QoS 제약이 있는 네트워크에서 tree-search를 활용한 UAV 기지국의 경로 계획 및 자원 최적화 기법

Tree Search-Based Trajectory Planning and Resource Management for a UAV-BS under QoS Constraints

초록

최근 IoT 네트워크를 서비스하는 UAV 기지국 제어 기법에 대한 많은 연구가 제안되었다. UAV 기지국은 이동이 가능하다는 장점 덕분에 넓은 지역에 분포해 있는 IoT 기기에 높은 데이터 전송율을 보장할 수 있다. 이 논문은 IoT 기기의 통신 주기와 QoS 제한 조건이 있는 네트워크에서 경로, 사용자 연결, 자원 할당 및 전력 제어를 통해 비례 공정성을 최대화하는 새로운 기법을 다룬다. 이 연구는 최적화 문제의 비선형성을 해결할 수 없는 iterative algorithm을 대체할 수 있는 깊이 우선 탐색 기법을 적용한 UAV 기지국의 경로 최적화 기법을 최초로 제안하였다. 시뮬레이션 결과는 제안한 기법이 세계 최고 기술과 비교했을 때 95% 높은 비례 공정성 값을 가지는 것을 보였다.

Key Words : UAV base station, trajectory-planning, user association, resource allocation, power control, depth-first search, quality-of-service, request period.

1 Introduction

Communication technologies for the next-generation IoT networks have attracted substantial research interest, due to the prevalence of IoT technologies such as smart homes, smart farming, wearable medical devices, and logistics automation. According to Cisco's technical report(1), the number of IoT devices is expected to increase ninefold, from 8.4 billion in 2017 to 75.4 billion in 2023, thereby accounting for more than half of total IP traffic in 2023. For these reasons, it is pretty timely to consider practical communication systems for IoT devices.

Wireless communication technologies for practical IoT networks need to consider a characteristic of IoT networks: time-critical communications(2). That is, IoT devices share information by frequently exchanging a relatively small amount of data rather than requesting a large volume of data at a time due to the limited memory size. Furthermore, the devices are required to effectively use energy from their battery sources to prolong the device lifetime. Thus, IoT devices could save their energy by requesting short-term communications rather than maintaining network connections(3),(4). In cellular networks, one of the feasible solutions for serving widespread devices is deploying more ground base stations (GBSs), namely, ultra-dense networks. However, building additional GBSs is inefficient in providing access to IoT devices, because IoT devices are widely distributed and occasionally request communication services.

By virtue of their mobility, UAV base stations (UAV-BSs) have been studied as a promising technology for IoT networks(5),(6). UAV-BSs can cover a large service area while providing high signal-to-noise ratio (SNR) access links by flying close to on-demand users. Fig. 1 illustrates an example of the UAV-enabled IoT network with the trajectory planning of a UAV-BS. As depicted in the figure, the UAV-BS can enhance the network utility (e.g., sum-rate, energy efficiency, sum-rate, and so on) by carefully planning its trajectory to serve IoT devices.

Nonetheless, several obstacles arising from the mobility of UAV-BSs still need to be addressed. The maneuver of UAV-BSs causes time-varying channels, because the users' channel states are significantly influenced by the relative position of UAV-BSs that changes over time(7). For the same reason, the trajectory of UAV-BSs cannot be independently optimized without considering communication factors such as user association (UA), resource allocation (RA) and power control (PC)(8),(9).

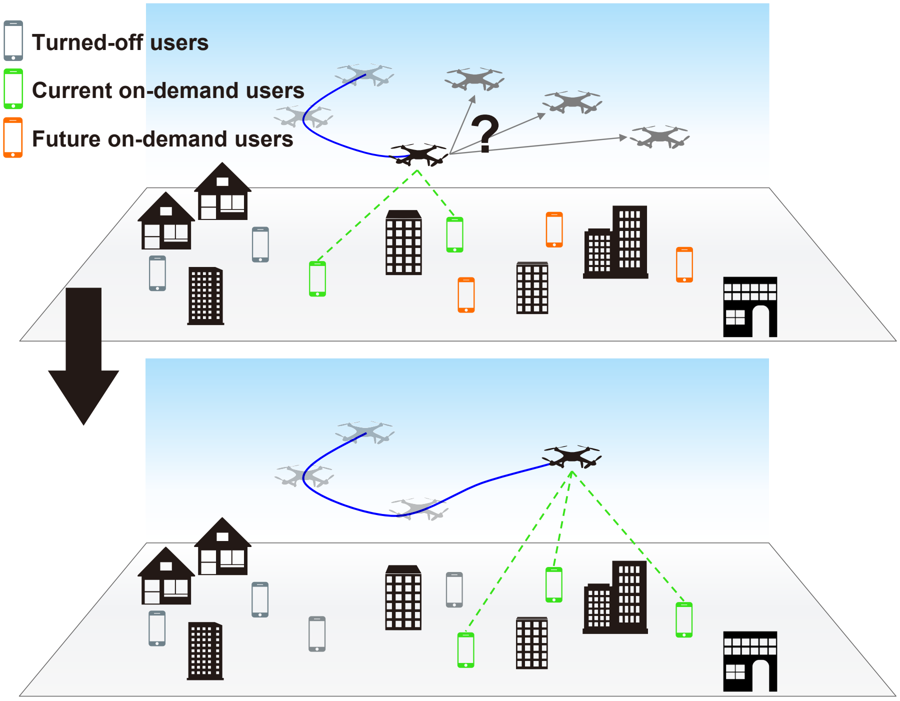


Figure 1: Illustration of a UAV-BS-enabled IoT network. Here, the IoT devices are deployed and periodically turned on/off to spend energy efficiently. The UAV-BS needs to set up its trajectory with the consideration of the on/off status of the devices.

Summarizing the aforementioned problems, trajectory-planning of UAV-BSs has the following main challenges:

- **Local optimality of trajectory-planning**: In the UAV-BS trajectory-planning problems, applying iterative methods(10) that alternately optimize trajectory and other optimization variables might degrade the utility of the network, although these methods have been widely used in the trajectory-planning problems(3,4,11–16). This degradation is caused by the non-convexity of network utility functions, which coerces iterative methods to find a local optimal solution. Iterative methods require a randomly chosen initial trajectory to optimize UA, RA, and PC variables. Then, a large change in the trajectory might decrease the network utility because the UA, RA, and PC variables are correspondingly updated with channel states obtained from the initial trajectory. In other words, if the near-optimal trajectories are remote from the arbitrarily chosen initial trajectory, they might not be found because future updates of the trajectory will be entrapped in the initial trajectory.

- **Intractably large parameter space**: Especially for the UA, the cardinality for the feasible set of the UA variables exponentially increases with the length of the total service time and the number of users. For instance, the total number of possible UA variable combinations is $2^{10\times 20} \approx 10^{60}$ if we design a UA problem with a single UAV-BS for 20 time slots and 10 users. To make things worse, the joint design of UA, RA, and PC becomes more intractable because the additional RA and PC variables are closely coupled with the UA and trajectory variables. Thus, designing an algorithm that achieves optimum (or sub-optimum with a small gap) within a reasonable computation time budget remains challenging.

1.1 Contributions

We resolve the above challenges by adopting a DFS algorithm that enables a UAV-BS to design an impromptu trajectory containing waypoints for the next few time slots. Specifically, we suggest a trajectory-planning problem for a single UAV-BS where IoT devices periodically request data stream over a certain data rate to guarantee their service requirements. Additionally, we optimize UA, RA, and PC variables, considering the change of channel states arising from the change of the UAV-BS position. To design the optimal trajectory, UA, RA, and PC variables, we formulate an optimization problem to maximize the PF of users[1]. We note that the PF is an appropriate objective function for UAV-enabled IoT networks because efficiency and fairness can be considered jointly. The salient contributions of this paper are summarized as follows:

- **Temporal decomposition of the problem:** We introduce a method that decomposes the proposed problem into several sub-problems containing a small block of consecutive time slots. The proposed problem involves a large number of variables that are intricately combined in the objective function. Thus, we decompose the objective function by reformulating users' sum-rate over time into the product of a sum-rate ratio. This decomposition of the objective function enables the proposed problem to be temporally decoupled into the summation of the sub-problems. Then, a combinatorial explosion from the large parameter spaces can be avoided by solving sub-problems on behalf of the original problem.

- **First tree search algorithm for UAV-BS:** Without loss of generality, we reformulate the sub-problems into an equivalent tree structure that includes all possible sub-trajectories. Finding the optimal solution to the sub-problems is still challenging though we decompose the proposed problem into several sub-problems, because these sub-problems have severe non-convexity that mainly arises from the trajectory of a UAV-BS.

  To bypass the non-convexity, we separate the problem into trajectory-planning and resource management, then utilize a DFS algorithm for trajectory-planning. Our DFS algorithm can find an optimal trajectory of the sub-problems for a given resource management scheme. Under stringent QoS constraints, the proposed DFS algorithm has 37% and 80% higher PF values than the circular and fixed trajectory, respectively (Fig. 5a).

- **Near-optimal resource management algorithm:** Motivated by the Lagrangian method, we design an iterative algorithm to find the optimal UA, RA, and PC variables. We show that the proposed UA algorithm accomplishes 99.8% PF of the exhaustive search method, decreasing the computational complexity from $\mathcal{O}(2^I)$ to $\mathcal{O}(I^2)$ for I users. Given UA and PC variables, we show that the objective function of the reformulated problem is concave for the RA variables. Then, the optimal RA variables can be found by applying the gradient descent method. Moreover, the optimal PC variable can be obtained by searching 1-dimensional space for the given UA and RA variables.

To the best of our knowledge, this work is the first attempt that introduces a tree reformulation of UAV-BSs' trajectory-planning problems and adopts a tree-search method to design a UAV-BS trajectory.

1.2 Related Works

1.2.1 UAV-BSs in Time-Critical Mobile Scenarios

Several research works have studied the control of UAV-BSs in time-critical wireless sensor network(3,4,16,17). Samir et al.(3) and Al-Hilo et al.(4) target to maximize the number of devices that successfully transmit data within the lifetime of data. Two algorithms are suggested to jointly solve 2-dimensional trajectory and bandwidth allocation of devices(3). Reconfigurable intelligent surfaces are additionally considered and jointly optimized with the trajectory of a UAV-BS by using a deep-reinforcement learning scheme(4). Hu et al.(16) propose an age of information minimization problem in wireless sensor network, where sensors are powered by UAV-BS using energy harvesting. These works enhance the network utility for timely data of wireless devices, but the fairness of the users is not considered. Without considering fairness, the trajectory of a UAV-BS can be biased according to the QoS constraints and channel state of users, thereby decreasing the number of served IoT devices.

1.2.2 UAV-BSs for Wireless Sensor Networks

You and Zhang(18) addresses a problem that maximizes the minimum average data rate in wireless sensor networks with angle-dependent Rician fading and the outage probability constraints. Zhan et al.(6) reformulate

[1] In this paper, the terms 'IoT device' and 'user' are used interchangeably.

Table 1: Summary of the related works

Ref.*	3D Traj.*	UA	RA	PC	QoS	Fairness	TCN*
(3)		✓	✓				✓
(4,18)		✓					✓
(6,19)	✓						
(14)	✓			✓			
(15)				✓			
(16,17)		✓			✓		✓
(20)	✓	✓				✓	
(21)		✓					
(22)	✓			✓	✓		
(12,13)	✓	✓	✓	✓	✓	✓	
Ours	✓	✓	✓	✓	✓	✓	✓

* Ref.: Reference, Traj.: Trajectory, TCN: Time-critical networks.

the trajectory-planning problem into an equivalent traveling salesman problem to maximize the number of visited wireless sensors. Zhu and Wang(19) suggest a trajectory-planning algorithm that minimizes the required time slots for collecting data from all wireless sensors. Zhu et al.(20) adopt a deep-reinforcement learning model to minimize energy consumption while collecting data from wireless sensors. These studies provide a well-designed trajectory in various mobile scenarios, but designing optimal RA and PC still remains unsolved.

1.2.3 Trajectory-Planning of UAV-BSs

There are several research works(14,15,21,22) that propose a trajectory-planning problem to maximize the sum-rate of users in UAV-enabled networks. These works significantly enhance the sum-rate by adopting reinforcement-learning (RL) methods(21,22) or utilizing successive convex approximation(14,15). However, none of these works considers the fairness of users and RA of UAV-BSs.

Zeng et al.(12,13) propose a trajectory-planning algorithm that jointly optimizes RA and PC parameters to maximize the fairness of users in the orthogonal frequency-division multiplexing access (OFDMA) network, where a single UAV-BS is utilized as a relay node. These algorithms sequentially find the optimal trajectory, RA, and PC parameters by optimizing these parameters for the next time slot based on the current location of the UAV-BS. The problems designed in these works are well-established, but the time-varying channels of subcarriers generated from the movement of a UAV-BS still need to be considered.

1.3 Paper Organizations

Section 2 introduces the system model of the IoT networks served by a UAV-BS and its mathematical representations. Section 3 formulates the proposed proportional fairness (PF) maximization problem of trajectory-planning, UA, RA, and PC (in Sec. 3.1) and decomposes the problem into sub-problems (n-step lookahead problems) for applying the DFS algorithm (in Sec. 3.2). Section 4 presents the solution to the sub-problems. Section 5 provides numerical evaluations of our proposed method. Section 6 concludes the paper.

2 System Model

2.1 Scenario Description

We consider an IoT network where a single UAV-BS serves I IoT devices for T service time, as previously illustrated in Fig. 1. In the network, the UAV-BS selectively provides service to users due to the limitations in the available bandwidth B and power P. Each user has a short-term period to request an access link and a QoS rate to satisfy its service requirements. The users are indexed by the index set $\mathcal{I} = \{1, \dots, I\}$.

We assume that the flight map and service timeline are discretized. The time slots are indexed by $\mathcal{T} = \{1, \dots, T\}$, the length of which is ΔT. The UAV-BS covers a square-shaped ground area with the size of w^2 m^2, and the flight map is represented by a set $\mathcal{Q}$ of three-dimensional grids discretized by an interval ΔQ as follows:

$$\mathcal{Q} = \Bigg\{ \Delta Q[i,j,k]^{\mathrm{T}} | [i,j,k]^{\mathrm{T}} \in \mathbb{N}^3, \\ i,j \in \left[0, \frac{w}{\Delta Q}\right], k \in \left[\frac{h_{\text{min}}}{\Delta Q}, \frac{h_{\text{max}}}{\Delta Q}\right] \Bigg\}, \tag{1}$$

where $h_{\min}$ and $h_{\max}$ are the minimum and maximum altitude of the UAV-BS, respectively.

When the t-th time slot begins, the UAV-BS must be located on the grid point of the flight map. We define $\mathbf{q}^{(t)}$ as the position of the UAV-BS at the beginning of t-th time slot, then $\mathbf{q}^{(t)}$ should satisfy

$$\mathbf{q}^{(t)} = [x^{(t)}, y^{(t)}, h^{(t)}]^{\mathrm{T}} \in \mathcal{Q}, \tag{2}$$

where the initial position of the UAV-BS is denoted as $\mathbf{q}^{(0)}$. The UAV-BS cannot exceed its maximum velocity v. In other words, the position vector $\mathbf{q}^{(t)}$ is constrained by

$$\|\mathbf{q}^{(t)} - \mathbf{q}^{(t-1)}\|_2 \leq v\Delta T, \ \forall t \in \mathcal{T}. \tag{3}$$

Also, the i-th user is located at $\mathbf{q}_i = [x_i, y_i, 0]^{\mathrm{T}}$. To prolong the device lifetime, each user requests a downlink service only if the current flight time slot t is within the short-term period $[s_i, s_i + T_i]$. We define a binary indicator $d_i^{(t)}$ to specify user indices requesting services at time slot t as follows:

$$d_i^{(t)} = \begin{cases} 1, & \text{if } s_i \leq t \leq s_i + T_i \\ 0, & \text{otherwise.} \end{cases} \tag{4}$$

For each time slot t, the UAV-BS determines a set of users to provide access links. Accordingly, we define the UA variables as

$$\alpha_i^{(t)} = \begin{cases} 1, & \text{if the UAV-BS serves user } i \text{ at time slot } t \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

We denote $\beta_i^{(t)}$ as the amount of frequency resources that the UAV-BS allocates to the i-th user at time slot t. The summation of the allocated bandwidth must not exceed the available bandwidth B, so we have the following constraint:

$$\sum_{i \in \boldsymbol{\alpha}^{(t)}} \beta_i^{(t)} \leq B, \ \forall t \in \mathcal{T}, \tag{6}$$

where $\boldsymbol{\alpha}^{(t)} = \{i \in \mathcal{I} | \alpha_i^{(t)} = 1\}$ denotes the set of associated users at time slot t.

The UAV-BS regulates its transmission power by adjusting the power spectral density (PSD). The variable $\rho_i^{(t)}$ is defined as the assigned PSD to the i-th user at time slot t. Denoting the maximum transmission power at each time slot as P, the transmission power is constrained as follows:

$$\sum_{i \in \boldsymbol{\alpha}^{(t)}} \rho_i^{(t)} \beta_i^{(t)} \leq P, \ \forall t \in \mathcal{T}. \tag{7}$$

2.2 Pathloss and Data Rate Model

The propagation channel of the i-th user follows a probabilistic LoS channel[23]. The probability that the i-th user has an LoS wireless link is defined as

$$P_i^{(t)} = \frac{1}{1 + a \cdot \exp(-b(\theta_i^{(t)} - a))}, \tag{8}$$

where $\theta_i^{(t)} = \arcsin\left(\frac{h^{(t)}}{\|\mathbf{q}^{(t)} - \mathbf{q}_i\|_2}\right)$ is the elevation angle between the UAV-BS and the i-th user at time slot t. Variable a and b in (8) are environmental parameters that are determined according to the environmental characteristics[7]. We remark that the i-th user has an NLoS link at time slot t with a probability of $1 - P_i^{(t)}$. The average pathloss for the i-th user at time slot t is defined as

$$\begin{aligned} \xi_i^{(t)} =& 20 \log \left(\frac{4\pi f \|\mathbf{q}^{(t)} - \mathbf{q}_i\|_2}{c} \right) \\ &+ P_i^{(t)} \eta_{\text{LoS}} + (1 - P_i^{(t)}) \eta_{\text{NLoS}}, \end{aligned} \tag{9}$$

where f is the carrier frequency and c is the speed of light. The first term of (9) is the free-space pathloss and the sum of the remaining terms is expected excessive pathloss obtained from averaging the excessive pathloss of LoS and NLoS links[23].

The data rate of the i-th user at time slot t is defined as

$$R_i^{(t)} = d_i^{(t)} \beta_i^{(t)} \log_2 \left(1 + \frac{\rho_i^{(t)} 10^{-\xi_i^{(t)}/10}}{N_0} \right), \tag{10}$$

where constant N_0 denotes noise PSD.

To ensure the served users' QoS requirements, the UAV-BS should serve user i with a link that has a data rate higher than r_i if the i-th user receives service at time slot t. Therefore, we have

$$R_i^{(t)} \geq \alpha_i^{(t)} r_i. \tag{11}$$

We define R_i as the sum-rate of the i-th user across the UAV-BS service timeline, denoted as

$$R_i = \sum_{t \in \mathcal{T}} R_i^{(t)}. \tag{12}$$

Then, the PF of the served users is written by $\sum_{i \in \boldsymbol{\alpha}} \log R_i$. We note that this objective function can consider both the total network throughput and the number of served users, because the log term makes the UAV-BS serve more users for enhancing the PF unless the sum-rate of users is too low.

3 Problem Formulation

3.1 Original Problem

By collecting the aforementioned constraints, the joint problem of trajectory-planning, UA, RA, and PC for the PF maximization is formulated as

$$
\begin{aligned}
\mathbf{P1}:\ \max_{\mathbf{q},\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\rho}} & \sum_{i\in\boldsymbol{\alpha}} \log R_i && \text{(13a)}\\
\text{s.t.}\ \ & \mathbf{q}^{(t)} \in \mathcal{Q}, && \text{(13b)}\\
& \|\mathbf{q}^{(t)} - \mathbf{q}^{(t-1)}\|_2 \le v\Delta T, && \text{(13c)}\\
& \beta_i^{(t)} \ge 0, \forall i \in \mathcal{I}, && \text{(13d)}\\
& \sum_{i\in\boldsymbol{\alpha}^{(t)}} \beta_i^{(t)} \le B, && \text{(13e)}\\
& \rho_i^{(t)} \ge 0, \forall i \in \mathcal{I}, && \text{(13f)}\\
& \sum_{i\in\boldsymbol{\alpha}^{(t)}} \rho_i^{(t)} \beta_i^{(t)} \le P, && \text{(13g)}\\
& R_i^{(t)} \ge \alpha_i^{(t)} r_i, \forall i \in \mathcal{I}, && \text{(13h)}\\
& \forall t \in \mathcal{T} \text{ for (13b)-(13h).}
\end{aligned}
$$

For brevity of the notations, we define augmented vectors and matrices[2] of variables as follows: $\mathbf{q} = [\mathbf{q}^{(t)}]^{t\in\mathcal{T}}$, $\boldsymbol{\beta} = [\beta_i^{(t)}]_{i\in\mathcal{I}}^{t\in\mathcal{T}}$ and $\boldsymbol{\rho} = [\rho_i^{(t)}]_{i\in\mathcal{I}}^{t\in\mathcal{T}}$. Also, $\boldsymbol{\alpha} = \bigcup_{t\in\mathcal{T}} \boldsymbol{\alpha}^{(t)}$ represents a set of users that have been served at least once during the UAV-BS service timeline.

Constraints (13b) to (13h) are described as follows: (13b) and (13c) correspond to (2) and (3), respectively; (13d) and (13f) represent the non-negativity of the allocated bandwidth and PSD, respectively; (13e), (13g) and (13h) are equivalent with (6), (7) and (11), respectively.

Problem $\mathbf{P1}$ is a non-convex mixed-integer problem that is generally known to be NP-hard(24), because the association variables $\alpha_i^{(t)}, \forall i, t$ are binary integers and the objective function is a non-convex function for the position vector $\mathbf{q}^{(t)}$. Hence, solving the problem within a feasible complexity becomes challenging if a large number of time slots is involved with Problem $\mathbf{P1}$, because Problem $\mathbf{P1}$ optimizes related parameters over the entire service timeline $\mathcal{T}$. To address the challenges, we propose a method that decomposes Problem $\mathbf{P1}$ into sub-problems that are independent of the number of time slots T.

3.2 Reformulation by Temporal Decoupling

Here, we first decompose Problem $\mathbf{P1}$ into sub-problems, each of which corresponds to a single time slot. We call the problem for a single time slot the 1-step lookahead problem. Then, by merging a number of time slots in a time slot block, we formulate the n-step lookahead problem, which corresponds to the optimization problem for n time slots. We note that the original problem $\mathbf{P1}$ can be solved by optimizing the sub-problems sequentially.

3.2.1 1-step Lookahead Problem

Since the trajectory-planning, UA, RA, and PC variables in different time slots are closely coupled in the objective function of $\mathbf{P1}$, we relax the objective function by adopting the following proposition.

[2] We define a vector-building operator $[e^{(t)}]^{t\in\mathcal{T}}$ to represent $[e^{(1)}, e^{(2)}, \dots, e^{(T)}]^T$. Then we define a matrix-building operator $[e_i^{(t)}]_{i\in I}^{t\in\mathcal{T}}$ as $[[e_1^{(t)}]^{t\in\mathcal{T}}, [e_2^{(t)}]^{t\in\mathcal{T}}, \dots, [e_I^{(t)}]^{t\in\mathcal{T}}]$.

Proposition 1 (Lower bound of Problem $\mathbf{P1}$). *The following inequality holds:*

$$\begin{aligned} &\max_{\mathbf{q},\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\rho}} \sum_{i\in\boldsymbol{\alpha}} \log R_i \\ &\geq \sum_{t=1}^{T} \max_{\mathbf{q}^{(t)},\boldsymbol{\alpha}^{(t)},\boldsymbol{\beta}^{(t)},\boldsymbol{\rho}^{(t)}} \sum_{i\in\boldsymbol{\alpha}^{(t)}} \log\left(1+\frac{R_i^{(t)}}{\sum_{k=0}^{t-1} R_i^{(k)}}\right). \end{aligned} \tag{14}$$

Proof. The proof is shown in Appendix A. ■

In Proposition 1, we decompose the original PF function for each time slot, whereas the relaxed form is a lower bound of the original objective function. The tightness of the lower bound will be considered in the next section.

With the relaxed objective function (14), we can formulate the sub-problem $\mathbf{P}^{(t)}$ to optimize the variables for time slot t as follows:

$$\mathbf{P}^{(t)}: \max_{\mathbf{q}^{(t)},\boldsymbol{\alpha}^{(t)},\boldsymbol{\beta}^{(t)},\boldsymbol{\rho}^{(t)}} \sum_{i\in\boldsymbol{\alpha}^{(t)}} \log\left(1+\frac{R_i^{(t)}}{\sum_{k=0}^{t-1} R_i^{(k)}}\right), \tag{15a}$$

$$\text{s.t.} \qquad \text{(13b)} - \text{(13h)}, \tag{15b}$$

where $\boldsymbol{\beta}^{(t)} = [\beta_i^{(t)}]_{i\in\mathcal{I}}$ and $\boldsymbol{\rho}^{(t)} = [\rho_i^{(t)}]_{i\in\mathcal{I}}$. In the relaxed problem $\mathbf{P}^{(t)}$, the variables for time slot t are optimized for given variables at previous time slots $k < t$. Therefore, Problem $\mathbf{P}^{(t)}, t \in \mathcal{T}$ can be solved by sequentially optimizing $\mathbf{q}^{(t)}$, $\boldsymbol{\alpha}^{(t)}$, $\boldsymbol{\beta}^{(t)}$, and $\boldsymbol{\rho}^{(t)}$ from $t = 1$ to $t = T$.

To alleviate the non-convexity caused by the position vector $\mathbf{q}^{(t)}$, we separate Problem $\mathbf{P}^{(t)}$ into two parts: trajectory-planning and resource management. Here, for brevity, the spanning function $S(\cdot)$ maps a position vector $\mathbf{q} \in \mathcal{Q}$ to a set of reachable grid points for a time slot, denoted as

$$S(\mathbf{q}) = \{\mathbf{q}' \in \mathcal{Q} |\ \|\mathbf{q}' - \mathbf{q}\|_2 \leq v\Delta T\}. \tag{16}$$

Then, without loss of generality, Problem $\mathbf{P}^{(t)}$ can be reformulated as

$$\max_{\mathbf{q}^{(t)}\in S(\mathbf{q}^{(t-1)})} f(\mathbf{q}^{(t)}), \tag{17}$$

where $f(\mathbf{q}^{(t)})$ is defined as

$$f(\mathbf{q}^{(t)}) = \max_{\boldsymbol{\alpha}^{(t)},\boldsymbol{\beta}^{(t)},\boldsymbol{\rho}^{(t)}} \sum_{i\in\boldsymbol{\alpha}^{(t)}} \log\left(1+\frac{R_i^{(t)}}{\sum_{k=0}^{t-1} R_i^{(k)}}\right) \tag{18a}$$

$$\text{s.t.} \quad \text{(13d)} - \text{(13h)}. \tag{18b}$$

Here, we formulate a 1-step lookahead problem by decomposing the original problem into sub-problems for each time slot. Next, we extend the 1-step lookahead problems to n-step lookahead problems to consider multiple time slots at once.

3.2.2 n-Step Lookahead Problem

We first slice the time slots $\mathcal{T}$ into several blocks, the length of which is n; next, we formulate the optimization problem for each block of time slots, i.e., n-step lookahead problem. We denote the n-step lookahead problem for the k-th block as $\mathbf{P}_n^{(k)}$.

Let us define the k-th block of the sliced time slots as $\mathcal{T}_n^{(k)} = \{t \in \mathcal{T} \cap [1 + (k-1)n, kn]\}$. Then, as we separate Problem $\mathbf{P}^{(t)}$ into trajectory-planning and resource management in (17), we define Problem $\mathbf{P}_n^{(k)}$ by extending (17) as

$$\mathbf{P}_n^{(k)}: \max_{\substack{\mathbf{q}^{(t)}\in S(\mathbf{q}^{(t-1)}),\\ t\in\mathcal{T}_n^{(k)}}} \sum_{t\in\mathcal{T}_n^{(k)}} f(\mathbf{q}^{(t)}). \tag{19}$$

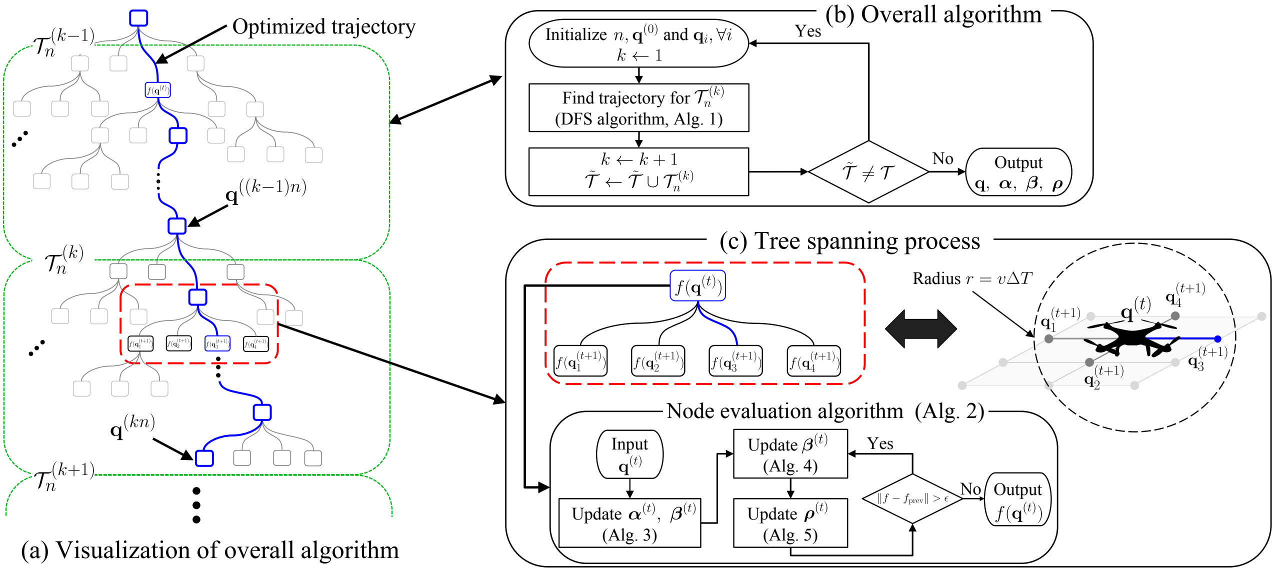
We note that the 1-step lookahead problem $\mathbf{P}^{(k)}$ is a special case of n-step lookahead problem $\mathbf{P}_n^{(k)}$ for $n = 1$. Similar to Problem $\mathbf{P}^{(t)}$, Problem $\mathbf{P}_n^{(k)}$ optimizes the k-th block of n time slots given the previously optimized trajectory, UA, RA, and PC variables.

For sufficiently large K satisfying $\bigcup_{k=1}^{K} \mathcal{T}_n^{(k)} = \mathcal{T}$, the following inequality holds:

$$\max_{\mathbf{q},\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\rho}} \sum_{i\in\boldsymbol{\alpha}} \log R_i \geq \sum_{k=1}^{K} \max_{\substack{\mathbf{q}^{(t)}\in S(\mathbf{q}^{(t-1)}),\\ t\in\mathcal{T}_n^{(k)}}} \sum_{t\in\mathcal{T}_n^{(k)}} f(\mathbf{q}^{(t)}), \tag{20}$$

where the inequality goes tighter as $n \to T$. Therefore, summation of Problem $\mathbf{P}_n^{(k)}$ for $k \in \{1, \ldots, K\}$ is also a lower bound of Problem $\mathbf{P1}$. To summarize, we maximize the lower bound of Problem $\mathbf{P1}$ by using a tree search algorithm, rather than directly solving Problem $\mathbf{P1}$, where the tightness is stronger if n goes to T. In numerical results (Sec. 5), we show that the PF performance asymptotically converges at $n = 5$.

4 Proposed solution



$\mathcal{T}_n^{(k-1)}$
Optimized trajectory
$f(\mathbf{q}^{(t)})$
$\mathbf{q}^{((k-1)n)}$
$\mathcal{T}_n^{(k)}$
$\mathbf{q}^{(kn)}$
$\mathcal{T}_n^{(k+1)}$
(a) Visualization of overall algorithm
(b) Overall algorithm
Initialize $n, \mathbf{q}^{(0)}$ and $\mathbf{q}_i, \forall i$
$k \leftarrow 1$
Find trajectory for $\mathcal{T}_n^{(k)}$
(DFS algorithm, Alg. 1)
$k \leftarrow k+1$
$\tilde{\mathcal{T}} \leftarrow \tilde{\mathcal{T}} \cup \mathcal{T}_n^{(k)}$
$\tilde{\mathcal{T}} \neq \mathcal{T}$
Yes
No
Output
$\mathbf{q},\ \boldsymbol{\alpha},\ \boldsymbol{\beta},\ \boldsymbol{\rho}$
(c) Tree spanning process
$f(\mathbf{q}^{(t)})$
$f(\mathbf{q}_1^{(t+1)})$
$f(\mathbf{q}_2^{(t+1)})$
$f(\mathbf{q}_3^{(t+1)})$
$f(\mathbf{q}_4^{(t+1)})$
Radius $r = v\Delta T$
$\mathbf{q}^{(t)}$
$\mathbf{q}_1^{(t+1)}$
$\mathbf{q}_2^{(t+1)}$
$\mathbf{q}_3^{(t+1)}$
$\mathbf{q}_4^{(t+1)}$
Node evaluation algorithm (Alg. 2)
Input
$\mathbf{q}^{(t)}$
Update $\boldsymbol{\alpha}^{(t)}$, $\boldsymbol{\beta}^{(t)}$
(Alg. 3)
Update $\boldsymbol{\beta}^{(t)}$
(Alg. 4)
Update $\boldsymbol{\rho}^{(t)}$
(Alg. 5)
$\|f - f_{\text{prev}}\| > \epsilon$
Yes
No
Output
$f(\mathbf{q}^{(t)})$

Figure 2: Overall algorithm overview.

This section describes an algorithm that optimizes the proposed problem. The illustration of the overall algorithm is depicted in Fig. 2.

The proposed algorithm solves Problem $\mathbf{P1}$ through three steps: i) temporal decomposition of the proposed problem (Fig. 2a), ii) depth-first search (DFS)-based trajectory-planning (Fig. 2b) and iii) resource management for jointly optimizing UA, RA, and PC variables (Fig. 2c).

As depicted in Fig. 2a, we decompose Problem $\mathbf{P1}$ into the n-step lookahead problems $\mathbf{P}_n^{(t)}$ (in Sec. 3.2). Then, we propose a DFS-based tree search algorithm to find the optimal trajectory for time slot blocks $\mathcal{T}_n^{(k)}$. Fig. 2b represents the block diagram of the overall algorithm. In the figure, we obtain the complete trajectory by concatenating the optimal trajectory for the time slot blocks $\mathcal{T}_n^{(k)}$. In the DFS algorithm, the UA, RA, and PC variables are assumed to be optimized according to the position of the UAV-BS, then the optimized UA, RA, and PC variables are utilized to evaluate the function $f(\mathbf{q}^{(t)})$ in (18). To compute the function $f(\mathbf{q}^{(t)})$, we propose a cascaded algorithm for UA, RA, and PC for position of the UAV-BS (Fig. 2c). We decompose the node evaluation problem (18) into three sub-problems: i) user association, ii) resource allocation and iii) power control. These three problems are iteratively optimized by finding the optimal solutions of the Lagrangian dual problems of (18).

4.1 Depth-First Search Algorithm

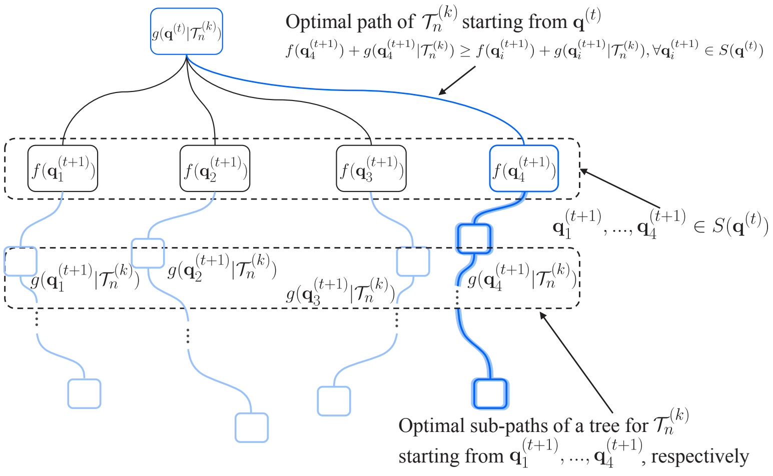

$g(\mathbf{q}^{(t)}|\mathcal{T}_n^{(k)})$
Optimal path of $\mathcal{T}_n^{(k)}$ starting from $\mathbf{q}^{(t)}$
$f(\mathbf{q}_4^{(t+1)}) + g(\mathbf{q}_4^{(t+1)}|\mathcal{T}_n^{(k)}) \geq f(\mathbf{q}_i^{(t+1)}) + g(\mathbf{q}_i^{(t+1)}|\mathcal{T}_n^{(k)}), \forall \mathbf{q}_i^{(t+1)} \in S(\mathbf{q}^{(t)})$
$f(\mathbf{q}_1^{(t+1)})$
$f(\mathbf{q}_2^{(t+1)})$
$f(\mathbf{q}_3^{(t+1)})$
$f(\mathbf{q}_4^{(t+1)})$
$\mathbf{q}_1^{(t+1)}, ..., \mathbf{q}_4^{(t+1)} \in S(\mathbf{q}^{(t)})$
$g(\mathbf{q}_1^{(t+1)}|\mathcal{T}_n^{(k)})$
$g(\mathbf{q}_2^{(t+1)}|\mathcal{T}_n^{(k)})$
$g(\mathbf{q}_3^{(t+1)}|\mathcal{T}_n^{(k)})$
$g(\mathbf{q}_4^{(t+1)}|\mathcal{T}_n^{(k)})$
Optimal sub-paths of a tree for $\mathcal{T}_n^{(k)}$ starting from $\mathbf{q}_1^{(t+1)}, ..., \mathbf{q}_4^{(t+1)}$, respectively

Figure 3: Visualization of recursive depth-first search algorithm in case that $\mathbf{n}_4$ is the optimal node when evaluating $g(\mathbf{q}|\mathcal{T}_n^{(k)})$ with (22). Non-optimal trajectories and nodes are omitted from the diagram.

The DFS algorithm aims to find the optimal trajectory $\mathbf{q}^{(t)},\ t \in \mathcal{T}_n^{(k)}$ for Problem $\mathbf{P}_n^{(k)}$, i.e., $\mathbf{q}^{(t)},\ t \in \mathcal{T}_n^{(k)}$ maximizing $\sum_{t\in\mathcal{T}_n^{(k)}} f(\mathbf{q}^{(t)})$. To solve the problem, we represent Problem $\mathbf{P}_n^{(k)}$ as a tree format (Fig. 3). We note that the optimal trajectory has the following property: the sub-trajectory of the optimal trajectory is the optimal sub-trajectory. Inspired by this property, the DFS algorithm can find the optimal trajectory. The pseudo-code of the DFS algorithm is indicated in Alg. 1.

For any given node $\mathbf{q} \in \mathcal{Q}$, we can span the node to its child nodes by depth 1 (Fig. 2b) by using the spanning mapping $S(\cdot)$. Then, the child nodes of $\mathbf{q}$ are elements of $S(\mathbf{q})$. We can make a tree with depth n by recursively spanning the child nodes n times from the initial node $\mathbf{q}^{((k-1)n)}$. Every trajectory that starts from the initial node (depth 0) and stops at the terminal node (depth n) is a candidate solution of Problem $\mathbf{P}_n^{(k)}$.

If we set the value of a node $\mathbf{q}$ as $f(\mathbf{q})$, a trajectory that maximizes the summation of values of nodes on the trajectory is equivalent to the optimal solution of (19). The detailed computation process of $f(\cdot)$ is provided in the next section.

For given $\mathcal{T}_n^{(k)}$, we define an auxiliary function $g : \mathcal{Q} \to \mathbb{R}$ to recursively compute function $f(\cdot)$, denoted as

$$g(\mathbf{q}^{(t)}|\mathcal{T}_n^{(k)}) = \max_{\substack{\mathbf{q}^{(t'+1)}\in S(\mathbf{q}^{(t')}),\\ t'>t, t'\in\mathcal{T}_n^{(k)}}} \sum_{\substack{t'>t,\\ t'\in\mathcal{T}_n^{(k)}}} f(\mathbf{q}^{(t')}) \tag{21}$$

$$= \max_{\mathbf{q}^{(t+1)}\in S(\mathbf{q}^{(t)})} f(\mathbf{q}^{(t+1)}) + g(\mathbf{q}^{(t+1)}|\mathcal{T}_n^{(k)}). \tag{22}$$

$g(\mathbf{q}^{(t)}|\mathcal{T}_n^{(k)})$ represents the summation of a node value for the optimal trajectory that starts from $\mathbf{q}^{(t)}$. In Fig. 3, we demonstrate how the function $g(\cdot)$ is related to the node evaluation function $f(\cdot)$. Then, the optimal value of Problem $\mathbf{P}_n^{(k)}$ can be computed by $g(\mathbf{q}^{((k-1)n)}|\mathcal{T}_n^{(k)})$, where $\mathbf{q}^{((k-1)n)}$ is the initial position of the $\mathcal{T}_n^{(k)}$. In Alg. 1, $g(\cdot)$ corresponds to function $\texttt{DFS}(\cdot)$.

The sub-optimal trajectory for Problem $\mathbf{P1}$ can be computed by concatenating trajectories of $P_n^{(k)}$ from $k = 1$ to $k = K$ for sufficiently large K satisfying $\bigcup_{k=1}^{K} \mathcal{T}_n^{(k)} = \mathcal{T}$ (Fig. 2c).

The DFS algorithm finds the optimal trajectory of Problem $\mathbf{P}_n^{(k)}$, assuming that $f(\mathbf{q})$ can be computed for any node $\mathbf{q}$ in the spanned tree. Hence, the remaining is the method for obtaining the value of $f(\mathbf{q})$. We note that computing $f(\mathbf{q})$ is NP-hard, because the problem (18) is still a non-concave and mixed-integer problem even though vector $\mathbf{q}$ is given.

Algorithm 1: Depth-First Search Trajectory-Planning

```
Input $n$, $\mathbf{q}^{(0)}$, $\mathbf{q}_i, \forall i \in \mathcal{I}$
Output $\mathbf{q}$
Initialize $k = 1$

Function DFS($\mathbf{q}^{(t)}$, $\mathrm{depth}$):
    if $\mathrm{depth} \geq n$ then
        return $[\mathbf{q}^{(t)}]^T$, $f(\mathbf{q}^{(t)})$
    end
    $\mathbf{Q}^* \leftarrow []$, $\Sigma^* \leftarrow 0$.
    $S \leftarrow S(\mathbf{q}^{(t)})$ (Eq. (16)).
    for $\mathbf{q}^{(t+1)}$ $\mathbf{in}$ $S$ do
        $\mathbf{Q}$, $\Sigma \leftarrow$ DFS($\mathbf{q}^{(t+1)}$, $\mathrm{depth} + 1$)
        if $\Sigma^* < \Sigma$ then
            $\mathbf{Q}^* \leftarrow \mathbf{Q}$, $\Sigma^* \leftarrow \Sigma$.
        end
    end
    $\mathbf{Q}^* \leftarrow [\mathbf{q}^{(t)}]^T \oplus \mathbf{Q}^*$ for concatenating operator $\oplus$.
    $\Sigma^* \leftarrow f(\mathbf{q}^{(t)}) + \Sigma^*$
    $\mathbf{return}$ $\mathbf{Q}^*$, $\Sigma^*$

while $\tilde{\mathcal{T}} \neq \mathcal{T}$ do
    $\mathcal{T}_n^{(k)} \leftarrow \{t \in \mathcal{T} \cap [1 + (k-1)n, kn]\}$.
    $[\mathbf{q}^{(t)}]^{t \in \mathcal{T}_n^{(k)}}, \Sigma \leftarrow$ DFS($\mathbf{q}^{((k-1)n)}$, $0$).
    $\tilde{\mathcal{T}} \leftarrow \tilde{\mathcal{T}} \cup \mathcal{T}_n^{(k)}$, $k \leftarrow k + 1$.
end
```

4.2 Node Evaluation Function $f(\mathbf{q})$

To evaluate the function $f(\mathbf{q})$, we iteratively optimize the UA, RA, and PC variables. The overall node evaluation algorithm is summarized in Alg. 2. In Line 4, we obtain initial UA and RA variables for given $\rho_i^{(t)}$ (Alg. 3). After we initialize the UA variable, the problem (18) can be decomposed into two sub-problems: one for $\boldsymbol{\beta}^{(t)}$ (solved in Alg. 4) and the other for $\boldsymbol{\rho}^{(t)}$ (solved in Alg. 5). We can obtain the optimal UA, RA, and PC variables and the node value $f(\mathbf{q})$ by initializing UA and RA variables and then iteratively solving these two sub-problems. The remainder of this section describes the initialization and two sub-problems in detail.

Algorithm 2: Node Evaluation Function

```
1:  Input $\mathbf{q}^{(t)}$, $\rho_i^{(t)} = P/B, \forall i \in \mathcal{I}$
2:  Output $f(\mathbf{q}^{(t)})$, $\boldsymbol{\alpha}^{(t)}$, $\boldsymbol{\beta}^{(t)}$, $\boldsymbol{\rho}^{(t)}$

3:  Function $f(\mathbf{q}^{(t)})$:
4:      Update $\boldsymbol{\alpha}^{(t)}$ and $\boldsymbol{\beta}^{(t)}$ by using Alg. 3.
5:      $f \leftarrow 0$, $f_{\mathrm{prev}} \leftarrow \infty$.
6:      while $\|f - f_{\mathrm{prev}}\| > \epsilon$ do
7:          Update $\boldsymbol{\beta}^{(t)}$ by using Alg. 4.
8:          Update $\boldsymbol{\rho}^{(t)}$ by using Alg. 5.
9:          $f_{\mathrm{prev}} \leftarrow f$.
10:         $f \leftarrow \sum_{i \in \boldsymbol{\alpha}^{(t)}} \log(1 + \frac{R_i^{(t)}}{\sum_{k=0}^{t-1} R_i^{(k)}})$
11:     end
```

4.2.1 Initial User Association and Resource Allocation

```
Algorithm 3: User Association Algorithm
Input $\mathrm{q}^{(t)}$, $\boldsymbol{\rho}^{(t)}$
Output $\boldsymbol{\alpha}^{(t)}$, $\boldsymbol{\beta}^{(t)}$
$\boldsymbol{\alpha}^{(t)} \leftarrow \{\}$, $f^* \leftarrow 0$.
while $\hat{\boldsymbol{\alpha}}^{(t)} \neq \boldsymbol{\alpha}^{(t)}$ do
    $\hat{\boldsymbol{\alpha}}^{(t)} \leftarrow \boldsymbol{\alpha}^{(t)}$
    for $i$ in $\mathcal{I}$ do
        $\hat{\boldsymbol{\alpha}}_i^{(t)} \leftarrow \hat{\boldsymbol{\alpha}}^{(t)} \cup \{\alpha_i^{(t)}\}$
        if $\hat{\boldsymbol{\alpha}}_i{}^{(t)}$ does not satisfies (26) then
            continue
        end
        Compute $\boldsymbol{\beta}^{(t)}$ for $\hat{\boldsymbol{\alpha}}_i^{(t)}$ by using (24).
        $f \leftarrow$ the value of (23a) for $\hat{\boldsymbol{\alpha}}_i^{(t)}$ and $\boldsymbol{\beta}^{(t)}$.
        if $f > f^*$ then
            $f^* \leftarrow f$, $\boldsymbol{\alpha}^{(t)} \leftarrow \hat{\boldsymbol{\alpha}}_i^{(t)}$.
        end
    end
end
Compute $\boldsymbol{\beta}^{(t)}$ for $\boldsymbol{\alpha}^{(t)}$ by using (24).
```

We first assume that the PSD variables $\rho_i^{(t)}$ are equally assigned as $\rho_i^{(t)} = P/B$ for all $i \in \mathcal{I}$. Then the problem (18) can be simplified as

$$\max_{\boldsymbol{\alpha}^{(t)},\boldsymbol{\beta}^{(t)}} \sum_{i\in\mathcal{I}} \log\left(1 + \frac{R_i^{(t)}}{\sum_{k=0}^{t-1} R_i^{(k)}}\right) \tag{23a}$$

$$\text{s.t.} \quad \beta_i^{(t)} \geq \alpha_i^{(t)} r_i / e_i^{(t)}, \forall i, \tag{23b}$$

$$\sum_{i\in\boldsymbol{\alpha}^{(t)}} \beta_i^{(t)} \leq B, \tag{23c}$$

where $e_i^{(t)} = \log_2(1 + \frac{\rho_i^{(t)} 10^{-\xi_i^{(t)}/10}}{N_0})$ is the spectral efficiency of the i-th user at time slot t.

To obtain the optimality condition of $\boldsymbol{\beta}^{(t)}$ for the problem (23), we apply the Karush-Kuhn-Tucker (KKT) condition to the problem. Then, the KKT condition implies that the problem can be solved by the water-filling algorithm with non-zero QoS constraints. In other words, for arbitrary $\boldsymbol{\alpha}^{(t)}$, the variable $\beta_i^{(t)}$ can be obtained by

$$\beta_i^{(t)} = \begin{cases} \max(\frac{r_i}{e_i^{(t)}}, \frac{1}{\lambda} - \frac{\sum_{k=0}^{t-1} R_i^{(k)}}{e_i^{(t)}}) & \text{if } \alpha_i^{(t)} = d_i^{(t)} = 1, \\ 0 & \text{otherwise,} \end{cases} \tag{24}$$

where λ can be uniquely determined to satisfy the constraint (23c) with equality condition. Since the variable λ is a scalar, the optimal value of the variable can be obtained by one-dimension search methods[3]. For the UA optimization, we propose an incremental algorithm that finds a local optimal solution of $\boldsymbol{\alpha}^{(t)}$ (Alg. 3). If an arbitrary feasible UA set $\hat{\boldsymbol{\alpha}}^{(t)}$ is given, we can associate a new user $i,\ i \notin \boldsymbol{\alpha}^{(t)}$ with the UAV-BS. Then, a new UA set $\boldsymbol{\alpha}_i^{(t)}$ is defined by changing one UA variable $\alpha_i^{(t)}, i \notin \boldsymbol{\alpha}^{(t)}$ from 0 to 1 as follows:

$$\hat{\boldsymbol{\alpha}}_i^{(t)} = \hat{\boldsymbol{\alpha}}^{(t)} \cup \{\alpha_i^{(t)}\}. \tag{25}$$

[3] We use the bisection method to find the corresponding λ^*.

We can determine the feasibility of the set by checking whether $\hat{\boldsymbol{\alpha}}_i^{(t)}$ satisfies

$$\sum_{i\in\hat{\boldsymbol{\alpha}}_i^{(t)}} \frac{r_i}{e_i^{(t)}} \leq B. \tag{26}$$

We can compute the objective function of the problem (23) for any feasible UA set, because RA variables are optimally determined by (24). Then, we can find the optimal $\boldsymbol{\alpha}^{(t)}$, comparing the maximum objective calculated by $\hat{\boldsymbol{\alpha}}^{(t)}$ and $\hat{\boldsymbol{\alpha}}_i^{(t)}, i \notin \boldsymbol{\alpha}^{(t)}$.

Starting from the empty UA set $\hat{\boldsymbol{\alpha}}^{(t)} = \{\}$, the local optimal UA set $\boldsymbol{\alpha}^{(t)}$ can be found by repeatedly implementing the above procedure until we reach the following convergence condition:

$$\boldsymbol{\alpha}^{(t)} = \hat{\boldsymbol{\alpha}}^{(t)}. \tag{27}$$

In the numerical results, we show that the proposed UA method achieves the 99.8% PF of its upper bound at the later section.

4.2.2 Resource Allocation

If $\boldsymbol{\alpha}^{(t)}$ and $\boldsymbol{\rho}^{(t)}$ are fixed, the problem (18) is redefined as

$$\max_{\boldsymbol{\beta}^{(t)}} \sum_{i\in\boldsymbol{\alpha}^{(t)}} \log\left(1 + \frac{R_i^{(t)}}{\sum_{k=0}^{t-1} R_i^{(k)}}\right) \tag{28a}$$

$$\text{s.t.} \sum_{i\in\boldsymbol{\alpha}^{(t)}} \beta_i^{(t)} \leq B, \tag{28b}$$

$$\sum_{i\in\boldsymbol{\alpha}^{(t)}} \rho_i^{(t)}\beta_i^{(t)} \leq P, \tag{28c}$$

$$\beta_i^{(t)} \geq \alpha_i^{(t)} r_i / e_i^{(t)}, \forall i. \tag{28d}$$

The optimal solution of the problem (28) can be obtained by using Lagrangian dual method. Then, $\boldsymbol{\alpha}^{(t)}$ can be optimized by globally searching all feasible $\boldsymbol{\alpha}^{(t)}$ for the objective function of (28).

The Lagrangian dual problem of (28) is

$$\min_{\boldsymbol{\mu}^{(t)},\lambda_1^{(t)},\lambda_2^{(t)}} \mathcal{L}_{\boldsymbol{\beta}}^*(\boldsymbol{\mu}^{(t)}, \lambda_1^{(t)}, \lambda_2^{(t)}) \tag{29a}$$

$$\text{s.t.} \quad \mu_i^{(t)} \geq 0, \forall i, \tag{29b}$$

$$\lambda_1^{(t)}, \lambda_2^{(t)} \geq 0, \tag{29c}$$

where $\boldsymbol{\mu}^{(t)} = [\mu_i^{(t)}]_{i\in\mathcal{I}}$ and

$$\begin{aligned}
&\mathcal{L}_{\boldsymbol{\beta}}^*(\boldsymbol{\mu}^{(t)}, \lambda_1^{(t)}, \lambda_2^{(t)}) \\
&= -\sum_{i\in\boldsymbol{\alpha}^{(t)}} \alpha_i^{(t)} - \sum_{i\in\boldsymbol{\alpha}^{(t)}} \log(\lambda_1^{(t)} + \lambda_2^{(t)}\rho_i^{(t)} - \mu_i^{(t)}) \\
&\quad - \sum_{i\in\boldsymbol{\alpha}^{(t)}} \mu_i^{(t)}(\frac{1}{w_i^{(t)}} + \frac{\alpha_i^{(t)} r_i}{e_i^{(t)}}) \\
&\quad + \lambda_1^{(t)}(\sum_{i\in\boldsymbol{\alpha}^{(t)}} \frac{1}{w_i^{(t)}} + B) + \lambda_2^{(t)}(\sum_{i\in\boldsymbol{\alpha}^{(t)}} \frac{\rho_i^{(t)}}{w_i^{(t)}} + P).
\end{aligned} \tag{30}$$

The derivation of the Lagrangian dual function $\mathcal{L}_{\boldsymbol{\beta}}^*(\boldsymbol{\mu}^{(t)}, \lambda_1^{(t)}, \lambda_2^{(t)})$ is shown in Appendix B. The duality gap between (28) and (29) is zero, because the objective in (28a) is concave and constraints (28b)-(28d) are linear.

Then, by using the sub-gradient descent method, the Lagrangian multipliers can be updated by

$$\mu_i^{(t)} \leftarrow \mu_i^{(t)} - \gamma\left(\frac{1}{o_i^{(t)}} - \frac{1}{w_i^{(t)}} - \frac{\alpha_i^{(t)} r_i}{e_i^{(t)}}\right), \tag{31}$$

$$\lambda_1^{(t)} \leftarrow \lambda_1^{(t)} - \gamma\left(-\sum_{i\in\boldsymbol{\alpha}^{(t)}} \frac{1}{o_i^{(t)}} + B + \sum_{i\in\boldsymbol{\alpha}^{(t)}} \frac{1}{w_i^{(t)}}\right), \tag{32}$$

$$\lambda_2^{(t)} \leftarrow \lambda_2^{(t)} - \gamma\left(-\sum_{i\in\boldsymbol{\alpha}^{(t)}} \frac{\rho_i^{(t)}}{o_i^{(t)}} + P + \sum_{i\in\boldsymbol{\alpha}^{(t)}} \frac{\rho_i^{(t)}}{w_i^{(t)}}\right), \tag{33}$$

where γ is a learning rate and $o_i^{(t)} = \lambda_1^{(t)} + \lambda_2^{(t)}\rho_i^{(t)} - \mu_i^{(t)}$. After that, we have the optimal value of $\boldsymbol{\beta}^{(t)}$ can be obtained by

$$\beta_i^{*(t)} = \frac{1}{\lambda_1^{(t)} + \lambda_2^{(t)}\rho_i^{(t)} - \mu_i^{(t)}} - \frac{1}{w_i^{(t)}}, \tag{34}$$

where $\boldsymbol{\beta}^{*(t)} = [\beta_i^{*(t)}]_{i\in\mathcal{I}}$.

Algorithm 4: Resource Allocation

```
Input $\mathbf{q}^{(t)}$, $\boldsymbol{\alpha}^{(t)}$, $\boldsymbol{\rho}^{(t)}$.
Output $\boldsymbol{\beta}^{(t)}$.
$\boldsymbol{\mu}^{(t)}, \lambda_1^{(t)}, \lambda_2^{(t)} \leftarrow 0$,
$\hat{\mathcal{L}} \leftarrow 0$, $\hat{\mathcal{L}}_{\text{prev}} \leftarrow \infty$.
while $\|\hat{\mathcal{L}} - \hat{\mathcal{L}}_{\text{prev}}\| > \epsilon$ do
    Update $\boldsymbol{\mu}^{(t)}, \lambda_1^{(t)}, \lambda_2^{(t)}$ according to (31)-(33).
    $\hat{\mathcal{L}}_{\text{prev}} \leftarrow \hat{\mathcal{L}}$.
    $\hat{\mathcal{L}} \leftarrow \mathcal{L}_\beta^*(\boldsymbol{\mu}^{(t)}, \lambda_1^{(t)}, \lambda_2^{(t)})$ using (30).
end
$\boldsymbol{\beta}^{(t)} \leftarrow [\frac{1}{\lambda_1^{(t)}+\lambda_2^{(t)}\rho_i^{(t)}-\mu_i^{(t)}} - \frac{1}{w_i^{(t)}}]_{i\in\mathcal{I}}$.
```

4.2.3 Power Control

If $\boldsymbol{\alpha}^{(t)}$ and $\boldsymbol{\beta}^{(t)}$ are given, the problem (18) is redefined as

$$\max_{\boldsymbol{\rho}^{(t)}} \sum_{i\in\boldsymbol{\alpha}^{(t)}} \log\left(1 + \frac{R_i^{(t)}}{\sum_{k=0}^{t-1} R_i^{(k)}}\right) \tag{35a}$$

$$\text{s.t. } \rho_i^{(t)} \geq 0, \forall i, \tag{35b}$$

$$\sum_{i\in\boldsymbol{\alpha}^{(t)}} \rho_i^{(t)}\beta_i^{(t)} \leq P, \tag{35c}$$

$$R_i^{(t)} \geq \alpha_i^{(t)} r_i, \forall i. \tag{35d}$$

Since $\log(1+x) \approx x$ when $x \ll 1$, we relax the objective function as

$$\sum_{i\in\boldsymbol{\alpha}^{(t)}} \log\left(1 + \frac{R_i^{(t)}}{\sum_{k=0}^{t-1} R_i^{(k)}}\right) \approx \sum_{i\in\boldsymbol{\alpha}^{(t)}} \frac{R_i^{(t)}}{\sum_{k=0}^{t-1} R_i^{(k)}}. \tag{36}$$

We remark that the above approximation holds more accurately as t increases, because the cumulative received data $\sum_{k=0}^{t-1} R_i^{(k)}$ increases over time. Also, the constraint (35d) can be equivalently expressed with respect to PSD of a user as follows:

$$\rho_i^{(t)} \geq \zeta_i^{(t)}, \forall i, \tag{37}$$

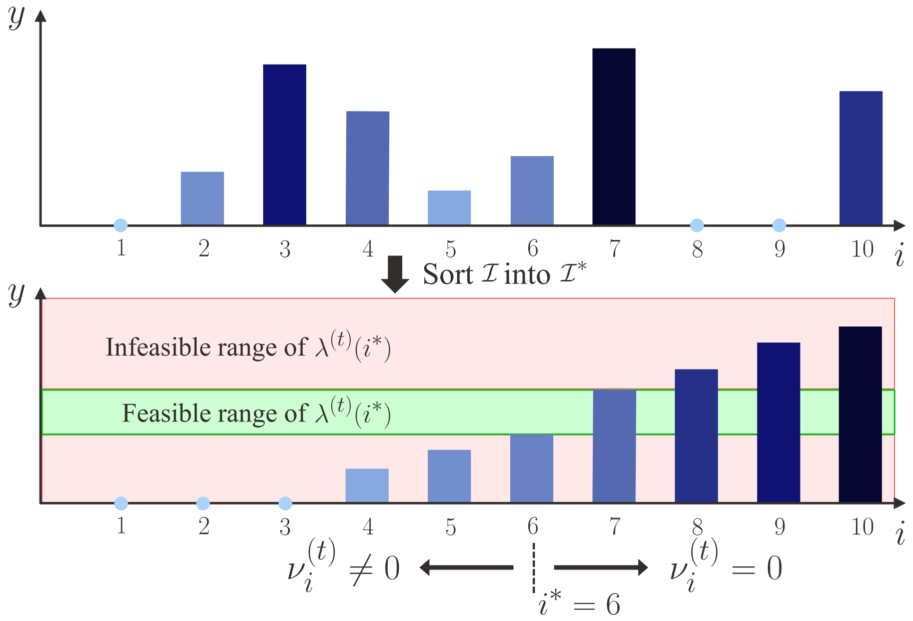


Figure 4: User index i vs. $y[i] = (\zeta_i^{(t)} + 1/\omega_i^{(t)})^{-1}\tau_i^{(t)}/\beta_i^{(t)}$ for given time slot t. We exemplify the case that the index 6 is chosen to be i^*. If $i^* = 6$ is a feasible candidate solution, it must satisfy $y[6] \leq \lambda^{(t)}[i^*] \leq y[7]$. Otherwise, $i^* = 6$ is not a feasible case.

where $\zeta_i^{(t)} = \frac{N_0}{10^{-\xi_i^{(t)}/10}}(2^{\alpha_i^{(t)} r_i/\beta_i^{(t)}} - 1)$. We can replace (35b) and (35d) by (37) because $\zeta_i^{(t)} \geq 0$.

By applying (36) and (37), the problem (35) can be reformulated as

$$\max_{\boldsymbol{\rho}^{(t)}} \sum_{i \in \boldsymbol{\alpha}^{(t)}} \frac{R_i^{(t)}}{\sum_{k=0}^{t-1} R_i^{(k)}} \tag{38a}$$

$$\text{s.t.} \sum_{i \in \boldsymbol{\alpha}^{(t)}} \rho_i^{(t)} \beta_i^{(t)} \leq P, \tag{38b}$$

$$\rho_i^{(t)} \geq \zeta_i^{(t)}, \forall i, \tag{38c}$$

The objective function and constraints of (38) are concave and linear, respectively. Therefore, we have zero duality gap for (38).

The Lagrangian $\mathcal{L}_{\boldsymbol{\rho}}$ of (38) is

$$\mathcal{L}_{\boldsymbol{\rho}}(\boldsymbol{\rho}^{(t)}, \boldsymbol{\nu}^{(t)}, \lambda^{(t)}) = \sum_{i \in \boldsymbol{\alpha}^{(t)}} \tau_i^{(t)} \log_2(1 + \omega_i^{(t)} \rho_i^{(t)}) \tag{39}$$

$$+ \sum_{i \in \boldsymbol{\alpha}^{(t)}} \nu_i^{(t)} (\rho_i^{(t)} - \zeta_i^{(t)}) - \lambda^{(t)} \Big(\sum_{i \in \boldsymbol{\alpha}^{(t)}} \rho_i^{(t)} \beta_i^{(t)} - P \Big),$$

where $\boldsymbol{\nu}^{(t)} = [\nu_i^{(t)}]_{i \in \mathcal{I}}$, $\tau_i^{(t)} = \alpha_i^{(t)} d_i^{(t)} / \sum_{k=0}^{t-1} R_i^{(k)}$ and $\omega_i^{(t)} = 10^{\xi_i^{(t)}/10}/n_0$.

We need to find a point at which derivative of $\mathcal{L}_\rho(\cdot)$ is zero:

$$\nabla_{\rho_i^{(t)}} \mathcal{L}_{\boldsymbol{\rho}}(\cdot) = \frac{\tau_i^{(t)} \omega_i^{(t)}}{1 + \omega_i^{(t)} \rho_i^{(t)}} + \nu_i^{(t)} - \lambda^{(t)} \beta_i^{(t)} = 0, \tag{40}$$

so we have

$$\rho_i^{(t)} = \frac{\tau_i^{(t)}}{\beta_i^{(t)} \lambda^{(t)} - \nu_i^{(t)}} - \frac{1}{\omega_i^{(t)}}. \tag{41}$$

The pair of $\boldsymbol{\rho}^{(t)}, \boldsymbol{\nu}^{(t)}$ and $\lambda^{(t)}$ should satisfy

$$\nu_i^{(t)}(\rho_i^{(t)} - \zeta_i^{(t)}) = 0 \tag{42}$$

to meet the complementary slackness of a KKT solution. If $\nu_i^{(t)} > 0$, then $\rho_i^{(t)} = \zeta_i^{(t)}$ and we have

$$\beta_i^{(t)}\lambda^{(t)} - (\zeta_i^{(t)} + \frac{1}{\omega_i^{(t)}})^{-1}\tau_i^{(t)} = \nu_i^{(t)} > 0, \tag{43}$$

and thus

$$\nu_i^{(t)} \neq 0 \to \lambda^{(t)} > (\zeta_i^{(t)} + \frac{1}{\omega_i^{(t)}})^{-1}\frac{\tau_i^{(t)}}{\beta_i^{(t)}}. \tag{44}$$

If $\nu_i^{(t)} = 0$, from (38c) and (41) we can derive

$$\nu_i^{(t)} = 0 \to \lambda^{(t)} \leq \underbrace{(\zeta_i^{(t)} + \frac{1}{\omega_i^{(t)}})^{-1}\frac{\tau_i^{(t)}}{\beta_i^{(t)}}}_{\text{(a)}}. \tag{45}$$

For fixed i, (a) in (45) is constant because $\mathbf{q}^{(t)}$, $\boldsymbol{\alpha}^{(t)}$ and $\boldsymbol{\beta}^{(t)}$ are assumed to be given.

If a user is not in its request period or a UAV-BS determine not to provide access to the user at the given time slot t, we have $\alpha_i^{(t)} d_i^{(t)} = 0$. Then (a) is trivially determined to be zero for these cases.

We can rearrange the user index set $\mathcal{I}$ into $\mathcal{I}^*$ by sorting $\mathcal{I}$ with (a). In other words, for all $i, j \in \mathcal{I}^*$, we have

$$i \leq j \to (\zeta_i^{(t)} + \frac{1}{\omega_i^{(t)}})^{-1}\frac{\tau_i^{(t)}}{\beta_i^{(t)}} \leq (\zeta_j^{(t)} + \frac{1}{\omega_j^{(t)}})^{-1}\frac{\tau_j^{(t)}}{\beta_j^{(t)}}. \tag{46}$$

Proposition 2 (Property of Lagrangian multipliers). *Let $\boldsymbol{\rho}^{(t)}$, $\boldsymbol{\nu}^{(t)}$ and $\lambda^{(t)}$ satisfy the KKT condition of* (38)*, and $i < j$ for $i, j \in \mathcal{I}^*$. Then the following two conditions hold:*

$$\nu_i^{(t)} \neq 0 \text{ if } \nu_j^{(t)} \neq 0, \tag{47}$$

$$\nu_j^{(t)} = 0 \text{ if } \nu_i^{(t)} = 0. \tag{48}$$

Proof. The proof is shown in Appendix C. ■

This proposition indicates that if we choose $i^* \in \mathcal{I}^*$ that satisfies

$$\nu_{i^*}^{(t)} \neq 0 \text{ and } \nu_{1+i^*}^{(t)} = 0, \tag{49}$$

then we have

$$\nu_i^{(t)} \neq 0 \text{ for } i \leq i^*\text{, and } \nu_i^{(t)} = 0 \text{ for } i > i^*. \tag{50}$$

For each $i^* \in \mathcal{I}^*$, we may find corresponding feasible solution $\boldsymbol{\rho}^{(t)}(i^*) = \left[\rho_i^{(t)}(i^*)\right]_{i\in\mathcal{I}^*}$ and $\lambda^{(t)}(i^*)$ if we assume that i^* satisfies (49) (Fig. 4). For feasible solution, we have

$$\sum_{i\in\boldsymbol{\alpha}^{(t)}} \beta_i^{(t)}\rho_i^{(t)}(i^*) = \sum_{i\in\mathcal{I}^*} \beta_i^{(t)}\rho_i^{(t)}(i^*) \tag{51}$$

$$= \sum_{i\leq i^*} \beta_i^{(t)}\rho_i^{(t)}(i^*) + \sum_{i>i^*} \beta_i^{(t)}\rho_i^{(t)}(i^*) \tag{52}$$

$$= \sum_{i\leq i^*} \beta_i^{(t)}\zeta_i^{(t)} + \sum_{i>i^*} \beta_i^{(t)}(\frac{\tau_i^{(t)}}{\beta_i^{(t)}\lambda^{(t)}(i^*)} - \frac{1}{\omega_i^{(t)}}) \leq P \tag{53}$$

from (38b). Then we can derive

$$\lambda^{(t)}(i^*) = \frac{\sum_{i>i^*} \tau_i^{(t)}}{P - \sum_{i\le i^*} \beta_i^{(t)}\zeta_i^{(t)} + \sum_{i>i^*} \beta_i^{(t)}/\omega_i^{(t)}} \tag{54}$$

as a candidate solution by assuming that equality holds on (54). Feasibility of the solution can be checked by computing

$$(\zeta_{i^*}^{(t)} + \frac{1}{\omega_{i^*}^{(t)}})^{-1}\frac{\tau_{i^*}^{(t)}}{\beta_{i^*}^{(t)}} \le \lambda^{(t)}(i^*) \le (\zeta_{1+i^*}^{(t)} + \frac{1}{\omega_{1+i^*}^{(t)}})^{-1}\frac{\tau_{1+i^*}^{(t)}}{\beta_{1+i^*}^{(t)}}. \tag{55}$$

Equation (55) can be obtained by combining (44) and (45) with (49).

If $\lambda^{(t)}(i^*)$ is a feasible solution, we have

$$\rho_i^{(t)}(i^*) = \begin{cases} \frac{\tau_i^{(t)}}{\beta_i^{(t)}\lambda^{(t)}(i^*)} - \frac{1}{\omega_i^{(t)}}, & \text{if } i > i^*, \\ \zeta_i^{(t)}, & \text{if } i \le i^*, \end{cases} \tag{56}$$

for $i \in \mathcal{I}^*$. $\rho_i^{(t)}(i^*)$ can be derived by combining (41) and (54).

We define the set of candidate solutions $\mathcal{P}$ as

$$\mathcal{P}^{(t)} = \{\boldsymbol{\rho}^{(t)}(i^*) | i^* \in \mathcal{I}^*, \lambda^{(t)}(i^*) \text{ satisfies (55)}\}. \tag{57}$$

Then, we can find the optimal value of (38) as by 1-dimensional search of the objective function (36) for the set $\mathcal{P}^{(t)}$.

Algorithm 5: Power Control

```
Input $\mathbf{q}^{(t)}$, $\boldsymbol{\alpha}^{(t)}$, $\boldsymbol{\beta}^{(t)}$.
Output $\boldsymbol{\rho}^{(t)}$.
$\mathcal{I}^* \leftarrow$ Sort $\mathcal{I}$ according to $(\zeta_i^{(t)} + \frac{1}{\omega_i^{(t)}})^{-1}\frac{\tau_i^{(t)}}{\beta_i^{(t)}}$.
for $i^*$ in $\mathcal{I}^*$ do
    $\lambda^{(t)}(i^*) \leftarrow \frac{\sum_{i>i^*} \tau_i^{(t)}}{P-\sum_{i\le i^*} \beta_i^{(t)}\zeta_i^{(t)}+\sum_{i>i^*} \beta_i^{(t)}/\omega_i^{(t)}}$.
    Check feasibility of $\lambda^{(t)}(i^*)$ using (55).
    if $\lambda^{(t)}(i^*)$ is feasible then
        Compute $\rho_i^{(t)}(i^*), \forall i \in \mathcal{I}^*$ according to (56).
        $\boldsymbol{\rho}^{(t)}(i^*) \leftarrow \left[\rho_i^{(t)}(i^*)\right]_{i\in\mathcal{I}^*}$.
        $\mathcal{P}^{(t)} \leftarrow \mathcal{P}^{(t)} \cup \{\boldsymbol{\rho}^{(t)}(i^*)\}$.
    end
end
$\boldsymbol{\rho}^{(t)} \leftarrow$ Find $\boldsymbol{\rho} \in \mathcal{P}^{(t)}$ maximizing the objective (35a).
```

4.2.4 Overall Node Evaluation Algorithm

The node evaluation function $f(\mathbf{q^{(t)}})$ in (18) can be computed by iteratively solving (28) and (35) (Alg. 2). The algorithm first initializes the power spectral density of users and begins iteration of Algs. 4, 5. In each iteration, the algorithm first optimizes $\boldsymbol{\alpha}^{(t)}$ and $\boldsymbol{\beta}^{(t)}$ with given $\boldsymbol{\rho}^{(t)}$ and $\mathbf{q}^{(t)}$ by solving the problem (28) using gradient descent. Then, the algorithm updates $\boldsymbol{\rho}^{(t)}$ with given $\boldsymbol{\alpha}^{(t)}$, $\boldsymbol{\beta}^{(t)}$ and $\mathbf{q}^{(t)}$ by finding the maximal $\boldsymbol{\rho}^{(t)}$ among the KKT solutions of the problem (35).

4.2.5 Computational Complexity

We can obtain the optimal $\boldsymbol{\alpha}^{(t)}$, $\boldsymbol{\beta}^{(t)}$, $\boldsymbol{\rho}^{(t)}$ and $\mathbf{q}^{(t)}$ for all $t \in \mathcal{T}$ by repeating the iterations until the objective converges. We determined that the Alg. 2 converges, if the difference between the objective functions for the previous iteration and the current iteration is less than ϵ. The computational complexity of Alg. 1 is $\mathcal{O}(\frac{I^2 m^n}{\epsilon^2})$, where m is the number of reachable grid points. This complexity is computed by multiplying the complexity of the DFS (Alg. 1) and the node evaluation (Alg. 2). The complexity of the DFS is well-known as $\mathcal{O}(m^n)$ and the complexity of Alg. 2 is $\mathcal{O}(\frac{I^2}{\epsilon^2})$.

5 Simulation Results

In various performance metrics, we compare the proposed scheme with other schemes that maximize the PF of users by optimizing the trajectory. Simulation parameters are chosen from existing works(7,25) and 3rd generation partnership project (3GPP) specifications(26), which are listed in Table 2.

Table 2: Parameter Configurations

Parameter	Value
Map width w (m)	600
Minimum altitude h_{min} (m)	50
Maximum altitude h_{max} (m)	200
Number of users I	10 to 80
Number of time slots T	20
Unit length of the time slot ΔT (s)	3
Vehicle velocity v (m/s)	15
Carrier frequency (GHz)	2
Bandwidth B (MHz)	{2, 5, 10}
Transmission power P (dBm)	23
Noise spectral density (dBm/Hz)	-173.8
Data rate constraint $r_i, \forall i \in \mathcal{I}$ (Mbps)	0 to 10
Environmental parameter a	9.64
Environmental parameter b	0.06
Average excessive LoS pathloss η_{LoS}	1
Average excessive NLoS pathloss η_{NLoS}	40

In our experiments, the length of the request period $T_i,\ \forall i \in \mathcal{I}$ and the initial request time $s_i,\ \forall i \in \mathcal{I}$ follow discrete uniform distribution uniform$(4, 8)$ and uniform$(0, T)$, respectively.

We have developed a trajectory-planning simulator that implements the proposed method[4] and several comparison schemes. The simulations are implemented on Python 3.9 and AMD Ryzen™ 9 5950X processor.

We evaluate our scheme for three kinds of lookahead steps (which corresponds to the depth of each tree): $n = 1,\ 3,\ 5$. The following comparison schemes are implemented in our simulations:

- **OFDMA (inspired from the other works(12,13))**: Although UAV-BSs are likely to have LoS links with users, the frequency selectivity still needs to be considered. Motivated from(12,13), we design a frequency selective OFDMA problem that jointly optimizes trajectory, UA, RA, and PC[5]. The differences from our scheme can be summarized as follows:
  - *The objective function*: the objective function is relaxed into weighted sum-rate formulation by using the approximation technique introduced in (36).
  - *Channel model*: Frequency selective channel model is additionally considered by adopting the Rician small scale fading. We adopt the value of the Rician K factor as $K = 12$ (dB) that is statistically determined from the extensive field measurements(25).

[4] Our code is available at: https://github.com/hslyu/dbspf

[5] Energy consumption model and relaying in(13) are not considered in our implementation.

- *QoS constraint*: While our scheme adopts the data rate threshold as a QoS constraint for end-to-end service, these works use the SNR threshold as a QoS constraint. For the fair comparison, the SNR thresholds γ_i^{OFDMA}, $i \in \mathcal{I}$ are chosen to be $r_i = B\log_2(1+\gamma_i^{\text{OFDMA}})$.
- *Trajectory-planning*: The trajectory is determined by sequentially optimizing the position of the UAV-BS for each time step. This trajectory-planning can be considered similar to 1-step lookahead problem $\mathbf{P}^{(t)}$ in (15).

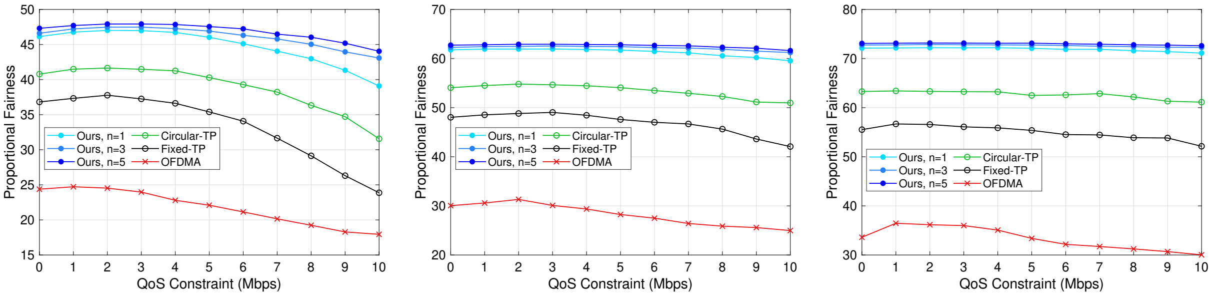
- **Circular-TP**: While UA, RA, and PC are optimized by Alg. 2, the UAV-BS position $\mathbf{q}^{(t)}$ is parametrized as

$$\mathbf{q}^{(t)} = \begin{bmatrix} 100\cos(\theta_t)+300 \\ 100\sin(\theta_t)+300 \\ 200 \end{bmatrix}, \tag{58}$$

where $\theta_t = \theta + \frac{v\Delta T}{r}t$ for randomly selected $\theta \in [0, 2\pi]$. Then, the variable θ determines the initial position, and the UAV-BS rotates the orbit with angular velocity $\frac{v\Delta T}{r}$.

- **Fixed-TP**: For all $t \in \mathcal{T}$, the UAV-BS position is fixed at $\mathbf{q}^{(t)} = [300, 300, 200]^T$. Same as the Circular-TP scheme, this scheme determines UA, RA, and PC by the Alg. 2 with the given position $\mathbf{q}^{(t)}$.

5.1 Proportional Fairness for Various QoS Constraints



(a) Bandwidth $B = 2$ (MHz) (b) Bandwidth $B = 5$ (MHz) (c) Bandwidth $B = 10$ (MHz)

Figure 5: Proportional fairness for 20 users with various QoS constraints.

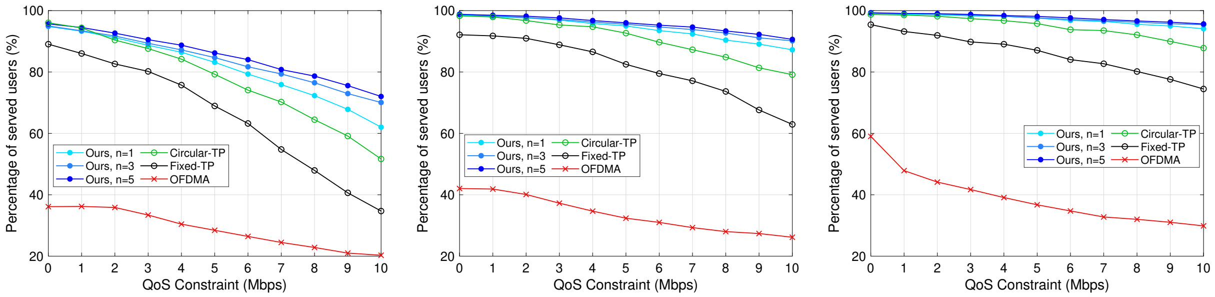
Fig. 5 shows the proportional fairness for 20 users under various QoS constraints. Because the OFDMA scheme adopts an approximated PF formulation, we evaluate the PF of the UAV-BS for all schemes by adding the logarithm of total received data for all served users, as formulated at the objective function (13a) of Problem $\mathbf{P1}$.

Fig. 5a depicts the PF of the UAV-BS at 2 MHz bandwidth. The figure shows that the PF of the proposed scheme with tree depth $n = 5$ achieves a 95% higher PF value than the OFDMA scheme even at the worst case. Because the objective function in the OFDMA scheme is reformulated into a weighted sum-rate maximization problem that is linear with respect to the users' sum-rate, providing all resources to one user with the highest efficiency might be an optimal strategy. However, this biased resource management decreases the fairness of the network in the time-critical mobile scenario, because the request period of other users might be expired while a few users are selectively serviced.

When the QoS constraints are configured to be about 2, the PF achieves around 2% higher value than unconstrained case ($r_i = 0,\ \forall i \in \mathcal{I}$) regardless of the schemes. This peak occurs because the QoS constraints enable the UAV-BSs to effectively provide service to users with high SNR links. Even at the stringent QoS constraints, our scheme shows smaller decline in the PF value than other schemes. Compared with the peak PF value, the PF values at QoS constraints of 10 Mbps decrease 25% for the circular-TP; 37% for the fixed-TP; 27% for the OFDMA scheme; 17%, 9%, and 8% for our scheme with $n = 1,\ 3,\ 5$, respectively (Fig. 5a).

Figs. 5b and 5c depict the PF value when the UAV-BS has 5 and 10 MHz bandwidth, respectively. Our schemes show the consistent PF values regardless of the QoS constraints, while the PF of the circular-TP and fixed-TP schemes slightly decreases. This implies that our schemes are benefited from the high SNR channels obtained from the DFS-based trajectory-planning, because the circular-TP and fixed-TP schemes share the same resource management scheme as our scheme.

5.2 Percentage of Served Users for Various QoS Constraints



(a) Bandwidth $B = 2$ (MHz) (b) Bandwidth $B = 5$ (MHz) (c) Bandwidth $B = 10$ (MHz)

Figure 6: Percentage of served users with various QoS constraints. The total number of users is configured to be 20.

Fig. 6 illustrates the percentage of served users for different QoS constraints. The percentage is computed by dividing the number of users who have been serviced at least once during the service timeline into the total number of users.

In Fig. 6a, the percentage decreases as the QoS constraints becomes stringent. Our scheme with the lookahead variable $n = 5$ serves 51 percent point (%p) more users than the OFDMA scheme. This gap is mainly derived by difference in the objective function formulation. As mentioned in the previous section 5.1, serving a few users is highly optimal UA scheme in the weight sum-rate formulation. However, the objective of the proposed problem (18) is reformulated as a summation of logarithms. Then, focusing all resources on one users might not be optimal because the gradient of the logarithm decreases as the sum-rate increases. Therefore, multiple users can be readily associated with the UAV-BS at a single time slot in the proposed scheme.

For the circular-TP and fixed-TP schemes, the gap with our scheme increases as users' QoS constraints increase. Our scheme shows a small gap at unconstrained case, but the gap increases up to 20%p for the circular-TP and 37%p for the fixed-TP as QoS constraints are configured to be $r_i = 10,\ \forall i \in \mathcal{I}$.

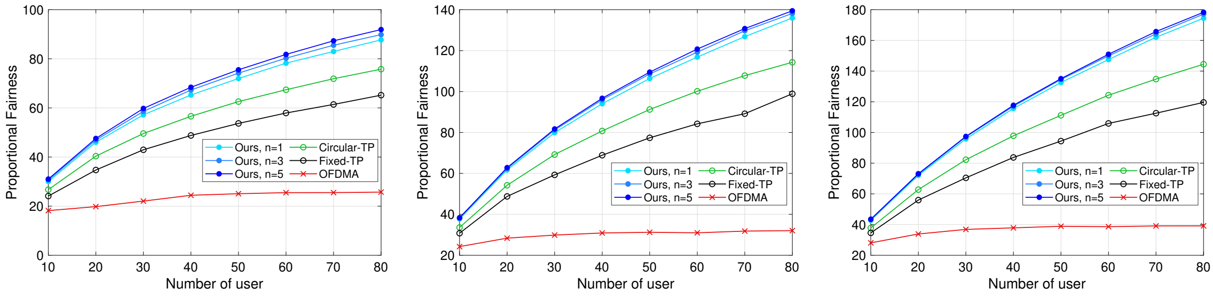
As depicted in Figs. 6b and 6c, our schemes converge to serving 100% users regardless of the lookahead variable n when the available bandwidth increases. However, the OFDMA scheme shows little enhancements due to the winner-take-all tendency in the user association.

5.3 Proportional Fairness for Various Number of Users

Fig. 7 presents the PF for 10 to 80 users with the 2, 5 and 10 MHz bandwidth. In the same condition as Fig. 7, we demonstrate the percentage of served users in Fig. 8. We set the QoS constraints as $r_i = 5$ for all $i \in \mathcal{I}$.

The PF of our schemes increases by 196% as the number of users increases 10 to 80 (Fig. 7a). Also, the circular-TP and fixed-TP schemes have similar increase, but they show 20% and 40% lower PF than the proposed scheme, respectively. This implies that our schemes take advantages of the high SNR channels obtained from the trajectory-planning scheme (Alg. 1), because the circular-TP and fixed-TP share the same UA, RA, and PC scheme with our schemes. The PF of the OFDMA scheme shows 41% increase that is mainly caused by the channel enhancement, not by the increase in the number of served users. Detailed analysis for the OFDMA scheme is provided in the next section.

As illustrated in Fig. 7b, the PF for 5 MHz available bandwidth shows similar trend with the PF for 2 MHz bandwidth (Fig. 7a). However, the PF gap between our schemes and the OFDMA scheme increases as the available bandwidth increases. For 5 MHz bandwidth, our scheme with $n = 5$ shows a 65% higher PF value

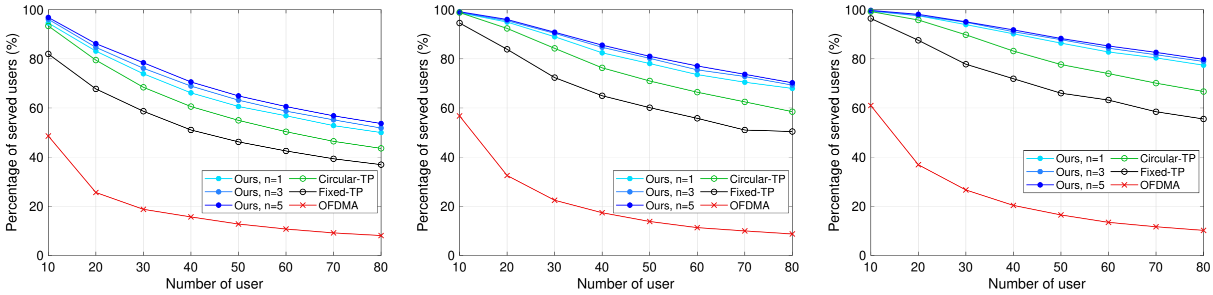


(a) Bandwidth $B = 2$ (MHz) (b) Bandwidth $B = 5$ (MHz) (c) Bandwidth $B = 10$ (MHz)

Figure 7: Proportional fairness with a various number of users. The rate constraints $r_i, \ \forall i \in \mathcal{I}$ are configured as $r_i = 5, \ \forall i \in \mathcal{I}$.

at 10 users; 241% higher PF value at 80 users compared with the OFDMA scheme. Similarly, when the UAV-BS has 10 MHz bandwidth (Fig. 7c), the PF gap between our scheme with $n = 5$ and the OFDMA scheme increases 52% for 10 users to 345% for 80 users.

5.4 Percentage of Served Users for Various Number of Users



(a) Bandwidth $B = 2$ (MHz) (b) Bandwidth $B = 5$ (MHz) (c) Bandwidth $B = 10$ (MHz)

Figure 8: Percentage of served users with a various number of users.

Fig. 8 depicts the percentage of served users for different number of users with 2, 5, and 10 MHz bandwidth. The QoS constraints are configured to be $r_i = 5, \ \forall i \in \mathcal{I}$.

In Fig. 8a, the average number of served users for the proposed scheme with $n = 5$ increases from 9.68 to 42.93 as the total number of users increases from 10 to 80. Meanwhile, the average number of served users for the OFDMA scheme increases from 4.85 to 6.43 at 2 MHz bandwidth. This implies that the increase in Fig. 7a of the OFDMA scheme is mainly caused by the channel enhancement from the high user-density, not by the increase in the number of served users.

In Fig. 8b, we note that the overall percentage of served users increases when the UAV-BS utilizes 5 MHz bandwidth. Our schemes with achieve about 17%p higher percentage for 5 MHz bandwidth than for 2 MHz, but the OFDMA scheme shows almost no difference regardless of the amount of the available bandwidth. Fig. 8c shows that the percentage of served users gradually approaches to 100% with having similar tendency with Figs. 8a and 8b.

5.5 Sum-Rate For Various QoS Constraints

Fig. 9 represents the average sum-rate of the UAV-BS per time slot when the UAV-BS utilizes 2 MHz bandwidth. The OFDMA scheme has the highest sum-rate (22 Mbps), because the OFDMA scheme tends to provide service to a few users with high SNR channels, as described in Section. 5.1.

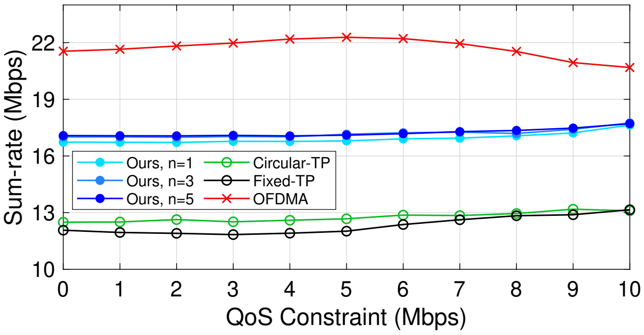


Figure 9: Sum-rate of the UAV-BS at the bandwidth $B = 2$ MHz.

Our schemes show around 4 Mbps higher sum-rate than the circular-TP and fixed-TP, because our schemes reduce the pathloss of users by using the DFS trajectory-planning algorithm.

We note that the sum-rate of our scheme maintain a certain level regardless of QoS constraints. However, as illustrated in Fig. 5a, our schemes decrease the number of served users to compensate for the increase in QoS constraints. The circular-TP and fixed-TP also have a consistent sum-rate, because this tendency is derived by the resource management scheme (Alg. 2).

5.6 Convergence and Computational Complexity Analysis

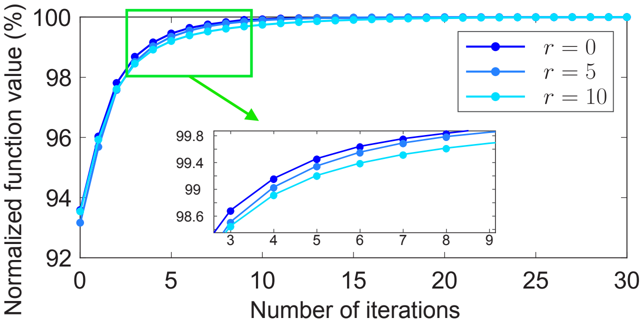


Figure 10: The normalized output value of the Alg. 2. The graph is normalized by the converged function value. The QoS constraints is configured as $r_i = r, \forall i \in \mathcal{I}$.

Table 3: Regularized PF and complexity of three UA Algorithms

Algorithm	Regularized PF (%)*			Complexity (Flops)
	10 users	20 users	40 users	
Exhaustive search	100%	100%	100%	$\mathcal{O}(2^I)$
Proposed method	**100%**	**99.91%**	**99.80%**	$\mathcal{O}(I^2)$
Max SINR	75.29%	66.76%	62.44%	$\mathcal{O}(I)$

*The PF is regularized by the PF of the exhaustive search method.

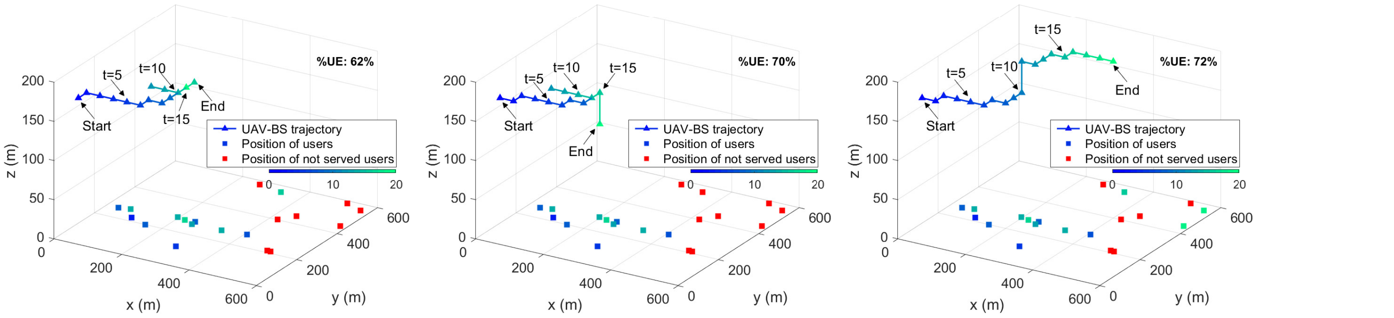
Fig. 10 depicts regularized PF for each iteration in Alg. 2. Zero iteration implies that the UA and RA variables are only optimized by the initial UA and RA scheme (Alg. 3) without implementing the subsequent RA (Alg. 4) and PC (Alg. 5) schemes. Even at the zero iteration, Alg. 3 achieves 93% of the regularized PF value. Then, the PF accomplishes 99% and 99.9% after 5 and 10 iterations, respectively.

Table 3 shows the regularized PF of the three UA methods: exhaustive search, proposed algorithm (Alg. 3) and max SINR method. Max SINR method means that the UAV-BS only communicates with the user whose SINR is greater than any other users. The number of time slots and QoS threshold are configured to be 20 and $r_i = 5, \forall i \in \mathcal{I}$, respectively. Our solution tightly accomplishes PF of the exhaustive search method, while decreasing computational complexity from $\mathcal{O}(2^n)$ to $\mathcal{O}(n^2)$.

5.7 The Effect of Lookahead Parameter

For all performance metrics, the proposed model with the higher lookahead value n always outperforms models with lower n value. In Fig. 5a, our scheme with $n = 5$ has a 12% higher PF than ours with $n = 1$. This implies that a proper lookahead variable needs to be considered to deploy the UAV-BS at the appropriate position for serving users newly starting to request communication.

Fig. 11 demonstrates the UAV-BS's trajectories for various lookahead variable n, when the UAV-BS has 2 MHz bandwidth. In addition, we assume that 20 users exist in the service area with QoS constraints $r_i = 10,\ \forall i \in \mathcal{I}$. As n increases, the number of served users accordingly increases. This is because that the UAV-BS can be closely deployed in advance by considering the future service requirement of users. Because the difference in trajectories accumulates along the service timeline, there might be a big gap in the number of served users between $n = 1$ and $n = 5$ as the service time of the UAV-BS increases.



(a) Tree depth $n = 1$ (b) Tree depth $n = 3$ (c) Tree depth $n = 5$

Figure 11: Visualized trajectories for various tree depths. The colors of the UAV-BS trajectory represent the index of time slots, and the colors in the position of users represent the time slot in which the user receives service for the first time. %UE represents the average percentage of the served users for 150 experiments.

6 Conclusion

This paper considers a DFS-assisted trajectory-planning technique for the single UAV-BS in time-critical networks. The proposed scheme aims to maximize proportional fairness by jointly considering the trajectory, UA, RA, and PC variables. We propose a temporal decoupling method, then reformulate the decoupled problems into tree search problems that contain a part of the trajectory. We find the optimal sub-trajectory of each tree search problem by using DFS algorithm, then find a lower-bound of the original problem by concatenating the sub-trajectories. To measure the value of the sub-trajectory, we suggest an algorithm that finds near-optimal UA, RA, and PC variables for the given sub-trajectory. Our UA, RA, and PC algorithm achieves 99.8% performance of the exhaustive search, while significantly decreasing computational complexity. Moreover, the numerical results show that the proposed scheme outperforms two naive-trajectory schemes and state-of-the-art OFDMA scheme.

Appendix A
Proof of Proposition 1

The logarithm of the sum-rate of i-th user is reformulated as

$$\log R_i = \log \sum_{t\in\mathcal{T}} R_i^{(t)} \tag{59}$$

$$= \log \prod_{t=1}^{T} \left(\frac{\sum_{k=0}^{t} R_i^{(k)}}{\sum_{k=0}^{t-1} R_i^{(k)}} \right) \cdot R_i^{(0)} \tag{60}$$

$$= \sum_{t=1}^{T} \log \left(1 + \frac{R_i^{(t)}}{\sum_{k=0}^{t-1} R_i^{(k)}} \right) + \log R_i^{(0)}, \tag{61}$$

where the auxiliary constant $R_i^{(0)}$ is identical for all $i \in \mathcal{I}$. In the above equation, we adopt the auxiliary constant $R_i^{(0)}$ to consistently decouple the problem for all time slots[6]. Then, the objective in (13a) is equivalent with

$$\sum_{i\in\boldsymbol{\alpha}} \log R_i = \sum_{t=1}^{T} \sum_{i\in\boldsymbol{\alpha}^{(t)}} \log \left(1 + \frac{R_i^{(t)}}{\sum_{k=0}^{t-1} R_i^{(k)}} \right) + c, \tag{62}$$

where $c = \sum_{i\in\boldsymbol{\alpha}} \log R_i^{(0)}$. Therefore, the following inequality holds:

$$\max_{\mathbf{q},\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\rho}} \sum_{i\in\boldsymbol{\alpha}} \log R_i \tag{63}$$

$$= \max_{\mathbf{q},\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\rho}} \sum_{t=1}^{T} \sum_{i\in\boldsymbol{\alpha}^{(t)}} \log \left(1 + \frac{R_i^{(t)}}{\sum_{k=0}^{t-1} R_i^{(k)}} \right) + c \tag{64}$$

$$\geq \sum_{t=1}^{T} \max_{\mathbf{q}^{(t)},\boldsymbol{\alpha}^{(t)},\boldsymbol{\beta}^{(t)},\boldsymbol{\rho}^{(t)}} \sum_{i\in\boldsymbol{\alpha}^{(t)}} \log \left(1 + \frac{R_i^{(t)}}{\sum_{k=0}^{t-1} R_i^{(k)}} \right). \tag{65}$$

The inequality (65) indicates that the addition of the objectives of Problem $\mathbf{P}^{(t)}, t \in \mathcal{T}$ is the lower bound of Problem $\mathbf{P1}$.

Appendix B
Derivation of the Lagrangian dual function

The Lagrangian dual function (30) is derived from the Lagrangian function of the problem (28), where the Lagrangian function is denoted as

$$\begin{aligned} &\mathcal{L}_{\boldsymbol{\beta}}(\boldsymbol{\beta}^{(t)}, \boldsymbol{\mu}^{(t)}, \lambda_1^{(t)}, \lambda_2^{(t)}) \\ &= \sum_{i\in\boldsymbol{\alpha}^{(t)}} \log(1 + w_i^{(t)} \beta_i^{(t)}) + \sum_{i\in\boldsymbol{\alpha}^{(t)}} \mu_i^{(t)} (\beta_i^{(t)} - \frac{\alpha_i^{(t)} r_i}{e_i^{(t)}}) \\ &\quad - \lambda_1^{(t)} (\sum_{i\in\boldsymbol{\alpha}^{(t)}} \beta_i^{(t)} - B) - \lambda_2^{(t)} (\sum_{i\in\boldsymbol{\alpha}^{(t)}} \rho_i^{(t)} \beta_i^{(t)} - P), \end{aligned} \tag{66}$$

where $w_i^{(t)} = \alpha_i^{(t)} d_i^{(t)} e_i^{(t)} / \sum_{k=0}^{t-1} R_i^{(k)}$.

[6] Otherwise, we have a different a different term at $t = 1$

The partial derivative of $\mathcal{L}(\cdot)$ with respect to $\beta_i^{(t)}$ is 0 at optimal resource allocation vector $\boldsymbol{\beta}^{*(t)}$, so we have

$$\nabla_{\beta_i^{(t)}}\mathcal{L}_{\boldsymbol{\beta}}(\cdot) = \frac{w_i^{(t)}}{1+w_i^{(t)}\beta_i^{(t)}} + \mu_i^{(t)} - \lambda_1^{(t)} - \lambda_2^{(t)}\rho_i^{(t)} \tag{67}$$

$$= 0, \tag{68}$$

Then, the optimal $\beta_i^{*(t)}$ is

$$\beta_i^{*(t)} = \frac{1}{\lambda_1^{(t)} + \lambda_2^{(t)}\rho_i^{(t)} - \mu_i^{(t)}} - \frac{1}{w_i^{(t)}}, \tag{69}$$

and $\boldsymbol{\beta}^{*(t)} = [\beta_i^{*(t)}]_{i\in\mathcal{I}}$. Therefore, we have

$$\mathcal{L}_{\boldsymbol{\beta}}^{*}(\boldsymbol{\mu}^{(t)}, \lambda_1^{(t)}, \lambda_2^{(t)}) = \mathcal{L}_{\beta}(\boldsymbol{\beta}^{*(t)}, \boldsymbol{\mu}^{(t)}, \lambda_1^{(t)}, \lambda_2^{(t)}). \tag{70}$$

Appendix C
Proof of Proposition 2

For (47), if $\nu_i^{(t)} = 0$, we have

$$\nu_j^{(t)} \neq 0 \rightarrow (\zeta_j^{(t)} + \frac{1}{\omega_j^{(t)}})^{-1}\frac{\tau_j^{(t)}}{\beta_j^{(t)}} \leq \lambda^{(t)}, \tag{71}$$

and

$$\nu_i^{(t)} = 0 \rightarrow \lambda^{(t)} \leq (\zeta_i^{(t)} + \frac{1}{\omega_i^{(t)}})^{-1}\frac{\tau_i^{(t)}}{\beta_i^{(t)}}, \tag{72}$$

which contradict to (46).

In similar way, if $\nu_j^{(t)} \neq 0$ for (48), we have

$$(\zeta_j^{(t)} + \frac{1}{\omega_j^{(t)}})^{-1}\frac{\tau_j^{(t)}}{\beta_j^{(t)}} \leq \lambda^{(t)} \leq (\zeta_i^{(t)} + \frac{1}{\omega_i^{(t)}})^{-1}\frac{\tau_i^{(t)}}{\beta_i^{(t)}}, \tag{73}$$

which also contradicts to (46).

References

[1] Cisco, “Cisco annual internet report (2018–2023),” Cisco, San Jose, CA, USA, Tech. Rep. C11-741490-01, Mar. 2020.

[2] H. Sarbazi-Azad and A. Y. Zomaya, *Analytical Model of Time-Critical Wireless Sensor Network: Theory and Evaluation*. Wiley-IEEE Press, 2014, pp. 183–202.

[3] M. Samir, S. Sharafeddine, C. M. Assi, T. M. Nguyen, and A. Ghrayeb, “UAV trajectory planning for data collection from time-constrained IoT devices,” *IEEE Trans. Wireless Commun.*, vol. 19, no. 1, pp. 34–46, 2020.

[4] A. Al-Hilo, M. Samir, M. Elhattab, C. Assi, and S. Sharafeddine, “RIS-assisted UAV for timely data collection in IoT networks,” *arXiv e-prints*, 2021. [Online]. Available: https://ui.adsabs.harvard.edu/abs/2021arXiv210317162A

[5] D. Ebrahimi, S. Sharafeddine, P.-H. Ho, and C. Assi, “UAV-aided projection-based compressive data gathering in wireless sensor networks,” *IEEE Internet Things J.*, vol. 6, no. 2, pp. 1893–1905, 2019.

[6] C. Zhan, Y. Zeng, and R. Zhang, “Trajectory design for distributed estimation in UAV-enabled wireless sensor network,” *IEEE Trans. Veh. Technol.*, vol. 67, no. 10, pp. 10 155–10 159, 2018.

[7] A. Al-Hourani, S. Kandeepan, and A. Jamalipour, “Modeling air-to-ground path loss for low altitude platforms in urban environments,” in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2014, pp. 2898–2904.

[8] M. Mozaffari, W. Saad, M. Bennis, Y.-H. Nam, and M. Debbah, “A tutorial on UAVs for wireless networks: Applications, challenges, and open problems,” *IEEE Commun. Surveys Tuts.*, vol. 21, no. 3, pp. 2334–2360, 2019.

[9] Y. Zeng, R. Zhang, and T. J. Lim, “Wireless communications with unmanned aerial vehicles: opportunities and challenges,” *IEEE Commun. Mag.*, vol. 54, no. 5, pp. 36–42, 2016.

[10] D. Palomar and M. Chiang, “A tutorial on decomposition methods for network utility maximization,” *IEEE J. Select. Areas Commun.*, vol. 24, no. 8, pp. 1439–1451, 2006.

[11] S. Zhang and N. Ansari, “3D drone base station placement and resource allocation with fso-based backhaul in hotspots,” *IEEE Trans. Veh. Technol.*, vol. 69, no. 3, pp. 3322–3329, 2020.

[12] S. Zeng, H. Zhang, and L. Song, “Trajectory optimization and resource allocation for multi-user OFDMA UAV relay networks,” in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2019, pp. 1–6.

[13] S. Zeng, H. Zhang, B. Di, and L. Song, “Trajectory optimization and resource allocation for OFDMA UAV relay networks,” *IEEE Trans. Wireless Commun.*, vol. 20, no. 10, pp. 6634–6647, 2021.

[14] C. Shen, T. Chang, J. Gong, Y. Zeng, and R. Zhang, “Multi-UAV interference coordination via joint trajectory and power control,” *IEEE Trans. Signal Processing*, vol. 68, pp. 843–858, 2020.

[15] O. Abbasi, H. Yanikomeroglu, A. Ebrahimi, and N. M. Yamchi, “Trajectory design and power allocation for drone-assisted NR-V2X network with dynamic NOMA/OMA,” *IEEE Trans. Wireless Commun.*, vol. 19, no. 11, pp. 7153–7168, 2020.

[16] H. Hu, K. Xiong, G. Qu, Q. Ni, P. Fan, and K. B. Letaief, “AoI-minimal trajectory planning and data collection in UAV-assisted wireless powered IoT networks,” *IEEE Internet Things J.*, vol. 8, no. 2, pp. 1211–1223, 2021.

[17] P. Tong, J. Liu, X. Wang, B. Bai, and H. Dai, “Deep reinforcement learning for efficient data collection in UAV-aided internet of things,” in *Proc. IEEE Int. Conf. on Commun. Workshops (ICC Workshops)*, June 2020, pp. 1–6.

[18] C. You and R. Zhang, “3D trajectory optimization in rician fading for UAV-enabled data harvesting,” *IEEE Trans. Wireless Commun.*, vol. 18, no. 6, pp. 3192–3207, 2019.

[19] Y. Zhu and S. Wang, “Efficient aerial data collection with cooperative trajectory planning for large-scale wireless sensor networks,” *IEEE Trans. Commun.*, vol. 70, no. 1, pp. 433–444, 2022.

[20] B. Zhu, E. Bedeer, H. H. Nguyen, R. Barton, and J. Henry, “UAV trajectory planning in wireless sensor networks for energy consumption minimization by deep reinforcement learning,” *IEEE Trans. Veh. Technol.*, vol. 70, no. 9, pp. 9540–9554, 2021.

[21] X. Liu, Y. Liu, Y. Chen, and L. Hanzo, “Trajectory design and power control for multi-UAV assisted wireless networks: A machine learning approach,” *IEEE Trans. Veh. Technol.*, vol. 68, no. 8, pp. 7957–7969, 2019.

[22] H. Bayerlein, P. D. Kerret, and D. Gesbert, “Trajectory optimization for autonomous flying base station via reinforcement learning,” in *’18 Proc. IEEE 19th Int. Workshop on Signal Processing Advances in Wireless Commun. (SPAWC)*, Aug. 2018, pp. 1–5.

[23] A. Al-Hourani, S. Kandeepan, and S. Lardner, “Optimal LAP altitude for maximum coverage,” *IEEE Commun. Lett.*, vol. 3, no. 6, pp. 569–572, 2014.

[24] S. Burer and A. N. Letchford, “Non-convex mixed-integer nonlinear programming: A survey,” *Surveys in Operations Research and Management Science*, vol. 17, no. 2, pp. 97–106, 2012. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1876735412000037

[25] D. W. Matolak and R. Sun, “Air–ground channel characterization for unmanned aircraft systems—part III: The suburban and near-urban environments,” *IEEE Trans. Veh. Technol.*, vol. 66, no. 8, pp. 6607–6618, 2017.

[26] 3GPP, “Typical traffic characteristics of media services on 3GPP networks,” 3rd Generation Partnership Project (3GPP), Technical Specification (TS) 26.925, 04 2022, version 17.1.0. [Online]. Available: https://portal.etsi.org/webapp/workprogram/Report$_{W}orkItem.asp?WKI_{I}D = 65053$