Joint Task Scheduling, Routing, and Charging for Multi-UAV Based Mobile Edge Computing

Jun Chen
Department of Aerospace Engineering
San Diego State University
San Diego, CA, 92182
Email: jun.chen@sdsu.edu

Junfei Xie
Department of Electrical and Computer Engineering
San Diego State University
San Diego, CA, 92182
Email: jxie4@sdsu.edu

ICC 2022 - IEEE International Conference on Communications | 978-1-5386-8347-7/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICC45855.2022.9839040

***Abstract*—Unmanned aerial vehicles (UAVs) based mobile edge computing (MEC) systems have attracted increasing research attention recently. They can provide on-demand computing services for ground users (GUs) without relying on any communication infrastructures and have the potential to provide better computing services with lower latency, compared with the conventional ground-based MEC or cloud-based systems. Considering the limited battery capacity of the UAVs, existing studies on UAV-based MEC have focused on using UAVs to serve GUs over small areas so that all tasks can be completed during a single flight. In this paper, we aim to remove this restriction and expand the range of users the UAV-based MEC system can serve, by integrating charge stations into the system. A joint task scheduling, routing, and charging problem is then formulated with the objective to minimize the total energy consumption, total service time, and total energy charged simultaneously. To solve this problem, we develop a mixed-integer programming (MIP) model and an equivalent mixed-integer linear programming (MILP) model. Comparative numerical studies demonstrate the optimal solutions found by the proposed approaches.**

I. Introduction

Recently, unmanned aerial vehicles (UAVs) based mobile edge computing (MEC) [1], [2] has attracted much attention. The UAVs, when equipped with computing capabilities, can function as MEC servers to provide computing services at the network edge and allow ground users (GUs) to offload their delay-sensitive and computation-intensive tasks to the UAVs. They can also function as relays to assist users during the task offloading process. Compared with the ground-based MEC systems that rely on terrestrial networks, UAV-based MEC systems have many attractive features. First, due to the low cost and high maneuverability of UAVs, UAV-based MEC systems can be quickly deployed at anytime and anywhere, without restricting to any infrastructures. Considering that the terrestrial network only covers around 20% of the total land area [3], such systems are particularly useful in areas without terrestrial networks and in emergency scenarios where communication infrastructures are destroyed by disasters. Second, UAV-based MEC systems can improve the computation performance as short-distance line-of-sight (LoS) links often exist for ultra-low latency task offloading. Last but not the least, by proactively planning the trajectories for the UAVs, UAV-based MEC systems can provide better computing services to meet the dynamically changing computing needs.

Despite the promising advantages the UAV-based MEC systems have, realizing such systems encounters many daunting challenges due to the unique features of the UAVs, such as the 3-dimensional (3D) mobility, limited battery capacity and payload, and stringent safety requirements. As it is not long after the UAV-based MEC was proposed, studies on such systems are still limited. Most existing studies have focused on a single UAV and investigates how to offload computation tasks from GUs to the UAV [4]–[8]. These studies mainly differ in the objectives and the decision variables.

There are a few studies that consider deploying multiple UAVs to provide computing services for GUs [9]–[11]. In [9] and [10], the optimal placement of UAVs and task assignment was studied, where each UAV hovers above GUs assigned to it and keeps static during service time. [11] considers the joint optimization of UAV trajectories, bit allocation and task assignment to minimize the energy consumption of GUs.

As UAVs are usually powered by batteries with a limited capacity, the service range and service time a UAV can provide in a single flight are limited. Aforementioned studies on UAV-based MEC thus restrict the range of users to be served and plan computation offloading tasks within a short computing cycle. To determine the values of decision variables, the computing cycle is often discretized into a sequence of time slots and decisions are made at each time slot. At the end of the computing cycle, the UAV is forced to reach a predetermined location for charging.

In this paper, we consider the use of multiple UAVs to provide computing services for GUs spatially distributed over a large area, and UAVs can be charged at charge stations or drone depots if running out of power. Therefore, UAVs are not restricted to visit a predetermined location every certain period of time. Instead, they can visit

978-1-5386-8347-7/22/$31.00 ©2022 IEEE

Authorized licensed use limited to: Beijing Information Science & Tech Univ. Downloaded on March 01,2023 at 13:09:59 UTC from IEEE Xplore. Restrictions apply.

nearby charge stations whenever charging is needed. The service range of the proposed UAV-based MEC system can also be easily expanded by deploying more charge stations. To the best of our knowledge, this problem has not been studied in the literature. In this preliminary study, we made the following contributions: 1) we model the above problem as a joint task scheduling, routing, and charging problem, with the objectives of minimizing the total energy consumption, total service time, and total energy charged simultaneously (Sec. II); 2) we develop a mixed-integer programming (MIP) model for the considered problem (Sec. III); 3) we further develop an equivalent mixed-integer linear programming (MILP) model to solve the problem more efficiently (Sec. III); and 4) we conduct comparative numerical studies to evaluate and demonstrate the performance of the proposed approaches (Sec. IV).

II. System Modeling

Consider a UAV-based MEC system that consists of multiple UAVs functioning as MEC servers and aims to provide on-demand computing services to GUs. A control center is responsible for assigning UAVs to fulfill the service requests and planning the route for each UAV. As service requests are expected to arrive continuously, the control center will manage these requests in a rolling horizon manner. Upon receiving a mission, each UAV departs from the depot, visits each service location to fulfill the request, and finally flies back to the depot after mission is completed. As the battery capacity of each UAV is limited, it can get charged at charge stations during the mission. For simplicity, we assume the GUs remain static while being served. Each service request is fulfilled by a single UAV and parallel/distributed computing is not deployed. Moreover, each service request should be fulfilled in a single visit and cannot be interrupted during execution.

A. Network Model

We model the UAV-based MEC system as a directed graph $\mathbf{G}(\mathbb{V}, \mathbb{E})$. Each node $i \in \mathbb{V} = \mathbb{V}^q \cup \mathbb{V}^c \cup \mathbb{V}^d$, located at l_i, represents a service point, a charge station, or a depot, where $\mathbb{V}^q$ and $\mathbb{V}^c$ represent the full set of service points and charge stations, respectively. $\mathbb{V}^d$ includes a single depot. Each edge $(i, j) \in \mathbb{E}$ represents a flight route from node i to node j, which is associated with a distance D_{ij} (m) and an average travel time T_{ij} (s).

Denote the set of service requests to be fulfilled in the current planning horizon as $\mathbb{V}^q$. We model the attributes of each service request $q \in \mathbb{V}^q$ as $\{l_q, A_q, F_q, t_q\}$, where l_q denotes the service location. A_q (bits) is the input data size for processing this request, F_q (cycles/bit) is the number of CPU cycles required for computing one bit of the input data for request q, and t_q is the required service start time.

In terms of the UAVs, the service providers, we model each UAV $k \in \mathbb{K}$ as characterized by attributes $\{v_k, M_k, f_k, \kappa_k, e_k, \overline{B}_k, B_k^0, \boldsymbol{r}_k\}$, where $\mathbb{K}$ is the full set of UAVs. v_k (m/s) is the speed of the UAV, which is assumed to be constant during the mission. M_k (kg) is the mass of the UAV. f_k (GHz) is the CPU-cycle frequency of the carried MEC server, and κ_k is its effective switched capacitance. e_k is the average energy consumption rate (J/s) for hovering. $\overline{B}_k$ (J) denotes the battery capacity, B_k^0 (J) is the initial energy, and $\boldsymbol{r}_k = \{r_{k,i}\}_{i \in \mathbb{V}^c \cup \mathbb{V}^d}$ is a set that specifies the average charging rate (J/s) at each charge station or depot. Typically, there are more charge stations than the depots, due to lower construction cost. Moreover, the level of power each charge station can provide to charge UAVs is often much lower than that provided at the depots. Therefore, although it may be more convenient to get charged at a nearby charge station than at a distant depot, it may take more time to get fully charged.

B. Communication Model

When a UAV k flies to a service location l_q to fulfill request q, the GU who requests the service will start to transmit the data for processing to the UAV at time t_q, and the UAV will hover at l_q during the execution. We neglect the time for sending the computation results back to the GU from the UAV, as the size of the computation results is usually much smaller than that of the input data for many computation tasks.

The communication time is then dominated by the uplink transmission time. For ease of exposition, we assume that the communication link between the UAV and GU is dominated by the LoS channel, and the doppler frequency shift in the communication can be perfectly compensated at the receiver. Therefore, the free-space path loss model [7] can be used to describe the channel quality. In particular, the channel gain is expressed as $g = \beta_0 \mathrm{d}^{-2}$, where β_0 is the channel power at the reference distance $\mathrm{d}_0 = 1$ (m) and d is the distance between the UAV and GU. Assume all UAVs fly at a same constant altitude H, we then have $g = \beta_0 H^{-2}$. Denote the channel bandwidth of the uplink as B, and the transmission power of the GUs as P, both of which are assumed to be constant and same for all uplinks and GUs. The transmission rate (bits/s) can then be described by [12] $R = B \log_2\left(1 + \frac{Pg}{\sigma^2}\right) = B \log_2\left(1 + \frac{P\beta_0}{\sigma^2 H^2}\right)$, where σ^2 is the power spectral density of the channel noise. Given A_q, the amount of data to be transmitted for service request q, the uplink transmission time can be computed by

$$t_{k,q}^{comm} = \frac{A_q}{R} \tag{1}$$

C. Computing Model

The computation time of a service request q depends on the computing capability of the MEC server mounted

Authorized licensed use limited to: Beijing Information Science & Tech Univ. Downloaded on March 01,2023 at 13:09:59 UTC from IEEE Xplore. Restrictions apply.

on the UAV k responsible for this request, as well as the computation load. In particular, it can be computed by

$$t^{comp}_{k,q} = \frac{F_q A_q}{f_k} \tag{2}$$

D. Energy Consumption Model

The UAV consumes battery energy mainly when flying from one location to another, when hovering at the service point for providing the service, and when computing the tasks. We neglect the energy consumed for departing and landing. At charge stations or depots, the UAVs are assumed to not consume any energy.

Assume each UAV k flies at a constant speed v_k. The propulsion energy consumption for UAV k to fly from node i to node j can be computed by [13]:

$$E^{fly}_{k,ij} = 0.5 M_k v_k^2 T_{ij} \tag{3}$$

The propulsion energy consumption for UAV k to hover at location l_q for service request q is

$$E^{hover}_{k,q} \quad = \quad t^{hover}_{k,q} e_k \tag{4}$$

where $t^{hover}_{k,q}$ is the time UAV k hovers at the service location for request q, which should satisfy $t^{hover}_{k,q} \geq t^{comm}_{k,q} + t^{comp}_{k,q}$.

The computation energy consumption for UAV k to process service q can be modeled by [13]:

$$E^{comp}_{k,q} \quad = \quad \kappa_k F_q A_q (f_k)^2 \tag{5}$$

E. Battery Charging Model

The UAV can get charged either at a charge station or at a depot. The charging amount at node i, $i \in \mathbb{V}^c \cup \mathbb{V}^d$, can be computed by the following equation:

$$E^{charged}_{k,i} = r_{k,i} t^{charged}_{k,i} \tag{6}$$

III. MIP and MILP Models

To solve the problem described in the previous section, which is a joint task scheduling, routing and charging optimization problem, we first build a mixed integer programming (MIP) formulation based on the classical network flow model [14]. We then transform it into a mixed integer linear programming (MILP) problem that can be solved with higher efficiency.

To describe the UAV-based MEC system as a network flow model [14], which has a single source and a single sink, we introduce a virtual origin v_0 and a virtual destination v_n to denote the initial and end location, respectively, of all UAVs. We then have $\mathbb{V}^d = \{v_0, v_n\}$, where both v_0 and v_n are located at the base depot.

A. Decision Variables

To capture request assignment and vehicle routing, we introduce a binary variable, $y_{k,ij}$, which equals 1 if UAV k traverses edge (i, j) and 0 otherwise. For formulation simplicity, we assume that each node can only be visited at most once. If multiple visits to a node is needed, we create duplicated nodes for the same location.

To capture the time constraints, we introduce a continuous decision variable $d_{k,i}$ to denote the hover time for UAV k at node i when $i \in \mathbb{V}^q$, and the charging time when $i \in \mathbb{V}^*$, where $\mathbb{V}^* = \mathbb{V}^c \cup \{v_0\}$. In case when $i \in \mathbb{V}^q$, the UAV hovers above the service point to provide computing services, and thus $d_{k,q} = t^{hover}_{k,q}$. Moreover, we have $d_{k,i} = t^{charged}_{k,i}$, when $i \in \mathbb{V}^*$.

Finally, we introduce a decision variable, $b_{k,i}$, to capture the battery level or State of Charge (SoC) of UAV k when arriving at node i. Note that $b_{k,v_0} = \frac{B^0_k}{\overline{B}_k}$.

B. MIP Model

We consider a multi-objective function defined as follows that takes energy consumption, quality of service, and energy charged into account.

$$J = \sum_{k \in \mathbb{K}} \Bigg\{ \alpha_1 \Big(\sum_{\substack{i \in \mathbb{V}, \\ j \in \mathbb{V}^q}} E^{demand}_{k,j} y_{k,ij} + \sum_{(i,j) \in \mathbb{E}} E^{fly}_{k,ij} y_{k,ij} \Big) + \alpha_2 \Big(\sum_{i \in \mathbb{V}, j \in \mathbb{V}^q} (t^{comm}_{k,j} + t^{comp}_{k,j}) y_{k,ij} \Big) + \alpha_3 \Big(\sum_{i \in \mathbb{V}, j \in \mathbb{V}^*} E^{charged}_{k,j} y_{k,ij} \Big) \Bigg\} \tag{7}$$

The first objective describes the total energy consumption, where $E^{demand}_{k,j} = E^{hover}_{k,j} + E^{comp}_{k,j} = d_{k,j} e_k + \kappa_k F_q A_q (f_k)^2$ (according to (4) and (5)) is the energy consumed at service point $i \in \mathbb{V}^q$. The second and third objectives capture the total service time and the total energy charged, respectively, where $E^{charged}_{k,j} = r_{k,i} d_{k,j}$ according to (6). α_1, α_2, and α_3 are the associated weights, whose values should be determined based on the application needs.

Based on the minimum cost network flow problem, we can then formulate the joint task scheduling, routing and charging problem as follows:

minimize: J

subject to:

$$\sum_{k \in \mathbb{K}} \sum_{i \in \mathbb{V}} y_{k,ij} = 1, \quad \forall j \in \mathbb{V}^q \tag{8}$$

$$\sum_{i \in \mathbb{V}} y_{k,ij} \leq 1, \quad \forall j \in \mathbb{V}^c, k \in \mathbb{K} \tag{9}$$

Authorized licensed use limited to: Beijing Information Science & Tech Univ. Downloaded on March 01,2023 at 13:09:59 UTC from IEEE Xplore. Restrictions apply.

$$\sum_{j\in\mathbb{V}} y_{k,v_0 j} = 1, \quad \forall k\in\mathbb{K} \tag{10}$$

$$\sum_{i\in\mathbb{V}} y_{k,iv_n} = 1, \quad \forall k\in\mathbb{K} \tag{11}$$

$$\sum_{i\in\mathbb{V}} y_{k,ih} - \sum_{j\in\mathbb{V}} y_{k,hj} = 0, \quad \forall h\in\mathbb{V}\backslash\mathbb{V}^d,\ k\in\mathbb{K} \tag{12}$$

$$t_j \geq (t_i + T_{ij} + d_{k,i}) y_{k,ij}, \quad \forall i\in\mathbb{V}\backslash v_n, j\in\mathbb{V}\backslash v_0, k\in\mathbb{K} \tag{13}$$

$$d_{i,k} \geq t_{k,i}^{comm} + t_{k,i}^{comp}, \quad \forall k\in\mathbb{K}, i\in\mathbb{V}^q \tag{14}$$

$$b_{k,j}\overline{B}_k = \sum_{i\in\mathbb{V}^q}\left(b_{k,i}\overline{B}_k - E_{k,ij}^{fly} - E_{k,i}^{demand}\right) y_{k,ij} + \sum_{i\in\mathbb{V}^*}\left(b_{k,i}\overline{B}_k - E_{k,ij}^{fly} + E_{k,i}^{charged}\right) y_{k,ij}, \quad \forall k\in\mathbb{K}, j\in\mathbb{V}\backslash v_0 \tag{15}$$

$$0 \leq b_{k,i}\overline{B}_k + E_{k,i}^{charged} \leq \overline{B}_k, \quad \forall k\in\mathbb{K}, i\in\mathbb{V}^* \tag{16}$$

$$\left(\sum_{i\in\mathbb{V}} y_{k,ij}\right)\underline{b}_k \leq b_{k,j} \leq 1, \quad \forall k\in\mathbb{K}, j\in\mathbb{V} \tag{17}$$

Among the above constraints, (8) ensures that each service request is fulfilled by a UAV. (9) restricts the number of visits by a UAV to each charge station to be at most one. Constraints (10) and (11) make each UAV to initially depart from the virtual depot v_0 and finally return to the virtual depot v_n, respectively. The flow balance is achieved by enforcing constraints (12).

Constraints (13) eliminate sub-tours and ensure the UAVs arrive at each service location no later than the required service start time. It is clear that for a service request $i\in\mathbb{V}^q$, t_i is the required service start time for this request. Nevertheless, for a charge station $i\in\mathbb{V}^c$, t_i here represents the required charging start time at this station. To allow UAVs to get charged at a charge station at different times, we duplicate each charge station for multiple times and assign a different charging start time to each duplicated node. In the special cases when $i = v_0$ and $i = v_n$, we set t_i to 0 and $\mathcal{T}$, respectively, where $\mathcal{T}$ is the length of the planning time horizon. Moreover, as the service time should not be shorter than the time required to complete a request, constraints (14) are introduced. Note that, UAVs may arrive at a node i before the service/charging start time t_i. To save energy, we assume the UAVs will wait on the ground before t_i.

Constraints (15) ensure that the amount of energy charged at a charge station or depot and the amount of battery energy consumed is balanced. Furthermore, constraints (16) guarantee the battery of each UAV is not overcharged. Lastly, to ensure there is enough reserved energy for landing in case of emergencies, constraints (17) are introduced, where $\underline{b}_k$ is the minimum battery level required for UAV k to land safely, which is often set to a value between 0.2 and 0.3 [15].

C. MILP Model

The MIP problem formulated in the previous subsection is not easy to solve, due to the nonlinear constraints (13) and (15). To simplify the problem, we apply the "big-M" formulation [16], [17] to transform the MIP into an MILP model. In particular, by defining a sufficiently large constant M, we can transform the nonlinear constraints (13) into an equivalent linear form:

$$t_j \geq t_i + T_{ij} + d_{k,i} - M(1 - y_{k,ij}), \quad \forall i\in\mathbb{V}\backslash v_n, j\in\mathbb{V}\backslash v_0, k\in\mathbb{K} \tag{18}$$

The nonlinear constraints (15) have two different linear cases since node i can be either a service point or a charge node (including the initial depot v_0). We thus consider the two cases separately and transform constraints (15) into the following linear forms when node i is a service point, i.e., $i\in\mathbb{V}^q$:

$$\begin{aligned} b_{k,j}\overline{B}_k &\leq b_{k,i}\overline{B}_k - E_{k,ij}^{fly} - E_{k,i}^{demand} + M(1 - y_{k,ij}),\\ b_{k,j}\overline{B}_k &\geq b_{k,i}\overline{B}_k - E_{k,ij}^{fly} - E_{k,i}^{demand} - M(1 - y_{k,ij}),\\ &\forall k\in\mathbb{K}, i\in\mathbb{V}^q, j\in\mathbb{V}\backslash v_0 \end{aligned} \tag{19}$$

and when node i is a charge node, i.e., $i\in\mathbb{V}^*$:

$$\begin{aligned} b_{k,j}\overline{B}_k &\leq b_{k,i}\overline{B}_k - E_{k,ij}^{fly} + d_{k,i}r_{k,i} + M(1 - y_{k,ij}),\\ b_{k,j}\overline{B}_k &\geq b_{k,i}\overline{B}_k - E_{k,ij}^{fly} + d_{k,i}r_{k,i} - M(1 - y_{k,ij}),\\ &\forall k\in\mathbb{K}, i\in\mathbb{V}^*, j\in\mathbb{V}\backslash v_0 \end{aligned} \tag{20}$$

After these linear transformations, all constraints are in a linear form. The resulting problem then becomes a standard MILP problem.

IV. Numerical Studies

In this section, we conduct numerical studies to evaluate the performance of proposed MIP and MILP models.

A. Experiment Setup

We consider a $4\text{km}\times 4\text{km}$ area with the drone depot, i.e., v_0 and v_n, located at the center of the map. Four charge stations are evenly distributed on the map. Each charge station has 6 virtual charge nodes with different and randomly generated charging start times (t_i). The planning time horizon $\mathcal{T}$ is set as 2 hours, which is long enough for a single UAV to fulfill 5-6 requests sequentially. To generate the service requests, we let the input data size A_q for processing each request q be randomly selected from the range $[10, 200]$MB, and let the service start time t_q be randomly selected from the range $[380, 3000]$s. The number of CPU cycles required for computing one bit of the input data, F_q, is set to 10^3 cycles/bit for all requests. The locations of the service requests are randomly generated inside the map.

3 heterogeneous UAVs with different computing capabilities and initial battery energies are considered. In particular, the CPU-cycle frequency f_k of each UAV

Authorized licensed use limited to: Beijing Information Science & Tech Univ. Downloaded on March 01,2023 at 13:09:59 UTC from IEEE Xplore. Restrictions apply.

TABLE I: Comparison of naive and optimal plans

Method	ID	Route	Energy Consumption(kJ)	Energy Charged(kJ)	Service time(s)	Delay(s)	Execution time(s)
MIP & MILP	1	0-15-3(c)-7-19	450	72	1155	0	34.893
	2	0-14-13-1(c)-17-8-10-6(c)-18-19	661	279	1725	0	
	3	0-11-9-12-4(c)-16-19	519	119	966	0	
Naive	1	0-15-9-2(c)-8-16-19	659	354	931	0	0.005
	2	0-14-11-17-12-1(c)-18(d)-19	541	385	1302	331	
	3	0-13-7-2(c)-10(d)-19	610	410	1790	102	

k is randomly selected from the range $[1.4, 2.6]$GHz. The initial battery energy, B_k^0, is randomly set between 90% to 100% of the battery capacity $\overline{B}_k$. All other attributes are assigned with the same values for all UAVs for simplicity. In particular, the weight, speed, effective switched capacitance, average energy consumption rate for hovering, battery capacity, the charging rate of each UAV and the minimum battery level are set to $M_k = 4$kg, $v_k = 15$m/s, $\kappa_k = 10^{-28}$, $e_k = 220$J/s, $\overline{B}_k = 500$kJ, $r_{k,i} = 300$J/s $(\forall i \in \mathbb{V}^c \cup \mathbb{V}^d)$, and $\underline{b}_k = 20\%$, respectively. The transmission rate of the UAV-to-ground links is set to $R = 12.46$MB/s.

All simulations are conducted on a computer with Intel i7-8850H CPU and 16GB RAM. The code is written in Python with Gurobi as the solver. The weights of the three objectives, α_1, α_2 and α_3, are set as 1.

B. Naive Method

A rule-based naive method is also implemented to fulfill the service requests. It simulates how a human dispatcher assigns a group of tasks to a UAV fleet. In particular, all requests are sorted by their required service start time (t_i) and processed sequentially based on the first-come first-served rule. Each request is assigned to the UAV with the earliest arrival time, $t_k^{arrival} = t_k^{available} + t_k^{fly}$, where $t_k^{available}$ is the next available time when UAV k is not charging or serving any users. When a request is assigned to UAV k, a charge decision will be made based on its SoC. If its SoC after arriving 1) will be below the required minimum battery level $\underline{b}_k$ OR 2) will not be large enough for the UAV to travel to the nearest charge station after arriving (if the next node is not the end depot), the UAV will move to its nearest charge station to get fully charged first. Then it will continue its mission and update its $t_k^{available}$. Note that the required service start time (t_i) is not enforced as a hard requirement in this method, so UAV k may arrive late for some requests, as we will show in the following experiment results.

C. Experiment Results

Fig. 1 visualizes the optimal task scheduling, routing, and charging plan found by MIP or MILP models when there are 12 service requests. The plan found by the naive method is visualized in Fig. 2. Comparing the two plans, the optimal plan assigns more requests to UAV 2 by jointly considering the routing and charging constraints. The detailed comparison results are summarized in Table I, where (c) in the third column indicates charging occurs and (d) indicates delay happens, i.e., the UAV arrives at the service node after the required service start time. As we can see, the optimal plan is better on all three objectives. Moreover, the optimal plan fulfills the requirements of all service requests without delays, which is not guaranteed by the naive method.

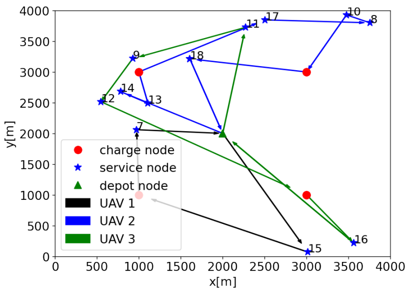


Fig. 1: Visualization of the optimal plan generated by MIP and MILP.

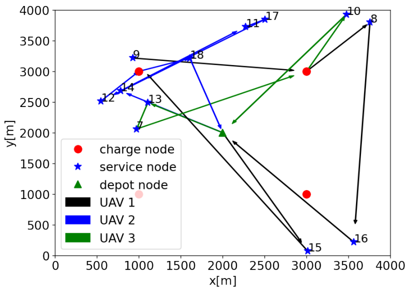
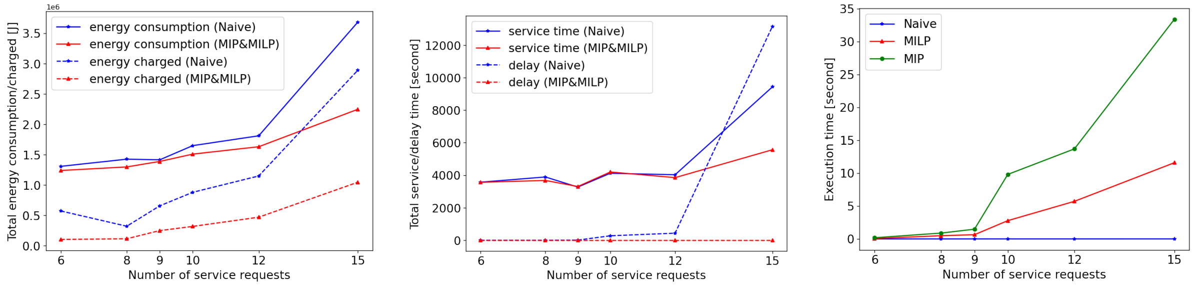


Fig. 2: Visualization of the plan generated by the naive method.

To better understand the performance of the three methods, we further vary the number of service requests. Fig. 3(a) shows the total energy consumption and the total energy charged associated with the plans generated by different methods. As expected, the total energy consumption increases with more service requests, which causes the corresponding energy charged to also increase to fulfill the demand. Also note that the energy gap between consumption and charging is bounded by the initial energy stored in the UAVs. Moreover, since the optimal methods (MIP and MILP) optimize the routing of UAVs and task scheduling, a lower energy consumption is achieved, which in turn leads to a lower energy charged and the saved energy becomes significant with more requests.

Fig. 3(b) shows the total service time and the total delay of the plans generated by different methods. Of interest, the difference of the two plans on the total service time is not big when the number of service

Authorized licensed use limited to: Beijing Information Science & Tech Univ. Downloaded on March 01,2023 at 13:09:59 UTC from IEEE Xplore. Restrictions apply.



(a)

(b)

(c)

Fig. 3: Performance of different methods with varying numbers of service requests in terms of a) the total energy consumption and total energy charged, b) the total service time and total delay, and the c) execution time.

requests is small. This indicates that the naive method performs well in task scheduling when the problem scale is small. However, if considering the total delay, the plan generated by the naive method cannot guarantee each service request to be served on time, and the incurred delays can be significant with more requests.

Fig. 3(c) shows the execution times of different methods. It can be observed that the naive method achieves the highest efficiency and the MIP is the most inefficient. The execution times of both MIP and MILP increase exponentially with more requests. On average, the MILP is about 3-5 times faster than MIP, which is important for large-scale problems.

V. Conclusion

This paper studies a joint task scheduling, routing and charging problem for multi-UAV based MEC systems, with the objectives of minimizing the total energy consumption, total energy charged and total service time simultaneously. To solve this problem, a MIP model was first developed. This model was then transformed into a MILP model with higher efficiency. To evaluate the performance of the proposed methods, a naive method was also implemented for comparison purpose. The experiment results demonstrate that the optimal plans generated by MIP and MILP models are better than the ones generated by the naive method on all the three objectives, and the naive method cannot guarantee all service requests to be fulfilled on time. Moreover, the MILP is about 3-5 times more efficient than the MIP. In the future, we will explore heuristic methods to solve large-scale problems efficiently.

Acknowledgment

We would like to thank the National Science Foundation (NSF) under Grants CI-1953048 and CAREER-2048266 for the support of this work.

References

[1] P. Mach and Z. Becvar, “Mobile edge computing: A survey on architecture and computation offloading,” *IEEE Communications Surveys & Tutorials*, vol. 19, no. 3, pp. 1628–1656, 2017.

[2] N. Cheng, W. Xu, W. Shi, Y. Zhou, N. Lu, H. Zhou, and X. Shen, “Air-ground integrated mobile edge networks: Architecture, challenges, and opportunities,” *IEEE Communications Magazine*, vol. 56, no. 8, pp. 26–32, 2018.

[3] J. Wei, J. Han, and S. Cao, “Satellite iot edge intelligent computing: A research on architecture,” *Electronics*, vol. 8, no. 11, p. 1247, 2019.

[4] S. Jeong, O. Simeone, and J. Kang, “Mobile edge computing via a uav-mounted cloudlet: Optimization of bit allocation and path planning,” *IEEE Transactions on Vehicular Technology*, vol. 67, no. 3, pp. 2049–2063, 2017.

[5] L. Lyu, F. Zeng, Z. Xiao, C. Zhang, H. Jiang, and V. Havyarimana, “Computation bits maximization in uav-enabled mobile edge computing system,” *IEEE Internet of Things Journal*, 2021.

[6] Q. Hu, Y. Cai, G. Yu, Z. Qin, M. Zhao, and G. Y. Li, “Joint offloading and trajectory design for uav-enabled mobile edge computing systems,” *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 1879–1892, 2018.

[7] M. Li, N. Cheng, J. Gao, Y. Wang, L. Zhao, and X. Shen, “Energy-efficient uav-assisted mobile edge computing: Resource allocation and trajectory optimization,” *IEEE Transactions on Vehicular Technology*, vol. 69, no. 3, pp. 3424–3438, 2020.

[8] F. Zhou, Y. Wu, R. Q. Hu, and Y. Qian, “Computation rate maximization in uav-enabled wireless-powered mobile-edge computing systems,” *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 9, pp. 1927–1941, 2018.

[9] Y. Wang, Z.-Y. Ru, K. Wang, and P.-Q. Huang, “Joint deployment and task scheduling optimization for large-scale mobile users in multi-uav-enabled mobile edge computing,” *IEEE transactions on cybernetics*, vol. 50, no. 9, pp. 3984–3997, 2019.

[10] 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.

[11] Y. Luo, W. Ding, and B. Zhang, “Optimization of task scheduling and dynamic service strategy for multi-uav-enabled mobile edge computing system,” *IEEE Transactions on Cognitive Communications and Networking*, 2021.

[12] A. Goldsmith, *Wireless communications*. Cambridge university press, 2005.

[13] J. Xiong, H. Guo, and J. Liu, “Task offloading in uav-aided edge computing: Bit allocation and trajectory optimization,” *IEEE Communications Letters*, vol. 23, no. 3, pp. 538–541, 2019.

[14] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, “Network flows,” 1988.

[15] J. Holden and N. Goel, “Fast-forwarding to a future of on-demand urban air transportation,” *San Francisco, CA*, 2016.

[16] J. Chen, D. DeLaurentis, and D. Sun, “Dynamic stochastic model for converging inbound air traffic,” *Journal of Guidance, Control, and Dynamics*, vol. 39, no. 10, pp. 2273–2283, 2016.

[17] J. Chen, “Integrated routing and charging scheduling for autonomous electric aerial vehicle system,” in *2019 IEEE/AIAA 38th Digital Avionics Systems Conference (DASC)*. IEEE, 2019, pp. 1–7.

Authorized licensed use limited to: Beijing Information Science & Tech Univ. Downloaded on March 01,2023 at 13:09:59 UTC from IEEE Xplore. Restrictions apply.