transmission range (s_i, f_i) and the UAV flight energy is minimized. On the one hand, if the UAV flies slow, it has time to collect data yet consumes more flight energy; on the other hand, fly fast may reduce energy consumption but shorten transmission time and may cause incomplete collection. Therefore, the best trade-off must be found.

Every GN has two transmission rates to work on: one is faster and the other slower. If the faster rate is adopted, the corresponding transmission range is smaller. This rate is used for data collection, and its range is called *data transmission range*. While the slower rate is with a larger transmission range, and is used to deliver control information, for example initialize connect and prepare for data transmission. Let the *data transmission range* of GN i be denoted as (s_i, f_i). So a UAV can collect data if $s_i \leq d \leq f_i$, where d represent the location of the UAV. Assume $s_1 = 0$. s_i is called *starting position* and f_i is called *ending position*. Each GN has a certain amount of data waiting to be collected. Let the minimal time require for the UAV to finish data collection from GN i be denoted as τ_i, which can be calculated given the data amount and the data transmission rate of GN i. Note that τ_i does not necessarily equal between GNs.

Assume GNs are heterogeneous, so they have different but aligned transmission ranges. That is, the range size $(f_i - s_i)$ varies from GN to GN. Assume $0 = s_1 < s_2 < \ldots < s_n$, then we have $0 < f_1 < f_2 < \ldots < f_n = D$. This is because range sizes are similar and no two GNs are placed closely to monitor the same location. Note that, $d = 0$ and $d = D$ is the UAV initial position and destination position. An example of the settings are given in Fig. 3. We assume every two adjacent GNs have overlap transmission range, otherwise there is a gap between them so this problem can be divided into two smaller independent sub-problem, because the speed schedule before and after this gap does not affect each other. There are n *starting positions* and n *ending positions* in total. Assume no two points overlap together, if they do, we can treat them being separated by an extremely small distance.

The UAV speed scheduling is represented by the *speed scheduling function*, that determines the UAV flight speed for any given time t, denoted as $v(t)$. The position of the UAV, *e.g.,* the distance from the initial point on the axis, can therefore be denoted by the following integration.

$$d(t) = \int_0^t v(\tau)\, \mathrm{d}\tau. \tag{1}$$

Function $d(t)$ is called the *distance accumulation function*, which is obviously a continuous monotonically increasing function.

We assume the UAV collects data from GNs when it is flying, but it collects from one GN at a time because they share the same communication channel. Since the transmission ranges are overlapped but aligned, it is clear that the UAV collects data from GN in the order of their locations, *e.g.,* following their index order. Therefore, there must be a switching time at which the UAV finishing collecting data from GN i and starting to collect from GN $i+1$. Let t_i denote such a switching time. There are all together $n-1$ switching time among the n GNs. Then, t_{i-1} is named the *collection starting time* and t_i named the *collection finishing time* for GN i, $i=1,\cdots,n$. Because the collection must be within its transmission range (s_i, f_i), we must have the following *range constraint*,

$$s_i \leq d(t_{i-1}) < d(t_i) \leq f_i, \qquad \forall i. \tag{2}$$

Because τ_i is the minimal transmission time require for the UAV to finish data collection from GN i, Hence, we have the following *completion constraint*,

$$t_i - t_{i-1} \geq \tau_i, \qquad \forall i. \tag{3}$$

Let $p(v)$ denote the flight power for a given UAV flight speed v. The energy consumption of UAV, denoted by E, can be calculated by the following equation.

$$E = \int_{t_0}^{t_n} p(v(t))\,\mathrm{d}t. \tag{4}$$

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

Definition 1 (USS-GTS problem)**.** *Given a set of GNs and models mentioned above, the UAV speed scheduling and GN transmission switching (USS-GTS) problem is to find a UAV* speed scheduling function $v(t)$ *and* transmission switching times $t_i, i=0,1,\cdots,n,$ *such that the energy consumption in Eq.(4) is minimized while* range constraint *Eq. (2) and* completion constraint *Eq. (3) are satisfied.*

This problem is called *offline problem* if all information is known before schedule; it is called *online problem* if GN information is not unavailable unless the UAV flies into the control information delivery range. We call the *speed scheduling function* in an optimal solution of the offline problem the *optimal speed scheduling function*, which is denoted as $v^{opt}(t)$.

IV. Crossing-the-Rooms Problem and Some Optimal Properties

In this section, we first redefine the USS-GTS problem, mapping it to the crossing-the-rooms problem, and then provide some interesting optimal properties for it.

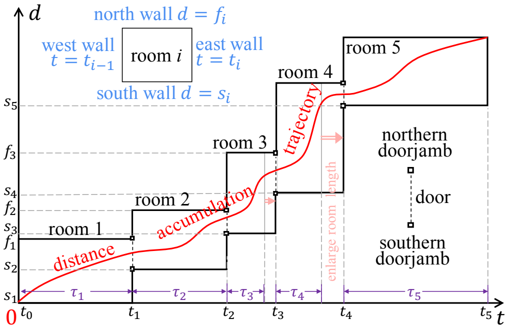


Fig. 4. The *distance accumulation trajectory* $d(t)$ crosses the rooms. For GN i, we draw a rectangle (virtual) Room i: north wall $d = f_i$, south wall $d = s_i$, west wall $t = t_{i-1}$, east wall $t = t_i$, while $t_i - t_{i-1} \geq \tau_i$. A feasible trajectory $d(t)$ must cross through all these room and pass through doors. The *crossing-the-rooms* problem essentially asks two questions: (1) how to construct the rooms, especially how to determine the length of each room, (2) how to design the trajectory crossing all the rooms and passing through doors, such that the UAV energy is minimized.

A. Crossing-the-Rooms *problem*

We first introduce the time-distance diagram. On a time-distance diagram, any point (t, d) represents reaching position d at time t. Originally, at $t = 0$, the UAV is at $d = 0$. Therefore, the *distance accumulation function* $d(t)$ can be plotted as a curve on this diagram, named as *distance accumulation trajectory*, which starts at the origin $(0, 0)$. The *speed scheduling function* $v(t)$ is essentially the slope of the *distance accumulation trajectory*. Hence, finding the optimal UAV speed scheduling is equivalent to finding the optimal distance accumulation trajectory on a time-distance diagram.

As shown in Fig. 4, such trajectory does not go freely on the diagram, it has constraints. These constraints come from the *range constraint* and the *completion constraint* imposed on the *speed scheduling function* $v(t)$. More specifically, the *range constraint* indicates any GN i has a spatial constraint on limited transmission range, between $d = s_i$ and $d = f_i$; and the *completion constraint* indicates a temporal constraint on required transmission times, between $t = t_{i-1}$ and $t = t_i$, where $t_i - t_{i-1} \geq \tau_i$. Consequently, for GN i, we draw a rectangle Room i: the north wall at $d = f_i$, the south wall at $d = s_i$, the west wall at $t = t_{i-1}$, the east wall at $t = t_i$. Since t_{i-1} and t_i are the *collection starting time* and *collection finish time* for GN i, between which the UAV must fly within transmission range (s_i, f_i), so the distance accumulation trajectory must be within such rectangle region. Therefore, we construct a serial of virtual rooms for a set of GNs, and Room i is with width (Y-axis) $(f_i - s_i)$ and length (X-axis) $(t_i - t_{i-1})$. Between two adjacent rooms, Room i and $i+1$, there is a *Door* i connecting them. The door is at $t = \sum_{j=1}^{i} \tau_j$ and with size $(s_{i+1} - f_i)$, as shown in Fig. 4. A feasible *distance accumulation trajectory* must cross through all rooms and pass through doors.

The *crossing-the-rooms* problem essentially asks:
(1) how to construct the rooms, especially how to determine the length of each room, and

(2) how to design the trajectory crossing all the rooms and passing through doors,
such that the UAV energy is minimized. Note that the shape of this trajectory directly determines the energy consumption because a convex function relates the flight speed and the power consumption.

Solving the *crossing-the-rooms* problem is equivalent to solving the USS-GTS problem. This is because the first question is equivalen to asking the GN transmission switching times; while the second question is equivalen to asking the UAV speed scheduling function. We will introduce the *looking before crossing* rooms algorithm to optimally solve the crossing-the-rooms problem which is inspired by the data flow model in [20]. The solution is uniquely mapped to the solution for the original USS-GTS problem.

B. Some optimal properties

An immediate lemma follows directly from the above discuss.

Lemma 1. *A* feasible distance accumulation trajectory *must be within the rooms.*

We want to find amongst all feasible trajectories the one with the minimal energy consumption.

Lemma 2. *For any given two time intervals, the UAV consumes the minimum energy if and only if a common flight speeds is used for both time intervals (if allowed).*

Proof. See Appendix A □

We have the following theory on the trajectory as a direct result from Lemma 2.

Theorem 1. *The optimal trajectory is straight between any two points, as long as this is feasible.*

Any non-straight trajectory between two points can be *straighten* to have the same slope, *i.e.,* speed, to save energy. This method is called *straightening*.

From Fig. 2, it can be easily seen that there is a speed at which the power consumption is the lowest. Let such speed denoted as $v^{\#}$. Note that, flying at $v^{\#}$ only means that the UAV consumes the minimum power. The flight energy consumption is not necessarily minimized, because speed $v^{\#}$ may be slow and cause a long duration to cover a certain distance. Therefore, the next lemma introduces a speed v^* that minimizes the energy consumption.

Lemma 3. *For any given flight distance, the UAV consumes the minimum energy if and only if it flies at speed $v = v^*$, where v^* is a constant as long as $p(v)$ is given and fix.*

Proof. See Appendix B. □

We have the following theorem on the trajectory.

Theorem 2. *Any point of the optimal trajectory has a slope no larger than v^*.*

Proof. See Appendix C. □

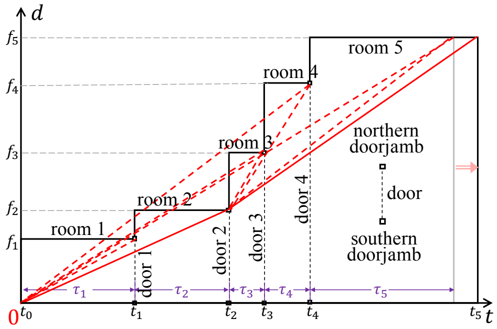


Fig. 5. The common starting position case. We first construct n rooms with minimal time as their length, *i.e.,* Room i with length τ_i. Then, standing at the origin, we look to the east. If multiple northern doorjambs is in view, we choose the farthest one and cross the rooms to walk there. Standing at the new position, we repeatedly look east and choose the farthest doorjamb as the next stop, until we reach the destination. In case trajectory slope is higher than v^* in some rooms, we reconstruct these rooms by enlarging the length such that the slope equals v^*.

V. Common Starting and Ending Position Cases

How to construct all these rooms and how to design an optimal trajectory crossing these rooms, they are still very challenge questions. To obtain some useful insights, this section studies two special cases, *i.e.,* the common starting position case and the common ending position case.

A. The common starting position case

In this case, every GN has the same transmission range starting position, $s_1 = s_2 = \cdots = s_n = 0$. But their ending positions are different: $0 < f_1 < f_2 < .. < f_n = D$. This special case is referred as the *USS-GTS problem with common starting position*. Before we present the *looking before crossing* virtual rooms algorithm that produces the optimal trajectory, we would like to introduce some properties of the optimal trajectory.

Lemma 4. *The optimal trajectory changes its direction only by increasing slope.*

Proof. See Appendix D □

Lemma 5. *The optimal trajectory changes its direction only at a northern doorjamb.*

Proof. See Appendix E □

The optimal trajectory therefore can be easily determined if all the optimal doorjambs at which the trajectory changes are known. But how to determine these doorjambs? We present the *looking before crossing* technique in three phases.

In the first phase, set length for each room i as τ_i, the minimal value. Obviously, these rooms have their south wall at the same location $d = 0$, while north walls are different, $d = f_i$. There is no need to consider the south wall and southern doorjamb. The east wall of Room i (west wall of Room $i+1$) is at $t = \sum_{j=1}^{i} \tau_j$, as shown in Fig. 5.

In the second phase, *i.e.,* the main steps, the core idea is quite simple. Standing at the current position (initially at the origin), look to the east. Multiple northern doorjambs may be in view. We choose the farthest northern doorjamb in view and cross the rooms to walk there. Standing at the new position, we repeatedly look east and choose the farthest doorjamb as the next stop. Assume the northeast corner of the last room is a virtual northern doorjamb, this procedure stops until no room to cross.

In the third phase, we modify and reconstruct some rooms if the slope is higher than v^*. The room lengths are enlarged such that the slope equals v^*.

We present the formal steps of this method in Algorithm USS-GTS-coStart.

Algorithm 1: USS-GTS-coStart

```
1  $k = 0, t_0 = 0, f_0 = 0$;
2  for $i = 1$ to $n$ do  $t_i = t_{i-1} + \tau_i$ ;
3  while $k < n$ do
4      $k_v^n = \arg\min_{k<i\le n}(f_i - f_k)/(t_i - t_k)$;
5      $v_v^n = (f_{k_v^n} - f_k)/(t_{k_v^n} - t_k)$;
6      if $v_v^n > v^*$ then break ;
7      Connect $(t_k, f_k)$ and $(t_{k_v^n}, f_{k_v^n})$;
8      $k = k_v^n$;
9  end
10 if $k < n$ then
11     $t_n = (f_n - f_k)/v^* + t_k$;
12     Connect $(t_k, f_k)$ and $(t_n, f_n)$;
13 end
```

These three phases are at Line 2, Line 3-9 and Line 10-13, respectively. It can be seen that the second phase, *i.e.,* the **while** loop, is the main phase. In each **while** iteration, one piece of the trajectory is calculated by finding the next stop point. k_v^n at Line 4 is such point and v_v^n at Line 5 is the trajectory slope. After the first changing point is found, the next stop point can be computed using the same method, but from the new position. Note that if the computed speed is larger than v^* as in Line 6, the third phase begins.

Theorem 3. *Algorithm* USS-GTS-coStart *produces the optimal* distance accumulation trajectory *for the offline USS-GTS problem with common transmission range starting position within $O(n^2)$ steps.*

Proof. See Appendix F. □

B. The common ending position case

We study another special case in this subsection and obtain more properties on the optimal trajectory. In the common ending position case, every GN has the same transmission range ending position, $f_1 = f_2 = \cdots = f_n = D$. This is called the *USS-GTS problem with common ending position.*

Similar to previous special case, we have the follow properties for the optimal trajectory. The proofs are similar to previous ones, and are left to the readers because of space limitation. It is suggested to draw a graph for this case similar to Fig. 5 to help understand the following two lemmas.

Lemma 6. *The optimal trajectory changes its direction only by decreasing slope.*

Lemma 7. *The optimal trajectory changes its direction only at a southern doorjamb.*

We apply the *looking before crossing* technique to find the optimal *distance accumulation trajectory* in three phases. We first construct rooms with minimal length. Then, we find the next stop by looking east and choose the farthest southern doorjamb. We then cross the rooms to walk there and repeatedly looking and crossing. Assume the northeast corner of the last room is a virtual southern doorjamb, this procedure stops until no room to cross. It is possible that in some rooms, the trajectory has slope large than v^* during the process. Suppose this happens for the first k rooms, then we enlarge these rooms in length such that the trajectory has slope equals v^*.

A formal algorithm and its proof is omitted because of the space limitation. They are similar to the previous algorithm and Theorem 3.

VI. Looking-before-Crossing Algorithm for General Case

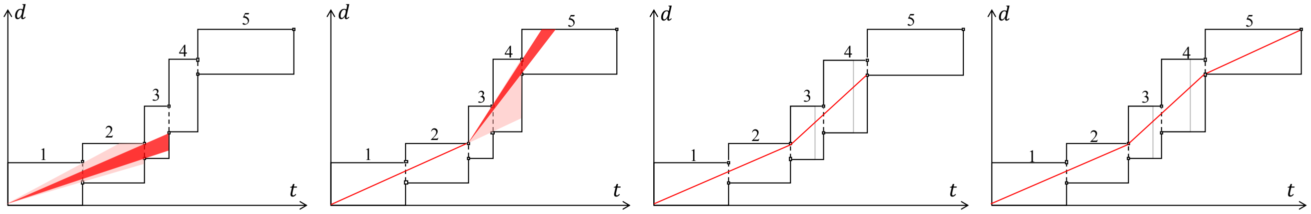
In the general case, there is no restrict on the range starting positions and ending positions. As a result, the optimal trajectory changes its direction by both increasing and decreasing slope. The following lemma states how it changes.

Lemma 8. *The optimal trajectory changes its direction only at doorjambs: increasing slope at a northern doorjamb, or decreasing slope at a southern doorjamb.*

This lemma is an combination of Lemma 4, 5, 6 and 7, therefore the proof is similar and left for the readers.

The *looking before crossing* algorithm produce the optimal trajectory in three phases. In the first phase, rooms are constructed, *i.e.,* each Room i with length τ_i, as shown in Fig. 4. Note that, each room may have a different south and north walls. In the second phase, we find a walking trajectory crossing all the rooms, starting at the origin and ending at the northeast corner. In the third phase, some rooms are reconstructed by enlarging their length. Note that, when a room length is enlarged, all the rooms on the east are moved accordingly. The first and third phase have been discussed in details in previous section.

The high level idea of the second phase, *i.e.,* the main part of the *looking before crossing* algorithm, is as follows. It is obvious that, if standing at the origin, we can view directly (through doors) the northeast corner, then this straight line is the optimal trajectory. However, if the view is blocked by walls, we need to find other way around. It is clear that the view angle is narrowed door after door. After looking through the first door, the northern boundary of the view angle is by the northern doorjamb and the southern boundary by the southern doorjamb. The view angle is further narrowed after looking through more doors, until it is blocked entirely. Suppose, standing at the origin, we can see as far as Room i, see part of its wall. Door i may be on the northern/southern side of the seen wall, we then walks along the north/south



(a) View ranges at the origin (b) New view ranges (c) Reconstruct rooms (d) The optimal trajectory

Fig. 6. *Looking before crossing* rooms. In (a), standing at the origin, the view angle through door 1 is in pink and angle through door 2 is in red. No view through door 3, which is beyond the northern boundary of the current (red) view angle. Therefore, we walk along the northern boundary, reaching the farthest doorjamb. In (b), standing at the new position, two new view angles are in pink and red respectively. Since the northeast corner is beyond southern boundary of the current view angle, we walk along the southern boundary to the farthest doorjamb. In (c), it can be seen that the lasted trajectory slope, in room 3 and 4, is larger than v^*, hence we enlarge the lengths of room 3 and 4 to reduce the slope to v^*. In (d), all room lengths and trajectory are determined.

Algorithm 2: USS-GTS-GENERAL

```
$k = 0, d = 0, t_0 = 0$;
$s_{n+1} = f_n$ // dummy for loop purpose
while $k < n$ do
    $v_n^v = \infty$, $v_s^v = 0$;
    for $j = k + 1$ to $n$ do
        $t_j = t_{j-1} + \tau_j$, $v_n^d = (f_j - d)/(t_j - t_k)$,
          $v_s^d = (s_{j+1} - d)/(t_j - t_k)$;
        if $v_s^d > v_n^v$ then
            $v_m = v_n^v$, $k_m = k_n^v$, $d_m = f_{k_n^v}$;
            break;
        else if $v_n^d < v_s^v$ then
            $v_m = v_s^v$, $k_m = k_s^v$, $d_m = s_{k_s^v+1}$;
            break;
        end
        if $v_n^v > v_n^d$ then $v_n^v = v_n^d$, $k_n^v = j$ ;
        if $v_s^v < v_s^d$ then $v_s^v = v_s^d$, $k_s^v = j$ ;
    end
    if $v_n^v == v_s^v$ then $v_m = v_n^v$, $k_m = n$, $d_m = f_n$ ;
    if $v_m > v^*$ then
        $x = d$;
        for $i = k + 1$ to $k_m$ do
            $t_i = \max\{(s_i - x)/v^*, \tau_i\} + t_{i-1}$;
            $x = x + (t_i - t_{i-1})v^*$;
        end
    end
    Connect $(t_k, f_k)$ and $(t_{k_m}, d_m)$;
    $d = d_m$, $k = k_m$;
end
```

boundary of the view angle, reaches and stops at the farthest doorjamb. Standing at the new position, the *looking before crossing* strategy repeats in the same way. Eventually, there is no room to cross.

We present the formal steps of this method in Algorithm USS-GTS-GENERAL.

Theorem 4. *Algorithm* USS-GTS-GENERAL *produces the optimal* distance accumulation trajectory *for the offline USS-GTS problem within $O(n^2)$ steps.*

Proof. See Appendix G. □

An example of the proposed *looking before crossing* algorithm is illustrated in Fig. 6.

VII. ONLINE HEURISTIC AND SIMULATIONS

A. *Online Heuristic*

In the previous section, an offline algorithm has been proposed to compute the optimal UAV speed scheduling and GN transmission switching for the USS-GTS problem. However, it is based on all GN information, including data transmission range starting position, ending position and the required transmission time. In practical, it is not always possible for the UAV to know all the GN information beforehand.

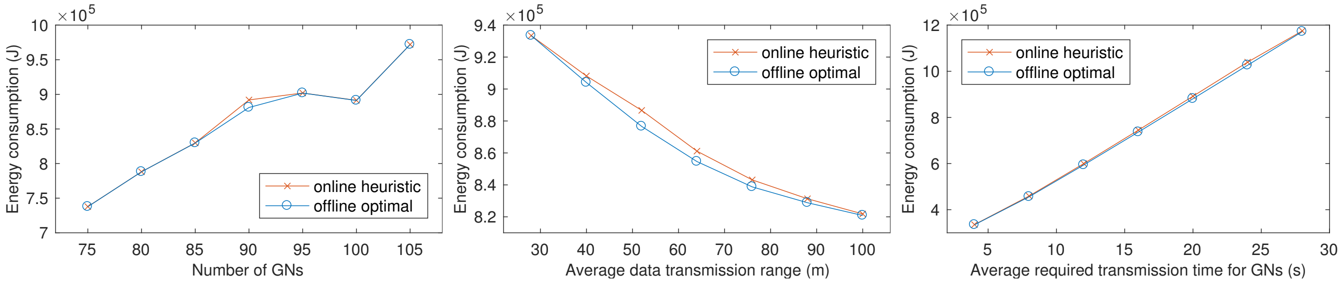
In this subsection, other than the *data transmission range*, we define the *control communication range* [21], which is generally much larger, but with slower transmission rate. In our online algorithm, we assume the information of a GN can be obtained only if the UAV has entered its *control communication range*.

The overview of the proposed online speed scheduling policy proceeds as follows [22]. The GNs keep broadcasting its information such as data amount to be transmitted, data transmission rate, location information, transmission starting position and ending position. Whenever the UAV enters a new GN's control communication range, such information is recorded. In the way, the UAV keeps a list of active GNs, and deletes a GN if all its data has been collected. Once this list is updated, it invokes the offline algorithm to compute a speed schedule.

B. *Simulation settings and results*

In this subsection, we implement the proposed offline and online algorithm respectively. Since there is no other algorithm focusing on the same USS-GTS problem with practical UAV flight energy model, we compare the online algorithm against the optimal offline algorithm.

In simulations, the power-speed function $p(v)$ is set to be $p(v) = 0.07v^3 + 0.0391v^2 - 13.196v + 390.95$, which is obtained from our real world measurements data. It can be easily computed that $v^\# = 7.74m/s$ and $v^* = 13.99m/s$. We consider a straight path with total length $10km$. Along such path, n GNs are assumed to be randomly deployed. For these GNs, the control data transmission range size is set to be $50m$, the average data transmission range size be b and the average time require for data transmission be τ. Parameters n,



(a) (b) (c)

Fig. 7. The performance comparison in UAV energy consumption. The default setting is: number of GNs $n = 90$, average transmission range size $b = 50$ meters, average required transmission time $\tau = 20$ seconds. The three parameters are changed one at a time as in sub-figures.

b and τ will be changed one at a time to evaluate their impacts on algorithm performance. For every parameter setting in our simulation, we randomly generate 100 instances of GNs, and use the mean results for comparisons.

The simulation results are presented in Fig. 7. We can see from Fig. 7(a) that the more GNs deployed, the more energy consumption for the UAV. This is not hard to explain: more GNs means longer total collect data time required, hence more time for the UAV to finish this task. Fig. 7(b) shows that, with the increase of average data transmission range size, the UAV energy consumption drops. This is because: the larger data range, the more overlap between GNs, the more flexible to schedule the transmission, the speed is more ‘equal’. It is shown in Fig. 7(c) that the larger required transmission time for GNs, the more energy consumption. Moreover, the energy consumption increases linearly with the GN required transmission time. This is obvious: more transmission time means more power spent to fly. In all of three subfigures, the online heuristic produce a result close to the offline optimal, generally within 102% of the optimal minimal result.

As a conclusion, the online algorithm performance is near optimal, because our online algorithm is designed based on the offline optimal properties.

VIII. Conclusion

This paper investigated a UAV data collection problem from GNs deployed along a straight line. Unlike existing works, we focused on the flight speed scheduling, *i.e.,* control the speed of the UAV to save energy. Based on our real-world flight tests, we disclosed a speed-related energy model, distinct from most existing works in the literature on wireless communication, which usually assume the distance-related or duration-related energy model. We proposed a novel *looking before crossing* algorithm on the time-distance diagram, to optimally solve the offline problem, where all information on GNs is available before scheduling. For the online problem, we presented a heuristic based on the offline algorithm. Simulations showed this online heuristic was within 102% of the offline optimal minimal value. Our study on the practical flight energy model and speed scheduling have shed light on a new direction on UAV-aided wireless communication.

Acknowledgment

This work was supported by the National Key Research and Development Program of China Grant 2017YFB1003000, National Natural Science Foundation of China Grants 61702097, 61632008, 61602112, 61872079, and 61702096, Aeronautical Science Foundation of China Grant 2017ZC69011, Science Technology and Innovation Committee of Shenzhen Municipality Under project JCYJ20170818095109386, Jiangsu Provincial Key Laboratory of Network and Information Security Grant BM2003201, and 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.

Appendix

A. Proof of Lemma 2

Suppose there are two time intervals with duration lengths τ_x and τ_y. The UAV flight speeds are constant inside each duration, and they are v_x and v_y, respectively. We need to show that using the average speed $\bar{v} = \frac{v_x\tau_x+v_y\tau_y}{\tau_x+\tau_y}$ in both intervals is more energy efficient than using any two different speed, $v_x \neq v_y$.

$$\begin{aligned} p(\frac{v_x\tau_x+v_y\tau_y}{\tau_x+\tau_y}) &= p(\frac{\tau_x}{\tau_x+\tau_y}v_x+\frac{\tau_y}{\tau_x+\tau_y}v_y) \\ &\overset{(a)}{<} \frac{\tau_x}{\tau_x+\tau_y}p(v_x)+\frac{\tau_y}{\tau_x+\tau_y}p(v_y). \end{aligned} \tag{5}$$

The inequation (a) is because the convex property of the function $p(v)$ and the fact that $v_x \neq v_y$. Hence, we have

$$(\tau_x+\tau_y)p(\frac{v_x\tau_x+v_y\tau_y}{\tau_x+\tau_y}) < \tau_x p(v_x)+\tau_y p(v_y),$$

which clearly shows that use a common speed can reduce energy consumption.

B. Proof of Lemma 3

Assume the given flight position interval is (d_x, d_y), hence the flight distance is $L = d_y - d_x$. Let t be the time spent to cover this distance. According to Lemma 2, the UAV consumes the minimum energy only when flying at a constant speed, $v = \frac{L}{t}$. So, the total energy consumption $E_0 = tp(v) = \frac{p(v)}{v}L$.

Define a function $h(v) = \frac{p(v)}{v}$, so $h'(v) = \frac{1}{v}(p'(v) - \frac{p(v)}{v})$. Since $p(v)$ has its minimum value at $v^{\#}$, when $v \in (0, v^{\#})$, $p(v)$ decreases, so $p'(v) < 0$, then $h'(v) < 0$. When $v \in (v^{\#}, +\infty)$, $p(v)$ increases, so $p'(v) > 0$ but the sign of $h'(v)$ depends on function $p(v)$. More specifically, the sign of $h'(v)$ depends on which one is larger, $p'(v)$ or $\frac{p(v)}{v}$. For any given position $(v, p(v))$, $v > v^{\#}$, on the speed-power diagram, $p'(v)$ can be represented by the slope of the tangent line, and $\frac{p(v)}{v}$ can be represented by the slope of the line connecting $(v, p(v))$ to $(0, 0)$. There are two possible cases: the two slopes equal at some point or they never equal. In case they equal, equation $p'(v) - \frac{p(v)}{v} = 0$ must have a solution, let it be $v = v_m$. Hence, $h'(v) < 0$ if $v < v_m$, and $h'(v) > 0$ if $v > v_m$, and $h(v)$ has the minimum value at $v = \min\{v_m, v_{max}\}$, where v_{max} is the maximum flight speed. In case they never equal, $h'(v) < 0$ for $\forall v \in (0, v_{max}]$, and $h(v)$ monotone decreases with minimum value at $v = v_{max}$.

As a conclusion, the UAV consumes the minimum energy when $v = v^*$, where v^* is equal to v_m or v_{max}, depending on $p(v)$.

C. Proof of Lemma 2

We only provide the sketch of the proof. Intuitively, for an interval in which the UAV speed exceeds v^*, we modify it to use v^* instead, which is more energy-efficient. In this way, the UAV flies slower and more time are available to collect data from GNs, so the upload duration requirements of every GN can still be satisfied.

D. Proof of Lemma 4

Suppose, on the contrary, the optimal trajectory changes its direction by decreasing its slope at point (t, d), from v_1 to v_2. Assume straight line between (t_1, d_1) and (t, d) is with slope v_1, while straight line between (t, d) and (t_2, d_2) is with slope v_2. Since $v_1 > v_2$, the trajectory between (t_1, d_1) and (t_2, d_2) can be *straighten* to be a straight line with slope v to save more energy according to Theorem 1. We next show straighten such two straight lines is feasible. Because south walls of all rooms are at $d = 0$, such straightening generally move the trajectory towards south, not crossing the south wall. In other words, after the modification, the UAV spending more time collection data in (d_1, d) and less time in (d, d_2) given $v_1 > v > v_2$. Since all GNs starting positions are $d = 0$, this modification is always feasible. This is a contradiction, since the optimal trajectory is modified to be even more energy efficient.

E. Proof of Lemma 5

We prove by contradiction. Consider part of the optimal trajectory, between point (t_1, d_1) and (t_2, d_2). Assume there is only one changing point (t, d), and it is not a northern doorjamb. We then try to *straighten* this two lines. There are two cases depending on whether connecting (t_1, d_1) and (t_2, d_2) directly is feasible. In case it is feasible, then a contradiction arise since *straightening* saves energy. In case it is not feasible, then there must be at least one northern doorjamb inside the triangle of (t_1, d_1), (t_2, d_2) and (t, d). We therefore choose one as the new slope change point (t', d'). Let $\tau_1, \tau_2, \tau_1', \tau_2'$ be the old and new time spent before and after changing, *i.e.,* $\tau_1 = t - t_1, \tau_2 = t_2 - t, \tau_1' = t' - t_1, \tau_2' = t_2 - t'$. Let v_1, v_2, v_1', v_2' be the old and new speeds before and after changing, *i.e.,* $v_1 = (d - d_1)/\tau_1, v_2 = (d_2 - d)/\tau_2, v_1' = (d' - d_1)/\tau_1', v_2' = (d_2 - d')/\tau_2'$. Then, we must have $v_1 < v_1'$ and $v_2 > v_2'$. Since the distance covered and duration spent do not change, we have the follow equations.

$$v_1\tau_1 + v_2\tau_2 = v_1'\tau_1' + v_2'\tau_2',$$
$$\tau_1 + \tau_2 = \tau_1' + \tau_2'.$$

Combine the two equations by division, we have

$$\frac{\tau_1}{\tau_1 + \tau_2}v_1 + \frac{\tau_2}{\tau_1 + \tau_2}v_2 = \frac{\tau_1'}{\tau_1' + \tau_2'}v_1' + \frac{\tau_2'}{\tau_1' + \tau_2'}v_2'.$$

By the convexity of the $p(v)$ function and $v_1 < v_2$, we have

$$\frac{\tau_1}{\tau_1 + \tau_2}p(v_1) + \frac{\tau_2}{\tau_1 + \tau_2}p(v_2) > \frac{\tau_1'}{\tau_1' + \tau_2'}p(v_1') + \frac{\tau_2'}{\tau_1' + \tau_2'}p(v_2').$$

So,

$$\tau_1 p(v_1) + \tau_2 p(v_2) > \tau_1' p(v_1') + \tau_2' p(v_2').$$

Thus, the optimal energy consumption is further improved, which is a contradiction.

F. Proof of Theorem 3

Since the algorithm repeats to find all trajectory pieces one by one, we prove the produced trajectory is optimal by showing its first piece is optimal. The first trajectory is set at either Line 7 or Line 12. We will prove both of they are optimal.

The farthest northern doorjamb in view k_v^n is computed at line 4, while its corresponding speed is given v_v^n at line 5. (1). We now prove, in case $v_v^n \leq v^*$, connecting (t_k, f_k) and $(t_{k_v^n}, f_{k_v^n})$ is the first optimal trajectory piece, where $k = 0$ in the first iteration. Suppose otherwise, the first piece ends at other point (t, d). According to Lemma 5, point (t, d) must be a northern doorjamb. Let $(t, d) = (t_{k^{opt}}, f_{k^{opt}})$, for some $k^{opt} \neq k_v^n$. It is impossible $k^{opt} > k_v^n$, because northern doorjamb beyond k_v^n is not in view, such trajectory piece will go outside rooms which is infeasible. It is impossible $k^{opt} < k_v^n$ as well, because v_v^n is the smallest by Line 4 and 5, if the first trajectory piece ends at any doorjamb before k_v^n, then there must be a following trajectory piece with a smaller slope, contradicting to Lemma 4. Hence, the first case is proved. (2). In case $v_v^n > v^*$, connecting (t_k, f_k) and (t_n, f_n) is the first optimal trajectory piece, where $k = 0$ in the first iteration. Because $t_n = (f_n - f_k)/v^* + t_k$ at Line 11, we have the slope of this trajectory piece as v^*. Beside, this trajectory piece is feasible, because, v_v^n is the smallest slope and v^* is even smaller. By Lemma 3, using v^* is optimal.

The dominated operation in this algorithm is the **while** loop. Inside the loop, the computation of k_v^n at Line 4 dominates, which takes n steps to calculate and find the minimum value. The **while** loop repeats at most n times, because in each iteration, variable k increases at least 1, and loop terminates after $k \geq n$. Therefore the time complexity of this algorithm is $O(n^2)$.

G. Proof of Theorem 4

Like the proof of Theorem 3, we prove the produced trajectory is optimal by showing its first piece is optimal. The first trajectory piece ends at (1) a northern doorjamb as at Line 8, (2) a southern doorjamb as at Line 11, and (3) directly at the destination as at Line 17.

The case (3) is obviously optimal. We prove (1) is optimal as well, and the case (2) is symmetry and left for the readers. In case (1), because the destination is not in directly view, then, according to Lemma 8, we know the first piece ends at a doorjamb point. It is known that at Line 8, the view angle has its northern boundary bounded by the northern doorjamb of Door k_n^v. At the same time, southern boundary bounded by the southern doorjamb of Door k_s^v. The southern doorjamb of the next door j is outside the view angle, on its north side. The algorithm has chosen the northern doorjamb of Door k_n^v as the ending of the first piece of trajectory. Suppose, on the contrary, another doorjamb is, instead, the optimal ending point of the first piece. Then, such doorjamb must not beyond j, because that will be infeasible (trajectory not within rooms). Such doorjamb must not be a northern doorjamb before or after k_n^v, because otherwise the trajectory can be improved. Similar, such doorjamb must not be a southern doorjamb before or after k_s^v. As a conclusion, the only possibility is the such doorjamb be the southern doorjamb of door k_s^v. We shown the resulting trajectory can be improve, too. To pass through door j, such trajectory must go north, crossing the northern boundary of the view angle. Let such crossing point be (t, d). By Theorem 1, the trajectory between point (t_k, d_k) and (t, d) can improved by *straightening*. This is a contradiction.

The **while** loop repeats one most n times, because each iteration k increases at least 1, and loop terminates after $k \geq n$. Inside the **while** body, the **for** loops no more than n times. Therefore the time complexity of this algorithm is $O(n^2)$.

References

[1] X. Xu, L. Duan, and M. Li, "Strategic learning approach for deploying uav-provided wireless services," *arXiv preprint arXiv:1907.00301*, 2019.

[2] X. Wang and L. Duan, "Dynamic pricing and capacity allocation of uav-provided mobile services," in *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*. IEEE, 2019, pp. 1855–1863.

[3] M. Moradi, K. Sundaresan, E. Chai, S. Rangarajan, and Z. M. Mao, "Skycore: Moving core to the edge for untethered and reliable uav-based lte networks," in *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking*. ACM, 2018, pp. 35–49.

[4] Y. Ma, N. Selby, and F. Adib, "Drone relays for battery-free networks," in *Proceedings of the Conference of the ACM Special Interest Group on Data Communication*. ACM, 2017, pp. 335–347.

[5] N. Cheng, F. Lyu, W. Quan, C. Zhou, H. He, W. Shi, and X. Shen, "Space/aerial-assisted computing offloading for iot applications: A learning-based approach," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 5, pp. 1117–1129, 2019.

[6] J. Gong, T.-H. Chang, C. Shen, and X. Chen, "Flight time minimization of uav for data collection over wireless sensor networks," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 9, pp. 1942–1954, 2018.

[7] 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 *IEEE INFOCOM 2019*. IEEE, 2019, pp. 2404–2412.

[8] 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 Journal on Selected Areas in Communications*, vol. 36, no. 9, pp. 2059–2070, 2018.

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

[10] C. Zhan and H. Lai, "Energy minimization in internet-of-things system based on rotary-wing uav," *IEEE Wireless Communications Letters*, 2019.

[11] Y. Zeng, J. Xu, and R. Zhang, "Energy minimization for wireless communication with rotary-wing uav," *IEEE Transactions on Wireless Communications*, vol. 18, no. 4, pp. 2329–2345, 2019.

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

[13] Z. Zhou, C. Zhang, C. Xu, F. Xiong, Y. Zhang, and T. Umer, "Energy-efficient industrial internet of uavs for power line inspection in smart grid," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 6, pp. 2705–2714, 2018.

[14] A. Mazayev, N. Correia, and G. Schütz, "Data gathering in wireless sensor networks using unmanned aerial vehicles," *International Journal of Wireless Information Networks*, vol. 23, no. 4, pp. 297–309, 2016.

[15] A. Rahmati, X. He, I. Guvenc, and H. Dai, "Dynamic mobility-aware interference avoidance for aerial base stations in cognitive radio networks," in *IEEE INFOCOM 2019*. IEEE, 2019, pp. 595–603.

[16] "Lora transceivers - semtech," www.semtech.com/products/wireless-rf/lora-transceivers, accessed Jul. 30, 2019.

[17] "Esp82766ex overview - espressif systems," www.espressif.com/en/products/hardware/esp8266ex/overview, accessed Jul. 30, 2019.

[18] J. Modares, F. Ghanei, N. Mastronarde, and K. Dantu, "Ub-anc planner: Energy efficient coverage path planning with multiple drones," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 6182–6189.

[19] K. Karydis and V. Kumar, "Energetics in robotic flight at small scales," *Interface focus*, vol. 7, no. 1, p. 20160088, 2017.

[20] M. A. Zafer and E. Modiano, "A calculus approach to energy-efficient data transmission with quality-of-service constraints," *IEEE/ACM Transactions on Networking (TON)*, vol. 17, no. 3, pp. 898–911, 2009.

[21] C. Zhan, Y. Zeng, and R. Zhang, "Energy-efficient data collection in uav enabled wireless sensor network," *IEEE Wireless Communications Letters*, vol. 7, no. 3, pp. 328–331, 2017.

[22] X. Ren, W. Liang, and W. Xu, "Data collection maximization in renewable sensor networks via time-slot scheduling," *IEEE Transactions on Computers*, vol. 64, no. 7, pp. 1870–1883, 2014.