

An Energy Optimization Algorithm for UAV-Assisted Satellite Mobile Edge Computing System

Chih-Hung Lu[†], Jang-Ping Sheu[‡], and Chi-Yu Hsieh[‡]

[†]Institute of Communications Engineering, National Tsing Hua University, Taiwan

[‡]Department of Computer Science, National Tsing Hua University, Taiwan

Emails: s109064532@m109.nthu.edu.tw, sheujp@cs.nthu.edu.tw, and zhiyujudy0422@gapp.nthu.edu.tw

Abstract—In recent years, mobile edge computing (MEC) has become one of the most popular applications in the Internet of Things (IoT). With the help of satellite communications, MEC can be realized in remote areas. However, when transmitting directly to satellites, the energy consumption of IoT devices remains a challenge. This paper studies an unmanned aerial vehicle (UAV) assisted MEC system in which the UAV and satellite are both feasible MEC servers providing computation services. We aim to minimize the total energy consumption among all IoT devices by jointly determining the offloading decision and UAV's trajectory under the constraint of an energy budget. To tackle the problem, we utilize an existing heuristic algorithm for solving the classic Orienteering Problem and propose a dynamic programming algorithm to reduce the hovering cost of the UAV to serve more IoT devices. Simulation results show that the performance of the proposed algorithm is better than the baselines.

Index Terms—Unmanned Aerial Vehicle (UAV), Mobile Edge Computing (MEC), Satellites, Internet of Things (IoT).

I. INTRODUCTION

Over the past few years, there has been a significant increase in the usage and development of IoT devices, resulting in the creation of smart environments that necessitate extensive data processing and analysis [1]. As a result, computing services are becoming increasingly important in the IoT industry. It should be noted that 5G and 6G systems have made great strides recently, which has led to a trend of computing services moving from the cloud to the edge, enabling real-time data processing. However, a traditional cellular network may not be capable of providing ubiquitous coverage and mobile edge computing (MEC) services to all devices, specifically in remote areas, due to the high infrastructure cost.

Fortunately, with the advance of wireless communications, the space-air-ground integrated network (SAGIN) is getting more attention. It can be seen as a system that combines satellites, aircraft, and ground nodes to provide seamless wireless communication coverage. With such properties, the low earth orbit (LEO) satellite is a potential solution for providing wireless communications in remote areas where traditional communication infrastructure is unavailable. LEO satellites orbit at altitudes between 160 to 2,000 kilometers and

can offer lower latency and higher bandwidth communication than geostationary satellites because they are closer to the Earth [2]. Additionally, IoT devices are usually limited in power and may not have a convenient way to recharge or replace batteries; hence, using UAVs in a remote area can also be an effective way to improve the performance of the MEC system due to its proximity to the IoT devices.

UAVs can provide better flexibility and mobility than traditional MEC servers and provide on-demand and targeted services to IoT devices in various locations, improving the performance and energy efficiency of the system [3]. The works in [4] [5] demonstrate how to utilize UAV to address the latency-aware problem in the UAV-aided MEC network, which aims to minimize the latency or total delay of all the users.

A hybrid satellite-terrestrial network utilizing satellites as MEC servers is an emerging area of research because they provide comprehensive coverage and seamless communications, making them well-suited for IoT applications [6]. A satellite with computing units onboard can serve as a MEC server for the remote area or as a data relay that forwards computing data between IoT devices and ground-based MEC servers [7]. In [6], the authors proposed a computation offloading scheme where ground users can compute their tasks locally, at LEO satellites, or on a backhaul cloud server. However, the transmission energy associated with satellite communications can be high, affecting performance. Hence, using both UAVs and satellites in remote areas could be more efficient than using satellites alone [8].

Most prior research assumes that a UAV hovers in a fixed location or follows a predetermined path without considering the UAV's energy limitations. This paper considers several IoT devices with computation requirements deployed in a remote area where a traditional cellular network is unavailable. A UAV equipped with a MEC server can serve IoT tasks in UAV flight and hovering time, and an LEO satellite can also provide MEC services to IoT devices. We want to minimize the total energy consumption among all the IoT devices while ensuring that UAV or LEO satellites can adequately execute all the computation tasks.

However, serving all IoT devices may not be feasible due to the limited energy budget of the UAV. It is worth noting

This work was supported in part by the National Science and Technology Council, Taiwan, under grant 112-2221-E-007-046-MY3 and Qualcomm Technologies, Inc. under grant SOW NAT-487844.

that choosing the served devices can significantly impact the UAV's trajectory and vice versa. To solve this problem, we first mapped it to the Orienteering Problem (OP) [9], an NP-hard optimization problem, and used a 4-phase heuristic algorithm [10] to determine the data offloading devices and the UAV's trajectory. Secondly, we propose a dynamic programming (DP) algorithm to optimize the task offloading in the UAV's flying and hovering time and reduce the UAV's energy consumption. Consequently, the saved energy on the UAV can be utilized to serve some unvisited IoT devices. Maximizing the number of IoT devices served by the UAV can result in significant energy savings for these devices since the UAV is closer to them than satellites, thereby minimizing the overall energy consumption of the IoT devices. Our proposed algorithm exhibits the lowest total energy consumption compared to other approaches based on the simulation results.

The rest of the paper is organized as follows. Section II presents our UAV-assisted satellite MEC system model and the objective of this paper. Then, we describe our proposed algorithm in Section III. In Section IV, the performance of the proposed method is evaluated by simulations. Finally, we conclude our work in Section V.

II. SYSTEM MODEL AND PROBLEM FORMULATION

A. System Model

The UAV-assisted satellite MEC model consists of a large number of IoT devices with computation requirements, denoted as a set of $\mathcal{I} = \{1, 2, \dots, I\}$, and $\pi_i = (x_i, y_i)$ denotes the location of device $i \in \mathcal{I}$ in the 2D coordinate system. The UAV flies at altitude h over the area following a determined trajectory with a constant speed v to provide MEC service for IoT devices. The IoT devices are either served by the UAV or LEO satellite as a MEC server. The task of the IoT device $i \in \mathcal{I}$ is represented as $\mathcal{T}_i = \{d_i, c_i\}$, where d_i and c_i denote the data size and required CPU cycle per bit of the IoT device, respectively. Let r_i denote the communication radius of the IoT device i . The IoT device can offload its task to the UAV as it enters its communication radius. Let $q_u(t) \triangleq (x_u(t), y_u(t))$ denote the horizontal location of the UAV at time t . The UAV starts from an initial location $\pi_0 = (x_u^0, y_u^0)$, flies across multiple IoT devices, then returns to its initial location (x_u^0, y_u^0) at the end of its mission. The UAV's trajectory is determined by our algorithm for selecting the IoT devices to visit. If there are n IoT devices selected for offloading their tasks to UAV, we relabel the n IoT devices in the trajectory as $\{1, 2, \dots, n\}$ according to their visiting sequence. The unserved IoT devices are relabeled as $\{n+1, n+2, \dots, I\}$. We present our system model's communication and energy consumption in the following.

1) *Communication Model*: We assume that line-of-sight (LoS) links dominate the communication between IoT devices and UAVs. Let the distance between the IoT device and UAV at time t be represented as $d_{i,u}(t)$. We calculate the data rate of IoT devices to UAV and satellite links as follows.

The transmission rate between IoT device i and UAV can be calculated as

$$R_i^u(t) = B_u \cdot \log_2\left(1 + \frac{P_{i,u}\beta_0}{d_{i,u}(t)^2\sigma^2}\right), \quad (1)$$

where B_u denotes the bandwidth for the IoT device to UAV link, $P_{i,u}$, β_0 , and σ^2 represent the transmission power from the IoT device i to UAV, the channel power gain at a reference distance $d = 1\text{m}$, and the noise power at the UAV receiver, respectively [11]. In satellite transmission links, the IoT devices transmit data to the satellite using a line-of-sight (LoS) connection. We can calculate the channel capacity from an IoT device to the satellite by

$$R_s = B_s \cdot \log_2\left(1 + \frac{P_s G_{tx} G_{rx} c^2}{(4\pi d_s f_s)^2 k_B T_s B_s}\right), \quad (2)$$

where B_s is the bandwidth between the IoT device and satellite, P_s represents the transmit power of the device to the satellite, G_{tx} and G_{rx} are the transmitter and receiver antenna gains of the IoT device and satellite, respectively, and c is the speed of light. f_s , d_s , k_B and T_s denote the carrier frequency, transmission distance between device and satellite, Boltzmann's constant, and the system noise temperature, respectively [12]. Note that the device offloading indicator is a critical aspect of the system that determines which IoT devices can be served by the UAV. It can be expressed as a binary variable $o_i^u \in \{0, 1\}$, where a value of 1 means that the UAV serves device i and a value of 0 indicates that the satellite serves the device.

2) *Energy Consumption*: Our model's energy consumption comprises two parts: (i) the energy consumed by the UAV and (ii) the energy consumed by the IoT devices. The energy consumption of the UAV includes flying energy E_f^u , hovering energy E_h^u , and computing energy E_c^{ui} . The IoT device includes the energy needed to transmit computation tasks to the UAV or satellite, i.e., E_i . Based on [13], the overall energy consumption of the UAV can be represented as

$$E_t^u = E_f^u + E_h^u + \sum_{i=1}^n E_c^{u,i}, \quad (3)$$

where E_f^u , E_h^u , and $E_c^{u,i}$ denote UAV's flying, hovering, and computing energy consumption for serving IoT device i , respectively.

For the energy consumption of IoT devices, transmission energy is the only term considered in our model. The IoT devices can offload data to the UAV while flying or hovering. Hence, the offloading time from the IoT device to the UAV comprises the offloading time when the UAV hovers on t_h^i and UAV is flying t_f^i . In addition, the offloading time from the IoT device to the satellite can be calculated as $\frac{d_i}{R_s}$. Therefore, we can calculate the transmission energy consumption of IoT devices by

$$E_i = \begin{cases} (t_f^i + t_h^i)P_{i,u}, & \text{if } o_i^u = 1, \\ \frac{d_i}{R_s}P_s, & \text{if } o_i^u = 0. \end{cases} \quad (4)$$

B. Problem Formulation

The main objective of this article is to minimize the total energy consumption of IoT devices under the constraints of the limited energy budget of the UAV. By jointly optimizing the offloading indicator, the UAV's trajectory, offloading points, and the UAV's hovering time, the optimization problem can be formulated as

$$\begin{aligned} \mathbf{P1} : \quad & \min_{\mathcal{Q}_u, \{o_i^u\}, \{p^{s_i}, p^{e_i}\}, \{t_h^i\}} \sum_{i=1}^I E_i \quad (5) \\ \text{s.t.} \quad & \\ & \text{C1: } o_i^u \in \{0, 1\}, \forall i, \\ & \text{C2: } E_t^u \leq E_b^u, \\ & \text{C3: } \|p^{s_i} - \pi_i\| \leq r_i, \forall i \in \mathcal{I}, o_i^u = 1, \\ & \text{C4: } \|p^{e_i} - \pi_i\| \leq r_i, \forall i \in \mathcal{I}, o_i^u = 1, \end{aligned}$$

where E_b^u is the available energy budget of the UAV, which can be utilized for MEC service. Constraint C1 represents the binary variable of the offloading indicator for each IoT device. Constraint C2 states that the total energy consumption of the UAV cannot be greater than the energy budget E_b^u . Constraints C3 and C4 ensure that the selected offloading start point (p^{s_i}) and endpoint (p^{e_i}) of IoT device i are located within the communication radius of the corresponding IoT device.

Our objective problem **P1** can be mapped to the NP-hard Orienteering Problem (OP) by assuming that the MEC service provided by the UAV is only available when it hovers above the IoT devices [9]. Hence, we have mapped our objective problem to the OP solely to determine the UAV's trajectory and the devices for data offloading. First, we defined the profit in the OP as the energy cost of an IoT device that can be saved by utilizing the UAV as an MEC server and is represented as

$$E_s^i = \frac{d_i}{R_s} P_s - t_h^i P_{i,u}, \quad (6)$$

where t_h^i represents the time required for an IoT device to complete its offloading only when the UAV hovers above it. The optimization problem of OP is to determine the offloading decision for all the IoT devices and find an efficient route to maximize the total saved energy cost E_s^i .

III. FLYING AND HOVERING PLANNING DP-BASED ALGORITHM (FHPDP)

Here, we present a dynamic programming (DP) scheme to optimize the UAV trajectory's offloading cost, called the Flying and Hovering Planning DP-based Algorithm (FHPDP). First, we use a 4-phase heuristic algorithm proposed in [10] to solve the OP. The time complexity of this algorithm is $O(n^3)$. This algorithm determines the offloading indicator o_i^u for each IoT device i and the trajectory of the UAV in polynomial time. After the 4-phase heuristic algorithm, we will obtain a trajectory \mathcal{Q}_u that starts from $\pi_0 = (x_u^0, y_u^0)$ and visits n IoT devices that satisfy the energy constraint before returning to π_0 at the end. Moreover, the offloading indicator of the n visited IoT devices will be set to $o_i^u = 1$, which means that the UAV

will provide MEC service for these devices. Then, we propose a dynamic programming (DP) algorithm that further optimizes the offloading time and reduces the overall energy cost of the UAV. Finally, the saved energy budget can be utilized to serve some unvisited IoT devices and minimize the overall problem of energy consumption.

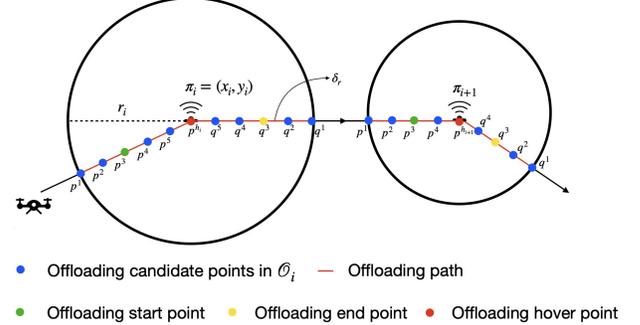


Fig. 1: Illustration of offloading points for IoT device without overlap

The task offloading in \mathcal{Q}_u is based on the UAV hovering above IoT devices. However, if the task size is large, the hovering duration of the UAV will be long and consume a lot of energy. To reduce the hovering time, we can offload the tasks if the UAV is within the communication range of the IoT devices. Here, we first consider the communication range of the IoT device without overlapping with other devices. To determine the optimal offloading start and end points for device i , we give several candidate points with distance δ_r as shown in the red line segments of Fig. 1. Let the candidate offloading points on the entry segment be $\{p^1, \dots, p^{n_i}\}$, and the candidate offloading points on the exit segment be $\{q^1, \dots, q^{n_i}\}$, where $n_i = \frac{r_i}{\delta_r}$. Note that p^1 and q^1 are the UAV entry and exit points for the communication range of the IoT device, respectively. For example, in Fig. 1, the IoT device i can offload its task to the UAV from the green point p^3 , hovering at p^{h_i} , and complete the offloading at the yellow point q^3 finally.

Assuming offloading starts at p^{a_i} and ends at q^{b_i} , the data offloaded at the flying period between p^{a_i} and q^{b_i} is calculated as

$$d_f^{a_i, b_i} = \int_{p^{a_i}(t)}^{p^{a_i}(t) + p^{a_i, h_i}(t)} R_i^u(t) dt + \int_{p^{h_i}(t)}^{p^{h_i}(t) + q^{h_i, b_i}(t)} R_i^u(t) dt, \quad (7)$$

where $p^{a_i}(t)$ denotes the time of UAV arriving p^{a_i} , and $p^{a_i, h_i}(t)$ denotes the duration for UAV flying from p^{a_i} to p^{h_i} , $p^{h_i}(t)$ denotes the time when the UAV reaches the hover point p^{h_i} , and $q^{h_i, b_i}(t)$ denotes the duration for the UAV flying from p^{h_i} to q^{b_i} . Note that the UAV needs to hover above the device i at p^{h_i} for some time if the offloading data size is greater than the data transmitted during the flying period. Thus, if $d_f^{a_i, b_i} < d_i$, the UAV needs to hover at p^{h_i} to satisfy the data offloading constraint. Then we have

$$R_i^u(p^{h_i}(t)) t_h^{a_i, b_i} = d_i - d_f^{a_i, b_i}, \quad (8)$$

where $R_i^u(p^{h_i}(t)) t_h^{a_i, b_i}$ is the task offloaded while the UAV hover at p^{h_i} for a time duration $t_h^{a_i, b_i}$. Otherwise, the hover

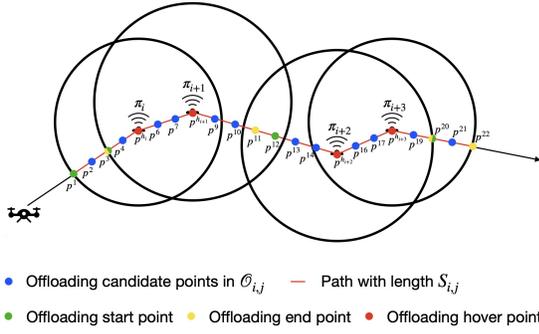


Fig. 2: Illustration of offloading candidate points for IoT device with overlap time for offloading pair (p^{a_i}, q^{b_i}) is 0. Finally, we select the offloading points $(p^{a_i^*}, q^{b_i^*})$ with the least hover time as optimal offloading start and end points for the IoT device i . Since the data rate at point p^j equals to q^j , for $1 \leq j \leq n_i$, we can determine the optimal offloading points (p^{a_i}, q^{b_i}) by calculating $d_f^{a_i, b_i}$ in (7) for $a_i = b_i \in \{1, \dots, n_i\}$.

Next, we consider the overlapping communication ranges of IoT devices, as shown in Fig. 2. We propose a DP scheme to solve this problem. Assume there are $j - i + 1$ consecutive overlapping IoT devices, where i and j denote the first and last indices of IoT devices in an overlapping group. Let $S_{i,j} = r_i + r_j + \sum_{k=i}^{j-1} \|\pi_k - \pi_{k+1}\|$ be the path length from the entry point of device i to the exit point of device j as the red line segments shown in Fig. 2, where $j = i + 3$ in this example. Let $\mathcal{O}_{i,j} = \{p^1, p^2, \dots, p^{n_{i,j}}\}$ be the set of candidate points on the path of length $S_{i,j}$ for overlapping IoT device $\{i, \dots, j\}$, where $n_{i,j} = S_{i,j}/\delta_r$ is the number of candidate points. We must determine which candidate points in $\mathcal{O}_{i,j}$ are within the communication range for each overlap IoT device. Let \mathcal{O}_k be a subset of $\mathcal{O}_{i,j}$ which consists of candidate points that are within the communication range of IoT device $k \in \{i, \dots, j\}$. Let p^{h_k} denote the hover point in \mathcal{O}_k . \mathcal{O}_k can be constructed as $\{p^{c_k}, p^{c_k+1}, \dots, p^{h_k}, \dots, p^{d_k-1}, p^{d_k}\}$, where c_k and d_k are the first and last indices within the communication range of IoT device k , respectively. For example, in Fig. 2, $\mathcal{O}_1 = \{p^1, \dots, p^5, \dots, p^9\}$, where $c_1 = 1, h_1 = 5, d_1 = 9$.

Furthermore, a pair of offloading start and end points for an overlapping IoT device i is defined as (p^{a_i}, p^{b_i}) , $\forall a_i, b_i \in \{c_i, \dots, h_i, \dots, d_i\}$, where p^{a_i} and p^{b_i} are the offloading start and end points, respectively. Note that the offloading start point p^{a_i} can be after to the hover point p^{h_i} , and the offloading end point p^{b_i} can be before the hover point p^{h_i} . Hence, the offloading process of (p^{a_i}, p^{b_i}) can be classified into three cases. Thus, the offloaded data at flying period of (p^{a_i}, p^{b_i}) for the first case is the hover point p^{h_i} located between p^{a_i} and p^{b_i} , which is represented as $d_f^{a_i, b_i}$ and can be calculated by equation (7). The second and third cases are p^{a_i} and p^{b_i} before and after p^{h_i} , respectively. Both cases can be calculated by

$$d_f^{a_i, b_i} = \int_{p^{a_i}(t)}^{p^{a_i}(t)+p^{b_i}(t)} R_i^u(t) dt, \quad (9)$$

where $p^{a_i, b_i}(t)$ represents the time it takes for UAV to fly from p^{a_i} to p^{b_i} . If $d_f^{a_i, b_i} < d_i$, the hover time $t_h^{a_i, b_i}$ for (p^{a_i}, p^{b_i}) can

be calculated by the equation (8) when substituting $d_f^{a_i, b_i}$ into it. Otherwise, $t_h^{a_i, b_i} = 0$. To determine the optimal offloading points for consecutive overlapping of IoT devices $\{i, \dots, j\}$, we define the problem as

$$\mathbf{P2} : \min_{\{(p^{a_k}, p^{b_k})\}} \sum_{k=i}^j t_h^{a_k, b_k}, \quad (10)$$

s.t.

- C1: $a_k \leq b_k, \forall k \in \{i, \dots, j\}$,
- C2: $\|p^{a_k} - \pi_k\| \leq r_k, \forall k$,
- C3: $\|p^{b_k} - \pi_k\| \leq r_k, \forall k$.

A valid pair (p^{a_i}, p^{b_i}) has to satisfy constraints C1, C2, and C3. Constraint C1 represents that the index of the start point must not be after the end point's index. Constraints C2 and C3 state that a valid pair must be within an IoT device's communication range. We solve this problem for each consecutive overlapping group in the UAV's trajectory to minimize the overall hover time of the UAV. The problem **P2** can be solved by solving a sequence of Bellman equations. Let $\text{MinT}_{k+1}(p^{a_{k+1}}, p^{b_{k+1}})$ be the minimum accumulative hover time at $(p^{a_{k+1}}, p^{b_{k+1}})$ for IoT device $k + 1$, which is the summation of the hover time from the IoT device i to the device $k + 1$. The Bellman equation can be formulated as

$$\begin{aligned} \text{MinT}_{k+1}(p^{a_{k+1}}, p^{b_{k+1}}) &= \min_{p^{a_k}, p^{b_k} \in \mathcal{O}_k} [\text{MinT}_k(p^{a_k}, p^{b_k})] + t_h^{a_{k+1}, b_{k+1}}, \\ \forall p^{a_{k+1}}, p^{b_{k+1}} &\in \mathcal{O}_{k+1}, \\ \forall a_k, b_k &\in \{c_k, \dots, d_k\} \text{ and } a_k, b_k \leq a_{k+1}, \end{aligned} \quad (11)$$

where $i \leq k \leq j - 1$. Initially, the $\text{MinT}_i(p^{a_i}, p^{b_i}) = t_h^{a_i, b_i}$ for the first IoT device i in an overlapped group for all $a_i, b_i \in \{c_i, \dots, d_i\}$ and $a_i \leq b_i$. Besides, the offloading end point of (p^{a_k}, p^{b_k}) must be before the start point of $(p^{a_{k+1}}, p^{b_{k+1}})$. DP can solve the Bellman equations, as detailed in Algorithm 1.

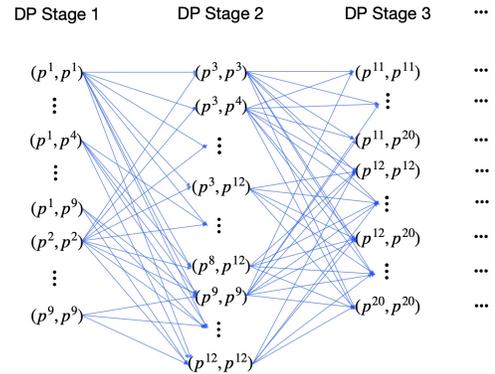


Fig. 3: Example of DP stages for overlap IoT devices in Fig. 2

Fig. 3 shows an example of three stages of the DP algorithm for Fig. 2. We compute the equation (11) for all combinations of candidate points at each stage. The pairs connected with the blue arrow between the two stages mean that the offloading

points of the pair in the previous stage are before the offloading start point of the pair in the next stage. However, some pairs in the previous stage cannot connect to the next stage while solving the equation (11). For example, the pair (p^1, p^4) in stage 1 cannot connect to the pair (p^3, p^3) in stage 2 since the end point of (p^1, p^4) is after the start point of (p^3, p^3) . The DP algorithm will continue until all the $j - i + 1$ stages are computed. We choose the optimal solution for the offloading points in each stage, leading to minimum accumulative hover time.

In Algorithm 1, steps 1 to 4 calculate the hover time for each pair of offloading points in $\mathcal{O}_{i,j}$. In steps 5 to 11, it solves the equation (11) iteratively, the $\text{PrePoint}_{k+1}(p^{a_{k+1}}, p^{b_{k+1}})$ in step 9 denotes which pair of offloading points of the IoT device before $k + 1$ leads to the minimum $\text{MinT}_{k+1}(p^{a_{k+1}}, p^{b_{k+1}})$. Steps 12 to 13 determine the optimal offloading points and hover time of the last overlapped IoT device j . Steps 14 to 16 determine the optimal offloading points and a hover time of IoT devices $j - 1$ to i according to $\text{PrePoint}_{k+1}(p^{a_{k+1}}, p^{b_{k+1}})$ iteratively. The time complexity of Algorithm 1 is $O(n \cdot n_{i,j}^4)$ since there could be at most n IoT devices in an overlapping group for $i = 1$ and $j = n$. Since the number of candidate points in $\mathcal{O}_{i,j}$ is $n_{i,j}$, at most $n_{i,j}^2$ offloading pairs will be computed in the DP for each IoT device. When determining $\text{PrePoint}_{k+1}(p^{a_{k+1}}, p^{b_{k+1}})$ in step 9 of Algorithm 1, it examines a maximum of $n_{i,j}^2$ pairs among all the candidate pairs from the previous stage. Thus, the time complexity can be calculated as $O(n \cdot n_{i,j}^4)$.

Algorithm 1 Flying and Hovering Planning DP-based Algorithm

Input: UAV's trajectory \mathcal{Q}_u , Offloading candidate set $\mathcal{O}_{i,j}$
Output: $\{(p^{a_k}, p^{b_k}), t_h^k\}$, for $i \leq k \leq j$

- 1: **for** $k = i$ to j **do**
- 2: **for** $a_k = c_k$ to d_k **do**
- 3: **for** $b_k = a_k$ to d_k **do**
- 4: Compute $t_h^{a_k, b_k}$ as in (8)
- 5: **for** $k = i$ to $j - 1$ **do**
- 6: **for** $a_{k+1} = c_{k+1}$ to d_{k+1} **do**
- 7: **for** $b_{k+1} = a_{k+1}$ to d_{k+1} **do**
- 8: Compute $\text{MinT}_{k+1}(p^{a_{k+1}}, p^{b_{k+1}})$ as (11)
- 9: Let $\text{PrePoint}_{k+1}(p^{a_{k+1}}, p^{b_{k+1}}) = \text{argmin}_{p^{a_k}, p^{b_k} \in \mathcal{O}_k}$
- 10: $[\text{MinT}_k(p^{a_k}, p^{b_k})] + t_h^{a_{k+1}, b_{k+1}}$,
- 11: $\forall a_k \leq b_k$ and $a_k, b_k \leq a_{k+1}$
- 12: Let $(p^{a_j}, p^{b_j}) = \text{argmin}_{p^{a_j}, p^{b_j} \in \mathcal{O}_j} \text{MinT}_j(p^{a_j}, p^{b_j}), \forall a_j \leq b_j$
- 13: $t_h^j \leftarrow t_h^{a_j, b_j}$
- 14: **for** $k = j - 1$ to i **do**
- 15: $(p^{a_k}, p^{b_k}) \leftarrow \text{PrePoint}_{k+1}(p^{a_{k+1}}, p^{b_{k+1}}), t_h^k \leftarrow t_h^{a_k, b_k}$
- 16: **return** $\{(p^{a_k}, p^{b_k}), t_h^k\}$, for $i \leq k \leq j$

The following is the last phase of our algorithm. Let E_u^s denote the UAV's energy budget saved by offloading proceeding at the flying period rather than hovering above IoT devices. In the last phase, we will utilize E_u^s to serve the unvisited IoT devices not included in \mathcal{Q}_u . Note that the UAV will follow the original trajectory \mathcal{Q}_u to serve the unvisited IoT devices by hovering at their optimal hovering locations

on trajectory \mathcal{Q}_u . The location in the trajectory closest to the selected IoT devices will be determined as the optimal hovering location. Let $p_u^{h_i}$ denote the optimal hovering location for an unvisited IoT device on the trajectory \mathcal{Q}_u . The data rate for an unvisited IoT device i to offload the task to UAV at $p_u^{h_i}$ can be calculated by substituting $d_{i,u}(t)$ in (1) with distance $\|p_u^{h_i} - \pi_i\|$, and the hovering time for completion of offloading is $t_h^i = \frac{d_i}{R_u^i}$. Hence, the hovering energy consumption on UAV to serve the unvisited IoT device $i \in \mathcal{M}$ can be calculated as $E_{u,i}^h = \eta_h \cdot t_h^i$, where η_h is the energy consumption rate of hovering. The unvisited IoT device to be served by UAV will be determined according to a profit-to-cost ratio defined as $\mu_i = E_s^i / E_{u,i}^h$, where E_s^i is defined in (6). Finally, we can select the unvisited candidate IoT device that satisfied $\|p_u^{h_i} - \pi_i\| \leq r_i$ with the highest μ_i value to be served by the UAV greedily. This phase continues until the remaining energy budget E_u^s is depleted.

IV. SIMULATIONS

In this Section, we compare the performance of our algorithm with different methods. Our simulation result is the average of 30 experiments. In our simulations, IoT devices are randomly deployed in a 500m×500m area. Each IoT device has a communication radius of [30, 70] meters, and the data size is in [30, 70] Mb. The UAV is flying with a constant speed of 10m/s at a fixed altitude of 100 m to serve IoT devices with a limited energy budget of 75,000 joules. The transmit power $P_{i,u}$ from the IoT device to the UAV is in the range of [0.1, 0.3] watts, corresponding to the communication radius of the IoT device. The transmit power P_s for IoT devices to satellite is set to one watt. The bandwidth B_u (IoT to UAV) and B_s (IoT to satellite) is 10MHz. The noise power σ^2 is -100dBm, and the channel power gain β_0 is -80dB [14]. Before our simulations, it is necessary to establish the offloading point distance δ_r for the DP algorithm. Based on the results of our experiments, the performance when $\delta_r = 10m$ is comparable to when $\delta_r < 10m$. However, we noticed a rapid increase in execution time as δ_r decreased. Thus, the δ_r is set at 10m.

We compare our algorithm with four methods: Heuristic-OP, TSP with Neighborhoods (TSPN), and TSPN-DP. The Heuristic-OP is the 4-phase heuristic algorithm that solved the OP, which can be used as a baseline for comparison. TSPN improves the energy consumption of the UAV based on the Heuristic-OP. The TSPN is a scheme proposed in [15] using the CVX tool to solve TSP with neighborhoods. The TSPN changes the hovering point of the UAV from above the IoT device to an optimal location within the communication range of IoT devices to reduce the trajectory distance. The TSPN-DP algorithm utilizes our proposed DP scheme on the trajectory obtained from TSPN to reduce the hover time in the offloading process and save energy on the UAV.

Fig. 4 shows the total energy consumption among all IoT devices concerning the different numbers of IoT devices, while the mean data size is set to 50Mb. The proposed FHPDP algorithm outperformed the baseline by 55% and beat TSPN by 48% in this experiment. The TSPN has the second-worst

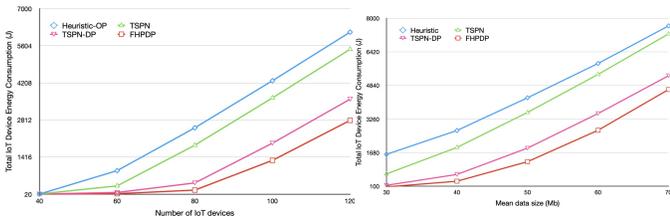


Fig. 4: Total IoT device energy consumption vs. the number of IoT devices

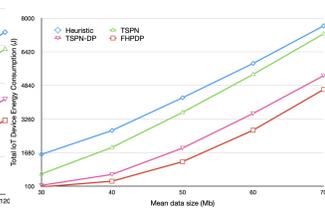


Fig. 5: Total IoT device energy consumption vs. different data sizes

energy consumption since optimizing the flying distance of the trajectory will also increase the hover time due to the increasing distance between the hover point and the IoT device. Hence, the energy budget saved by TSPN can only serve a few additional IoT devices not included in the trajectory. Applying our DP scheme to the TSPN trajectory effectively reduces the UAV's energy consumption and can serve more IoT devices. However, the overall energy consumption of IoT devices is still higher than FHPDP. This is because the shorter the flying distance, the shorter the time for offloading in the flying period. Hence, our proposed FHPDP algorithm outperformed TSPN-DP by 22%. Fig. 5 shows the total energy consumption among all IoT devices for different mean data sizes. The data size of IoT devices varies according to uniform distribution with 5Mb as the standard deviation, and the number of IoT devices is set to 100 in this experiment. Similar to Fig. 4, the proposed FHPDP algorithm outperforms the other methods, and this vantage remains as the data size increases. The energy consumption of FHPDP is 39% lower than baseline while the data size is 70Mb.

Fig. 6 shows UAV serves IoT devices concerning the number of IoT devices. The FHPDP has the highest UAV-served IoT devices, which is 9% more than TSPN-DP. TSPN-DP has a shorter flying distance, and the hover point is not above the IoT device, which increases the hover time. Hence, as the number of IoT devices increases, FHPDP can save more energy than other approaches by reducing hover time. Eventually, the number of UAV-served IoT devices for FHPDP will also be higher than other algorithms. Fig. 7 illustrates the fly and hover time of the UAV for the MEC service concerning different algorithms while the number of IoT devices is 100, where the red part represents the fly time with transmission and the yellow part means the fly time without transmission. The FHPDP has a higher fly transmission time and a lower hover time than the TSPN-DP. This translates to greater energy savings for the UAV during the DP process and eventually serves more IoT devices.

V. CONCLUSION

This paper investigates the energy optimization problem in a UAV-assisted satellite edge computing system. We map the issue to the orienteering problem and use a 4-phase heuristic algorithm to determine the data offloading devices and the UAV's trajectory as an initial solution. Next, we proposed a DP algorithm for overlap and non-overlap IoT devices that minimizes the offloading hover time by determining the

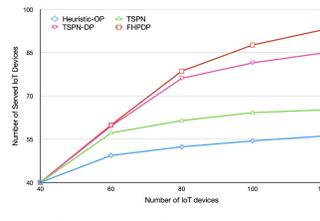


Fig. 6: Number of IoT devices served by UAV vs. the number of IoT devices

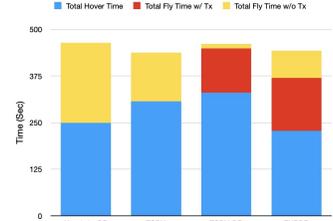


Fig. 7: Fly and Hover time vs. different algorithms

offloading start and end points at the UAV's flying period to save energy of UAV and utilize the saved energy to serve more unvisited IoT devices. The simulation results show that our FHPDP algorithm outperforms the other algorithms on the total energy consumption of IoT devices and the total number of UAV-served IoT devices.

REFERENCES

- [1] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: The communication perspective," *IEEE Communications Surveys and Tutorials*, vol. 19, no. 4, pp. 2322–2358, 2017.
- [2] Z. Zhang, W. Zhang, and F.-H. Tseng, "Satellite mobile edge computing: Improving QoS of high-speed satellite-terrestrial networks using edge computing techniques," *IEEE Network*, vol. 33, no. 1, pp. 70–76, 2019.
- [3] Z. Yang, C. Pan, K. Wang, and M. Shikh-Bahaei, "Energy efficient resource allocation in UAV-enabled mobile edge computing networks," *IEEE Transactions on Wireless Communications*, vol. 18, no. 9, pp. 4576–4589, 2019.
- [4] L. Zhang and N. Ansari, "Latency-aware IoT service provisioning in UAV-aided mobile-edge computing networks," *IEEE Internet of Things Journal*, vol. 7, no. 10, pp. 10573–10580, 2020.
- [5] Y. Wang, H. Guo, and J. Liu, "Cooperative task offloading in UAV swarm-based edge computing," in *2021 IEEE Global Communications Conference*, Madrid, Spain, 2021, pp. 1-6.
- [6] Q. Tang, Z. Fei, B. Li, and Z. Han, "Computation offloading in LEO satellite networks with hybrid cloud and edge computing," *IEEE Internet of Things Journal*, vol. 8, no. 11, pp. 9164–9176, 2021.
- [7] Y. Wang, J. Zhang, X. Zhang, P. Wang, and L. Liu, "A computation offloading strategy in satellite terrestrial networks with double edge computing," in *2018 IEEE International Conference on Communication Systems*, Chengdu, China, 2018, pp. 450-455.
- [8] S. Mao, S. He, and J. Wu, "Joint UAV position optimization and resource scheduling in space-air-ground integrated networks with mixed cloud-edge computing," *IEEE Systems Journal*, vol. 15, no. 3, pp. 3992–4002, 2021.
- [9] R. Kant and A. Mishra, "The orienteering problem: A review of variants and solution approaches," *World Multi-Conference on Systemics, Cybernetics and Informatics*, Florida, USA, 2022, pp. 41-46.
- [10] R. Ramesh and K. M. Brown, "An efficient four-phase heuristic for the generalized orienteering problem," *Computers & Operations Research*, vol. 18, no. 2, pp. 151–165, 1991.
- [11] M. Hua, Y. Wang, Z. Zhang, C. Li, Y. Huang, and L. Yang, "Power-efficient communication in UAV-aided wireless sensor networks," *IEEE Communications Letters*, vol. 22, no. 6, pp. 1264–1267, 2018.
- [12] Z. Jia, M. Sheng, J. Li, D. Niyato, and Z. Han, "LEO-satellite-assisted UAV: Joint trajectory and data collection for internet of remote things in 6G aerial access networks," *IEEE Internet of Things Journal*, vol. 8, no. 12, pp. 9814–9826, 2020.
- [13] 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.
- [14] Q. Wu, Y. Zeng, and R. Zhang, "Joint trajectory and communication design for multi-UAV enabled wireless networks," *IEEE Transactions on Wireless Communications*, vol. 17, no. 3, pp. 2109–2121, 2018.
- [15] J. Zhang, Y. Zeng, and R. Zhang, "UAV-enabled radio access network: Multi-mode communication and trajectory design," *IEEE Transactions on Signal Processing*, vol. 66, no. 20, pp. 5269–5284, 2018.