Joint Optimization of Trajectory, Offloading, Caching, and Migration for UAV-Assisted MEC

Mingxiong Zhao, *Member, IEEE*, Rongqian Zhang, Zhenli He, *Senior Member, IEEE*, and Keqin Li, *Fellow, IEEE*

***Abstract*—UAV-assisted MEC revolutionizes edge computing by deploying UAVs for real-time data processing in areas lacking infrastructure, supporting a wide range of applications from emergency responses to smart cities. Unlike edge servers, UAVs face substantial computational constraints, necessitating a comprehensive strategy that integrates UAV trajectory with task offloading, caching, and migration. Existing studies often overlook the synergy among these strategies, impacting their overall effectiveness. Furthermore, the focus on content pre-caching overlooks task caching's critical role in addressing high computational demands with limited UAV resources. This research aims to jointly optimize UAV trajectories and task management strategies, including offloading, caching, and migration. Utilizing the Lyapunov optimization framework, we break down the complex optimization problem into manageable subproblems: UAV placement, user-UAV association, task offloading, scheduling, and bandwidth allocation, addressed iteratively using the Block Coordinate Descent method. Specifically, the scheduling subproblem is transformed into a non-convex quadratically constrained quadratic programming problem, managed effectively through semidefinite relaxation and a probabilistic mapping approach. Our simulations show that this integrated approach significantly boosts system throughput and reduces execution times compared to conventional methods. This study enhances the understanding of the interplay between UAV trajectory planning and task management, offering vital theoretical insights for advancing UAV-assisted MEC systems.**

***Index Terms*—Caching, Lyapunov optimization, migration, offloading, scheduling cost, trajectory, UAV-assisted MEC.**

Received 16 March 2024; revised 30 September 2024; accepted 23 October 2024. Date of publication 28 October 2024; date of current version 5 February 2025. This work was supported in part by the National Natural Science Foundation of China under Grant 62361056 and Grant 62362068, in part by the Applied Basic Research Foundation of Yunnan Province under Grant 202201AT070156, Grant 202201AT070203, Grant 202301AT070194, and Grant 202301AT070422, in part by the Open Foundation of Yunnan Key Laboratory of Software Engineering under Grant 2023SE208, and in part by the Open Foundation of Yunnan Key Laboratory of Smart City in Cyberspace Security under Grant 202105AG070010-ZN-10. Recommended for acceptance by A. Garcia-Saavedra. *(Corresponding author: Zhenli He.)*

Mingxiong Zhao and Zhenli He are with the Engineering Research Center of Cyberspace, Yunnan Key Laboratory of Software Engineering, Kunming 650091, China, and also with the School of Software, Yunnan University, Kunming 650091, China (e-mail: jimmyzmx@gmail.com; hezl@ynu.edu.cn).

Rongqian Zhang is with the School of Information Science and Engineering, Yunnan University, Kunming 650091, China (e-mail: zhangrongqian@stu.ynu.edu.cn).

Keqin Li is with the Department of Computer Science, State University of New York, New Paltz, NY 12561 USA (e-mail: lik@newpaltz.edu).

This article has supplementary downloadable material available at https://doi.org/10.1109/TMC.2024.3486995, provided by the authors.

Digital Object Identifier 10.1109/TMC.2024.3486995

I. Introduction

A. Motivation

As MOBILE Edge Computing (MEC) becomes increasingly prevalent, it signifies a critical advancement in computational efficiency at the network's edge. The advent of UAV-assisted MEC systems emerges as a key evolution [1]. This shift not only enhances the capabilities of traditional MEC infrastructures but also offers a revolutionary approach to overcome the inherent constraints of ground-based MEC deployments. UAV-assisted MEC enables real-time data analytics for precision agriculture, rapid emergency responses for public safety, and advanced urban planning for smart cities. The adaptive nature of UAVs to meet varying computational demands enables the provision of real-time processing power in areas previously limited by infrastructure. Integrating UAV-assisted MEC with burgeoning sectors like the Internet of Things (IoT), virtual reality (VR), and live streaming promises a leap towards a more agile and efficient computational paradigm [2].

However, this integration presents complex challenges, primarily due to UAVs' limited computational resources and communication coverage, as well as user mobility. Achieving optimal resource utilization requires a comprehensive strategy that seamlessly integrates UAV trajectory control, strategic task offloading, caching, and migration techniques. The complexity arises from the interdependencies among these strategies. For instance, decisions on UAV trajectory directly impact the efficiency of task offloading, while approaches to caching and migration significantly influence subsequent UAV positioning. This complex interplay necessitates advanced optimization techniques to effectively harmonize these strategies, highlighting the challenge of designing an efficient UAV-assisted MEC network.

Existing research efforts have explored various aspects of this challenge, albeit often in isolation. Joint optimization studies of UAV trajectories and task offloading focus on improving communication efficiency, reducing energy consumption, and optimizing computational resource allocation to enhance mobile edge computing performance [3]. Concurrently, task migration and caching strategies have been investigated to alleviate computational burdens and enhance system responsiveness [4], [5]. Despite these advancements, the literature reveals a notable gap: the tendency to overlook the synergistic potential among different strategies, which could limit their collective effectiveness. Moreover, existing studies on caching strategies within UAV-assisted MEC primarily focus on hot data or content pre-caching, such as videos, images, and audio, neglecting the crucial aspect of computational task caching. This oversight is significant as computational task caching plays a vital role in addressing the contradiction between intense computational demands and severely limited computational resources. By incorporating computational task caching mechanisms, the probability of service denial events can be significantly reduced, thereby further improving system throughput, request acceptance rate, and response speed.

This paper seeks to bridge this gap by examining the complex relationships among UAV trajectory control, computational offloading, task migration, and caching strategies within the UAV-assisted MEC framework, with a particular focus on the underexplored area of computational task caching. By adopting a holistic optimization approach, this research aims to elucidate the intricate dynamics of these strategies and their cumulative impact on system performance, thereby contributing both to the theoretical understanding of these interactions and providing practical insights to inform the architectural development of future UAV-assisted MEC systems.

B. Our Contributions

In this study, we explore the joint optimization of UAV trajectories and task-related strategies, including offloading, caching, and migration, within UAV-assisted MEC environments to enhance system throughput. We examine the relationship between UAV trajectory control and task management strategies. By applying the Lyapunov method, we balance minimizing system costs with maximizing throughput, addressing the challenge of optimizing interconnected variables using the Block Coordinate Descent (BCD) method. Our experimental results highlight the effectiveness of our comprehensive framework in improving system throughput while meeting scheduling cost constraints. Our contributions include:

- We establish a multi-UAV, multi-user, and multi-time slot MEC system model that captures the evolution of user tasks over time and integrates these tasks with UAV trajectory planning for optimal scheduling operations. We then formally define the optimization problem addressed in this paper.
- We propose an online optimization algorithm based on the Lyapunov method to address the optimization problem. This approach deconstructs the complex original problem into a manageable single time slot scenario, enabling a strategic equilibrium between task scheduling costs and system throughput.
- We develop a suite of algorithms to address the single time slot challenge as outlined by the Lyapunov method. This suite includes a task scheduling-oriented UAV deployment algorithm, a system-adaptive task offloading algorithm, and a multi-coupled binary integer programming solution derived from the semi-positive definite quadratically constrained quadratic programming (QCQP) problem. These algorithms facilitate critical task scheduling decisions, significantly influenced by UAV trajectory optimization. Consequently, we utilize task migration and caching data as heuristic indicators to refine UAV trajectories, focusing on determining the optimal deployment locations for each time slot.
- We execute a thorough complexity analysis of our proposed algorithm, supplemented by experimental simulations and comparative evaluations. These investigations not only affirm the robustness of our theoretical model but also emphasize the distinct advantage of our specialized approach over traditional algorithms.

This paper is structured as follows: Section II reviews the literature. Section III models the scenarios and defines the optimization problem. Section IV introduces a Lyapunov-based algorithm for simplifying the problem per time slice. Section V details a BCD strategy, addressing sub-problems related to UAV trajectory and task management. Sections VI and VII describe the proposed algorithms and analyze their performance. The paper concludes in Section VIII.

II. Related Work

This section reviews the existing work related to our study. *Due to space constraints, a more detailed analysis of the related work discussed is available in Section I of the supplementary material, available online.*

A. Joint Optimization of UAV Trajectories and Task Offloading

In UAV-assisted MEC networks, task offloading involves transferring intensive or sensitive tasks from ground devices to UAVs or indirectly to edge servers, with methods categorized into binary and partial offloading. Managing UAV flight paths effectively is crucial due to their high mobility, which ensures optimal service quality and offloading efficiency. The authors in [6] enhanced load balancing and reduced energy consumption through strategic UAV trajectory optimization. In [7], the authors improved system performance using a deep reinforcement learning approach that synchronizes task offloading and UAV trajectory planning. The authors in [8] optimized task processing fairness and energy efficiency by fine-tuning offloading strategies and flight paths.

Dynamic resource allocation between UAVs and edge computing resources is essential for efficient task offloading. The authors in [9] developed methods to minimize energy usage while optimizing offloading and resource allocation, utilizing advanced programming techniques. Integrated strategies for UAV trajectory and resource management, as discussed in [3], [10], [11], quickly adapt to network conditions, thereby enhancing offloading efficiency and system performance.

Despite these advancements, a critical challenge persists: UAVs' limited computing resources, compared to traditional edge servers, often result in service denials post-offloading due to insufficient capacity. To address this, enhancing computing capacity through innovative offloading, caching, and migration strategies is crucial to overcoming the inherent limitations of UAVs and ensuring robust MEC service delivery.

B. Optimization of Caching in UAV-Assisted MEC Networks

Network load fluctuations necessitate effective caching strategies in MEC systems to enhance overall efficiency. The authors in [5] devised computational offloading and caching strategies that reduce UAV operational times by utilizing temporary task storage. Similarly, the authors in [12] and [13] integrated task offloading with resource allocation to optimize energy consumption and reduce latency. Further advancing this approach, the authors in [14] implemented a cloud-edge architecture where UAVs and cloud systems collaboratively provide efficient caching and computational services.

Economic and operational enhancements are also crucial. For instance, the authors in [15] optimized content caching in 5G networks to improve profitability through incentive mechanisms. In [16], the authors introduced a neural-blockchain method for on-demand caching, which enhances mobile edge communication reliability. These innovations not only address the operational challenges associated with UAVs' limited computational capacities but also pave the way for more advanced MEC systems.

However, the strategic use of caching to optimize task processing during peak times remains under-explored. Current research primarily focuses on performance metrics such as cache hit rates and specific applications like content streaming [17], [18], [19], [20], [21], [22]. Investigating the potential to cache time-sensitive tasks during computational peaks and processing them as capacity allows could significantly enhance system responsiveness. Moreover, further exploration into the interplay between caching, UAV trajectory planning, and task offloading could lead to substantial improvements in MEC system performance.

C. Optimization of Migration in UAV-Assisted MEC Networks

UAVs utilize onboard computational resources to dynamically process tasks. When tasks exceed their capabilities, migrating them to other UAVs or servers with sufficient resources becomes essential. The authors in [4], [23], and [24] illustrate systems that enhance processing efficiency through inter-UAV task migration, reducing latency and supporting applications with complex dependencies. Additionally, the authors in [2] developed a system that integrates UAVs with terrestrial base stations to optimize resource allocation and reduce latency, addressing the energy demands of mobile users.

However, existing literature often overlooks the complex interactions among UAV trajectory control, task caching, migration, and offloading. An in-depth exploration of these dynamics could substantially improve the efficiency and performance of MEC networks. Integrating trajectory optimization with robust task management can enhance UAV operations, making them more responsive and energy-efficient, especially in dynamic environments. This approach necessitates advanced algorithms capable of adapting to evolving conditions, ensuring optimal UAV performance across varying network scenarios.

Our research significantly diverges from existing studies in UAV-assisted MEC networks, characterized by the following unique aspects:

- We introduce a task caching mechanism for UAVs, allowing them to cache tasks in addition to performing computations and migrations. This approach mitigates computational burdens, prevents task accumulation, and ensures efficient task processing over time.
- Recognizing the dynamics between offloading, computation, migration, and caching, we develop dynamic scheduling and caching strategies for UAVs using Lyapunov optimization. This method optimizes system throughput and manages scheduling costs while considering computational needs, task latency, and UAVs' energy and caching capacities.
- We present a UAV deployment algorithm designed for a single time slot, aiming to significantly boost system throughput. We innovate by combining UAV trajectory planning with task scheduling strategies, enhancing the K-means algorithm with heuristic considerations such as task migration and caching. This results in adaptive UAV deployment for each time slot, optimizing trajectories.

To the best of our knowledge, this is the first attempt to propose a dynamic task scheduling strategy which comprehensively incorporates UAV trajectory, task offloading, migration, and caching within a UAV-assisted MEC framework.

III. System Model and Problem Formulation

This section details the service scenarios addressed and rigorously formalizes the optimization problem. *Symbols and their definitions can be found in Section II of the supplementary material, available online*.

A. Service Scenario

The service scenario in our system involves multiple UAVs and mobile users within overlapping regions. Users freely transmit data using Frequency Division Multiple Access (FDMA) technology, while UAVs dynamically allocate computing services across these areas.

For conciseness, let $\mathcal{U} = \{1, 2, \ldots, U\}$ denote the set of UAVs, and $\mathcal{I} = \{1, 2, \ldots, I\}$ represent the set of mobile users with computing demands. To address the unpredictable mobility of users, we employ a discrete-time slot model, as previously explored in [11], [25], [26]. The total duration D is segmented into τ discrete time slots, indexed by $\mathcal{T} = \{0, 1, \ldots, t, \ldots, \tau - 1\}$, where t represents each time slot. Hence, the UAV trajectory can be depicted as $\mathcal{Q}_u = \{\boldsymbol{q}_u(0), \boldsymbol{q}_u(1), \ldots, \boldsymbol{q}_u(t), \ldots, \boldsymbol{q}_u(\tau - 1)\}$, where $\boldsymbol{q}_u^{\text{UAV}}(t)$ denotes the UAV's position at time slot t. In each time slot, every user may generate a task based on their demand. These tasks are then associated with the UAV currently covering them, denoted by $s_{i,u}(t) = 1$, indicating that user i is associated with UAV u during time slot t. A non-association is represented by $s_{i,u}(t) = 0$. The task generated by user i at time slot t, $l_i(t)$, is characterized by two parameters: $\{e_i(t), w_i(t)\}$, which represent the amount of input data and the required computing resource for processing, respectively.

At time slot t, user i faces a decision regarding task offloading to UAV u, denoted by $z_{i,u}(t) = 1$. Conversely, $z_{i,u}(t) = 0$ indicates the decision not to offload during that time slot. It is assumed that the task must be offloaded to UAVs for computation assistance in the same time slot it is generated. Denote $a_{i,u}(t)$ as the computation assignment. Specifically, $a_{i,u}(t) = 1$ signifies that the task $l_i(t)$ is being processed by UAV u. Otherwise,

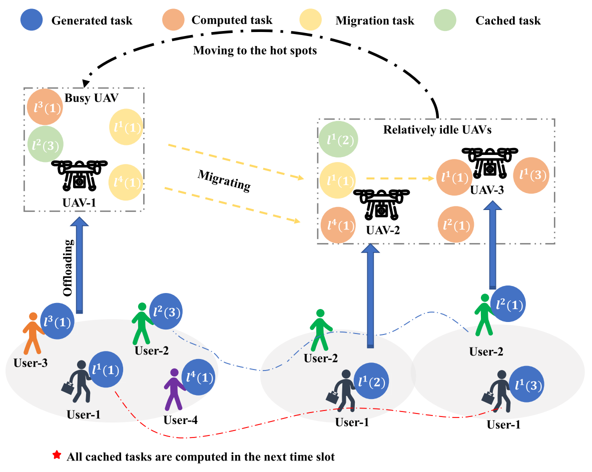


Fig. 1. System service scenario.

$a_{i,u}(t) = 0$. At each time slot, both users and UAVs possess the capability to initiate task-processing requests. This flexibility arises from the constraint that UAVs, with limited resources, may migrate tasks to subsequent UAVs for processing. Complementing computation and migration, tasks can also be cached on UAVs, and these cached tasks will be processed in the subsequent time slot.

Fig. 1 illustrates the decision-making processes for task offloading, computation, migration, and caching in a scenario with four users across three time slots. User-1 and User-2 generate tasks while moving randomly, whereas User-3 and User-4, staying within UAV-1's coverage, generate tasks $\{l^1(1), l^3(1), l^4(1)\}$ at the initial time slot. Initially, tasks from User-1, User-3, and User-4 are offloaded to UAV-1, where $l^3(1)$ is processed immediately, and $\{l^1(1), l^4(1)\}$ are migrated to UAV-2 due to computational constraints. In the second time slot, User-1's task is cached at UAV-2 before processing, while $l^4(1)$ is computed and $l^1(1)$ is sent to UAV-3 for further processing. By the third time slot, User-1, now under UAV-3's coverage, generates $l^1(3)$, which is processed along with previously offloaded tasks. If User-3 and User-4 generate additional tasks in future time slots, UAV-2 and UAV-3 relocate to this hotspot to address the computation deficiency.

B. Communication Model

After outlining the service scenario, we explore the communication model that facilitates interactions among UAVs and between UAVs and mobile users.

The positions of user i and UAV u at time slot t are described using the vectors $\boldsymbol{q}_i(t) = (x_i(t), y_i(t), 0)^T$ and $\boldsymbol{q}_u^{\text{UAV}}(t) = (x_u(t), y_u(t), h_u)^T$, respectively. Here, $x_i(t)$ and $y_i(t)$ represent the horizontal coordinates of user i, who is assumed to be at a constant altitude of 0. Similarly, the UAV's position is given by $x_u(t)$ and $y_u(t)$ in the horizontal plane, with a constant altitude denoted by h_u. The distances between UAV u and user i, as well as between UAV u and another UAV u', are calculated as $d_{i,u}(t) = \|\boldsymbol{q}_u^{\text{UAV}}(t) - \boldsymbol{q}_i(t)\|$ and $d_{uu'}(t) = \|\boldsymbol{q}_u^{\text{UAV}}(t) - \boldsymbol{q}_{u'}^{\text{UAV}}(t)\|$, respectively.

Considering the considerable likelihood of encountering non-line-of-sight (NLoS) paths in air-to-ground communications within 3D environments, the channel model for UAV-to-ground transmissions is designed to encompass both line-of-sight (LoS) and NLoS path loss probabilities [27]. In contrast, the channel model for UAV-to-UAV communications primarily operates under the assumption of LoS connections. The specifications for channel gains associated with LoS and NLoS links are detailed as $g_{i,u}^{\text{LoS}}(t) = \frac{g_0}{(d_{i,u}(t))^{\tilde{\iota}}}$, $g_{uu'}^{\text{LoS}}(t) = \frac{g_0}{(d_{uu'}(t))^{\tilde{\iota}}}$, and $g_{i,u}^{\text{NLoS}}(t) = \frac{kg_0}{(d_{i,u}(t))^{\tilde{\iota}}}$, respectively. Here, k is the NLoS attenuation factor, $\tilde{\iota}$ is the path loss exponent, and g_0 denotes the channel gain at a reference distance $d_0 = 1\,m$.

1) UAVs and Their Users: The composite channel gain between UAV u and its corresponding user i combines both LoS and NLoS components, given as $g_{i,u}(t) = \frac{\hat{P}_{i,u}^{\text{LoS}}(t) g_0}{(d_{i,u}(t))^{\tilde{\iota}}}$, where $\hat{P}_{i,u}^{\text{LoS}}(t) = P_{i,u}^{\text{LoS}}(t) + [1 - P_{i,u}^{\text{LoS}}(t)]k$ denotes the adjusted LoS probability, factoring in the attenuation due to the NLoS channel with $k < 1$. Moreover, $P_{i,u}^{\text{LoS}}(t) = \frac{1}{1+\alpha \exp(-\beta(\theta_{i,u}(t)-\alpha))}$ is the probability of LoS [28], where α and β represent environmental constants tailored to specific settings (urban, suburban, high-rise city, rural, etc.), and $\theta_{i,u}(t) = \frac{180}{\pi} \arctan\left(\frac{h_u}{\|(x_u(t), y_u(t)) - (x_i(t), y_i(t))\|}\right)$ is the elevation angle from the user to the UAV.

In this case, the data transmission rate between the user and the UAV is given as $r_{i,u}(t) = B_i(t) \log_2(1 + \frac{p_i(t) g_{i,u}(t)}{N_0 B^i(t)})$, where $B_i(t)$ is a fixed bandwidth allocated to user i, $p_i(t)$ denotes the user's transmission power in time slot t, and N_0 is the noise spectral density.

2) UAV To UAV: With bandwidth allocation between UAVs contingent upon migration demands, the transmission rate between them, primarily influenced by the prevailing LoS connections, is given by $r_{i,uu'}(t) = b_{i,uu'}(t) B \log_2(1 + \frac{p_u(t) g_{uu'}^{\text{LoS}}(t)}{N_0 b_{i,uu'}(t) B})$, where $b_{i,uu'}(t)$ represents the proportion of bandwidth allocated to user i for its task migration between UAVs, $p_u(t)$ denotes the transmission power of the UAV, and B represents the total bandwidth available between UAVs.

C. Scheduling Model

Our scheduling model considers migrating or caching tasks for deferred processing when immediate execution on UAVs is not feasible.

We define $m_{i,uu'}(t) \in \{0, 1\}$ to indicate whether UAV u migrates task $l_i(t)$ to UAV u' in time slot t, and $o_{i,u}(t) \in \{0, 1\}$ to signify whether the task of user i is cached on UAV u in time slot t. The set $\mathcal{I}_m^u(t)$ identifies tasks on UAV u that are awaiting migration, denoted by $\mathcal{I}_m^u$ for simplicity. The number of tasks in this set at time slot t is determined by the following expression

$$I_m^u(t) = \sum\nolimits_{i \in \mathcal{I}_s} z_{i,u}(t) + \sum\nolimits_{i \in \mathcal{I}_z} \sum\nolimits_{u' \in \mathcal{U} \setminus \{u\}} m_{u'u}^i(t-1) \\ - \sum\nolimits_{i \in \mathcal{I}_z} a_{i,u}(t) - \sum\nolimits_{i \in \mathcal{I}_z} o_{i,u}(t), \quad (1)$$

where $\mathcal{I}_s$ and $\mathcal{I}_z$ represent the sets of users with tasks associated and offloaded to the system, respectively. Additionally, it is important to note that task i stored on UAV u will occupy the amount of cache space, denoted as $o_{i,u}(t) e_{i,u}(t)$.

Once offloading is complete, the system evaluates computation, caching, and migration decisions, prioritizing task latency and UAV energy consumption. The scheduling cost is represented as

$$c_{i,u}(t) = \varpi_1 D_{i,u}(t) + \varpi_2 E_{i,u}(t), \tag{2}$$

where ϖ acts as a balancing parameter between delay and energy consumption. In this context, $D_{i,u}(t)$ encompasses the delay incurred during transmission, computation, and caching,

$$D_{i,u}(t) = \sum\nolimits_{u' \in \mathcal{U}\backslash\{u\}} m_{i,uu'}(t)\frac{e_{i,u}(t)}{r_{i,uu'}(t)} + a_{i,u}(t)\frac{e_{i,u}(t)}{w_{i,u}(t)} + o_{i,u}(t)\frac{D}{\tau}, \tag{3}$$

where $e_{i,u}(t) = e_i(t)$ and $w_{i,u}(t) = w_i(t)$, signifying that the quantity of input data and the necessary number of CPU cycles for the user's task remain constant. Here, u' is employed to denote the specific UAV responsible for processing that task.

Moreover, $E_{i,u}(t)$ quantifies additional energy consumption resulting from task scheduling, expressed by

$$E_{i,u}(t) = \sum\nolimits_{u' \in \mathcal{U}\backslash\{u\}} m_{i,uu'}(t)p_u(t)\frac{e_{i,u}(t)}{r_{i,uu'}(t)} + a_{i,u}(t)\eta_c\frac{e_{i,u}(t)}{w_{i,u}(t)} + o_{i,u}(t)\eta_o\frac{D}{\tau}, \tag{4}$$

where η_c and η_o represent the energy consumption coefficients for UAV computation and caching, respectively [29], [30].

Given the cost advantage of caching over computation and migration, yet with limited UAV cache space, the system endeavors to achieve long-term stability in cache space [11]:

$$\lim_{\tau\to\infty}\frac{1}{\tau}\sum\nolimits_{t\in\mathcal{T}}\mathbb{E}\{O_u(t)\} \le \infty, \forall u, \tag{5}$$

where $O_u(t)$ aggregates the occupied space for tasks cached on UAV u, given by

$$O_u(t) = \sum\nolimits_{i\in\mathcal{I}_o} o_{i,u}(t)e_{i,u}(t), \tag{6}$$

where $\mathcal{I}_o$ is the set of cache tasks, and $\mathcal{I}_o \subset \mathcal{I}_z$. Notice that a task is exclusively either computed, cached by the UAV, or migrated to the next UAV. Consequently, $a_{i,u}(t)$, $m_{i,uu'}(t)$, and $o_{i,u}(t)$ cannot all be equal to 1 within the same time slot.

Moreover, the scheduling cost must comply with a budget constraint

$$\lim_{\tau\to\infty}\frac{1}{\tau}\sum\nolimits_{t\in\mathcal{T}}\mathbb{E}\{C_u(t)\} \le \tilde{C}, \forall u, \tag{7}$$

where $\tilde{C}$ signifies the system's average long-term cost budget, and $C_u(t)$ consolidates individual task costs for UAV u, expressed as

$$C_u(t) = \sum\nolimits_{i\in\mathcal{I}_z} c_{i,u}(t). \tag{8}$$

D. Problem Formulation

This paper aims to enhance the system's long-term average service quantity (throughput) [25]. We achieve this through a comprehensive optimization of task association, offloading, computation, migration, caching, bandwidth allocation, and UAV trajectories. The problem of maximizing long-term average service quantity is mathematically formulated as

$$\mathcal{P}: \max_{\substack{\boldsymbol{Q},\boldsymbol{S},\boldsymbol{Z},\\ \boldsymbol{A},\mathsf{M},\boldsymbol{O},\mathsf{B}}} \lim_{\tau\to\infty}\frac{1}{\tau}\sum_{t\in\mathcal{T}}\mathbb{E}\{Y(z_{i,u}(t))\} \tag{9a}$$

$$\text{s.t. } s_{i,u}(t) \in \{0,1\}, \forall i \in \mathcal{I}, \forall u, \tag{9b}$$

$$z_{i,u}(t) \in \{0,1\}, \forall i \in \mathcal{I}_s, \forall u, \tag{9c}$$

$$a_{i,u}(t) \in \{0,1\}, \forall i \in \mathcal{I}_z, \forall u, \tag{9d}$$

$$m_{i,uu'}(t) \in \{0,1\}, \forall i \in \mathcal{I}_z, \forall u, u' \in \mathcal{U}\backslash\{u\}, \tag{9e}$$

$$o_{i,u}(t) \in \{0,1\}, \forall i \in \mathcal{I}_z, \forall u, \tag{9f}$$

$$\sum\nolimits_{u\in\mathcal{U}} z_{i,u}(t) \le 1, \forall i \in \mathcal{I}, \tag{9g}$$

$$\sum_{u'\in\mathcal{U}\backslash\{u\}} m_{i,uu'}(t) + a_{i,u}(t) + o_{i,u}(t) = 1, \forall i \in \mathcal{I}_z, \forall u, \tag{9h}$$

$$z_{i,u}(t) \le s_{i,u}(t), \forall i \in \mathcal{I}, \forall u, \tag{9i}$$

$$z_{i,u}(t)\frac{e_{i,u}(t)}{r_{i,u}(t)} + \sum\nolimits_{u'\in\mathcal{U}\backslash\{u\}} m_{i,uu'}(t)\frac{e_{i,u}(t)}{r_{i,uu'}} + a_{i,u}(t)\frac{e_{i,u}(t)}{w_{i,u}(t)} \le \frac{D}{\tau}, \forall i \in \mathcal{I}_s, \forall u, \tag{9j}$$

$$\sum\nolimits_{i\in\mathcal{I}_m^u} b_{i,uu'}(t) \le 1, \forall u, u', \tag{9k}$$

$$\sum\nolimits_{i\in\mathcal{I}_z} a_{i,u}(t)w_{i,u}(t) \le W_u^a(t), \forall u, \tag{9l}$$

$$\lim_{\tau\to\infty}\frac{1}{\tau}\sum\nolimits_{t\in\mathcal{T}}\mathbb{E}\{O_u(t)\} \le \infty, \forall u, \tag{9m}$$

$$\lim_{\tau\to\infty}\frac{1}{\tau}\sum\nolimits_{t\in\mathcal{T}}\mathbb{E}\{C_u(t)\} \le \tilde{C}, \forall u, \tag{9n}$$

where $Y(z_{i,u}(t)) = \sum_{u\in\mathcal{U}}\sum_{i\in\mathcal{I}_s} z_{i,u}(t)$ represents the long-term system throughput. Define $\boldsymbol{Z} \triangleq \{z_{i,u}\} \in \mathbb{R}^{U\times I}$, $\boldsymbol{A} \triangleq \{a_{i,u}\} \in \mathbb{R}^{U\times I}$, $\mathsf{M} \triangleq \{m_{i,uu'}\} \in \mathbb{R}^{I\times U\times(U-1)}$, $\boldsymbol{O} \triangleq \{o_{i,u}\} \in \mathbb{R}^{U\times I}$, and $\mathsf{B} \triangleq \{b_{i,uu'}\} \in \mathbb{R}^{I_m\times U\times(U-1)}$, where '$(t)$' is omitted for simplicity. Moreover, we further define $\boldsymbol{Q} \triangleq \{\boldsymbol{q}_u\} \in \mathbb{R}^{U\times 3}$, and $\boldsymbol{S} \triangleq \{s_{i,u}\} \in \mathbb{R}^{U\times I}$ as the deployments of UAVs and the User-UAV association, respectively.

To be specific, constraint (9g) ensures each task is offloaded to a single UAV, while (9h) mandates that a task in any time slot must choose among computation, migration, and caching. Constraint (9i) dictates that a task is offloaded to the same UAV with which the user is associated, influenced by $s_{i,u}$ and UAV trajectories. According to (9j), tasks must be offloaded and processed within the same time slot, assuming negligible cache writing time [30], [31].

Bandwidth allocation for task migration among UAVs is limited by (9k), while (9l) ensures the total computational load on a UAV at time t, $W_u^a(t)$, does not exceed its capacity, defined as $W_u^a(t) = W_u - W_u^o(t)$, where W_u is the UAV's total computing capacity, and $W_u^o(t)$ is the load from ongoing caching tasks, determined by previous decisions, i.e., $W_u^o(t) = o_{i,u}(t-1)w_{i,u}(t-1)$. Given the dynamic nature of user requests, optimization focuses on maximizing long-term system utility within a set cost budget. Constraint (9m) ensures cache stability, and (9n) maintains scheduling cost, balancing migration, caching, and computation decisions [32], [33], [34]. Constraint (9g) mandates that tasks can only be offloaded to a single UAV. Constraint (9h) requires that a task in the same time slot must choose among computation, migration, and caching. Constraint (9i) ensures that user tasks are offloaded to the same UAV with which the user is associated. According to (9j), tasks must be offloaded and processed within the current time slot, assuming that cache usage occupies the entire time slice, thus not constrained. Bandwidth allocation must comply with (9k). Constraint (9l) stipulates that the total computational load on a UAV must not exceed its available capacity, defined as $W_u^a(t) = W_u - W_u^o(t)$, where W_u represents the UAV's total computing power and $W_u^o(t)$ represents the load from ongoing caching tasks. This load is determined by the caching decision from the previous time slot and is calculated as $W_u^o(t) = o_{i,u}(t-1)w_{i,u}(t-1)$.

Given the unpredictable nature of user service requests, optimizing long-term system utility within a long-term cost budget is common practice. Constraint (9m) ensures cache stability, while the scheduling cost must satisfy (9n). Stability in scheduling cost is crucial for balancing decisions related to migration, caching, and computation.

IV. Lyapunov-Based Online Optimization Framework

To address the complexity of problem $\mathcal{P}$, particularly its long-term objectives and constraints (9m) and (9n), we employ the Lyapunov optimization technique. This reduces the dynamic problem to a single-time slot, making it more manageable and computationally feasible [11], [25], [35].

For managing long-term scheduling costs and enhancing task scheduling efficiency, a virtual queue $G_u(t)$ is introduced for each UAV. This queue tracks the accumulated scheduling costs, reflecting any excess or deficit for UAV u at time slot t. The update mechanism for this virtual queue is defined as

$$G_u(t+1) = \max\{G_u(t) + C_u(t) - \tilde{C}, 0\}, \tag{10}$$

initiated with $G_u(0) = 0$. Concurrently, the actual cache queue $\Omega_u(t+1)$ functions to temporarily store tasks on each UAV, represented as

$$\Omega_u(t+1) = \max\{\Omega_u(t) + O_u(t) - \tilde{O}_u(t), 0\}, \tag{11}$$

with $\tilde{O}_u(0) = 0$, and $\tilde{O}_u(t) = \sum_{i\in\mathcal{I}_o} o_{i,u}(t-1)e_{i,u}(t-1)$. Notice that $\tilde{O}_u(t)$ reflecting decisions from the previous slot affects the current slot's caching state.

With a set of U UAVs in the system, each maintaining a queue, the composite vector of scheduling costs and cache space is

$$\Theta(t) = ([G_u(t), \Omega_u(t)]_{u\in\mathcal{U}}). \tag{12}$$

The quadratic Lyapunov drift plus function indicating the system queue status is given as

$$\mathbb{L}(\Theta(t)) = \frac{1}{2}\Big[\sum\nolimits_{u\in\mathcal{U}} G_u(t)^2 + \Omega_u(t)^2\Big], \tag{13}$$

which reflects the system's queue backlog with lower values indicating greater stability. The Lyapunov drift, capturing the expected change in the Lyapunov function from one slot to the next, is given as

$$\Delta(\Theta(t)) = \mathbb{E}\left\{\mathbb{L}(\Theta(t+1)) - \mathbb{L}(\Theta(t))|\Theta(t)\right\}. \tag{14}$$

This framework simplifies complex long-term objectives into sequential single-time slot decisions, improving the manageability and effectiveness of scheduling. Our attention then shifts to the one-slot conditional Lyapunov drift, which is associated with the scheduling cost and cache queue backlog, as constrained in the following lemma.

Lemma 1: The one-slot conditional Lyapunov drift of the scheduling cost and cache queue backlog has an upper bound function, given as

$$\Delta(\Theta(t)) \le F + \sum_{u\in\mathcal{U}} G_u(t)\mathbb{E}\{[C_u(t) - \tilde{C}]|\Theta(t)\} + \Omega_u(t)\mathbb{E}\{O_u(t) - \tilde{O}_u(t)|\Theta(t)\}, \tag{15}$$

where F is a constant given in proof of the lemma.

Proof: The proof of Lemma 1 is provided in Section III of the supplementary material, available online, due to space constraints. □

Accordingly, the optimization objective is formulated as the Lyapunov drift plus a penalty function, with the upper bound expressed as

$$\begin{aligned}\Delta(\Theta(t)) - V\mathbb{E}\{Y(z_{i,u}(t))|\Theta(t)\} \le F\\ + \sum_{u\in\mathcal{U}} G_u(t)\mathbb{E}\{C_u(t) - \tilde{C}|\Theta(t)\} - V\mathbb{E}\{Y(z_{i,u}(t))|\Theta(t)\}\\ + \Omega_u(t)\mathbb{E}\{O_u(t) - \tilde{O}_u(t)|\Theta(t)\},\end{aligned} \tag{16}$$

where V is a control parameter that balances queue stability and system service quantity. The system dynamically adjusts V based on the current queue backlog, allowing a nuanced trade-off between service quantity and stability.

Proof: Due to space constraints, the analysis and proof of the Lyapunov drift-plus-penalty can be found in Section III of the supplementary material, available online. □

Streamlining this inequality refines the optimization objective for a single time slot to minimizing the upper bound function. Where minimizing the conditional expectation of chance guides this approach, the solution seeks to minimize the upper bound without incorporating expectations [25]. The resulting optimization problem for a single time slot is thus formulated as

$$\min_{\substack{\boldsymbol{Q},\boldsymbol{S},\boldsymbol{Z},\\ \boldsymbol{A},\mathrm{M},\boldsymbol{O},\mathrm{B}}} \Gamma + \sum_{u\in\mathcal{U}} G_u(t)C_u(t) + \Omega_u(t)O_u(t) - VY(z_{i,u}(t))$$

$$\mathcal{P}' : \text{ s.t. (9c)–(9l)}, \tag{17a}$$

where $\Gamma = -[\sum_{u\in\mathcal{U}} G_u(t)\tilde{C} + \Omega_u(t)\tilde{O}_u(t)]$ remains constant within time slot t.

Building upon the preceding discussion, Algorithm 1 intricately delineates Lyapunov online control process. Subsequently, in the following section, we introduce an algorithm aimed at optimizing scheduling decisions and refining the UAV trajectory, ultimately maximizing system service quantity.

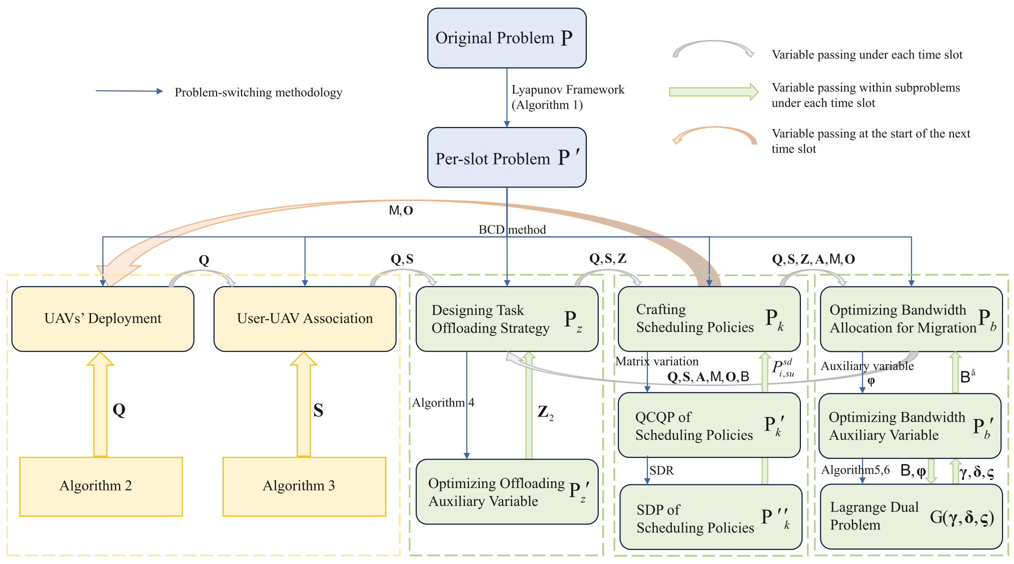

Original Problem P
Problem-switching methodology
Lyapunov Framework (Algorithm 1)
Variable passing under each time slot
Variable passing within subproblems under each time slot
Variable passing at the start of the next time slot
Per-slot Problem P′
M,O
BCD method
Q
Q,S
Q,S,Z
Q,S,Z,A,M,O
UAVs' Deployment
User-UAV Association
Designing Task Offloading Strategy P_z
Crafting Scheduling Policies P_k
Optimizing Bandwidth Allocation for Migration P_b
Matrix variation
Q,S,A,M,O,B
P^sd_i,su
Auxiliary variable
φ
B^å
Q
S
Algorithm 4
Z_2
QCQP of Scheduling Policies P′_k
Optimizing Bandwidth Auxiliary Variable P′_b
SDR
Algorithm5,6
B,φ
γ,δ,ς
Algorithm 2
Algorithm 3
Optimizing Offloading Auxiliary Variable P′_z
SDP of Scheduling Policies P″_k
Lagrange Dual Problem G(γ,δ,ς)

Fig. 2. Proposed solution framework and interrelation of optimization subproblems.

Algorithm 1: Lyapunov Online Control Process.

```
Input: Given control parameter $V$, constant $\Gamma$, scheduling cost budget $\tilde{C}$, and computing capacity $W_u$
Output: Scheduling cost queue $G_u(t+1)$ and cache queue $\Omega_u(t+1)$
1: Initialize $G_u(0)$, $\Omega_u(0)$ and $\tilde{O}_u(0)$ to 0;
2: for each $t \in \mathcal{T}$ do
3:     Solve $\mathcal{P}'$ to obtain the solution of each variable;
4:     for each $u \in \mathcal{U}$ do
5:         Calculate $C_u(t)$ and $O_u(t)$ based on (2) and (6);
6:         Update $G_u(t+1)$ based on (10);
7:         Obtain $\tilde{O}_u(t)$ based on $o_{i,u}(t-1)$ and $e_{i,u}(t-1)$;
8:         Update $\Omega_u(t+1)$ based on (11);
9:     end for
10: end for
```

V. Optimization Algorithm for Per-Slot Problem

In tackling the Lyapunov-transformed single-time slot problem $\mathcal{P}'$, we employ the BCD technique [30], [36] to simultaneously design a spectrum of crucial variables: UAVs' deployment $\boldsymbol{Q}$, task offloading strategies $\boldsymbol{Z}$, and scheduling policies $\{\boldsymbol{A}, \mathsf{M}, \boldsymbol{O}\}$, along with migration bandwidth allocation B. As a result, $\mathcal{P}'$ is decomposed into four distinct parts: 1) Customizing the initialization schemes for UAVs' deployment and User-UAV association, 2) Designing the optimal task offloading strategy $\mathcal{P}_z$, 3) Crafting effective task scheduling policies $\mathcal{P}_k$, and 4) Fine-tuning bandwidth allocation for migration $\mathcal{P}_b$, to maximize overall system throughput in an iterative manner, as shown in the Fig. 2. *Due to space limitations, the interdependencies among the sub-problems are detailed in Section IV of the supplementary material, available online.*

A. Initialization for UAVs' Deployment and User-UAV Association

In this subsection, we aim to develop a strategic plan for UAV deployment and establish the association between users and UAVs, creating a feasible initialization environment for each time slot. The problem is formulated as follows:

$$\mathcal{P}_q : \min_{\boldsymbol{Q},\boldsymbol{S}} \sum\nolimits_{u\in\mathcal{U}} G_u(t)C_u(t)$$
$$\text{s.t. (9b), (9i), (9j).} \tag{18a}$$

Given the unpredictable nature of user movements and task requests, UAV deployment at the beginning of each time slot initially bypasses specific constraints to focus on enhancing system throughput. Subsequent subproblems will address the constraints for $\mathcal{P}_q$. This approach ensures that while the initial UAV deployment centers on maximizing system throughput without immediate consideration of constraints, later adjustments and decisions are made to strictly adhere to the necessary constraints, thereby maintaining the integrity of $\mathcal{P}_q$'s constraints. It's important to note that the UAV deployment at the outset of each time slot is intricately linked to the migration and caching status of tasks from the preceding time slot within the system.

1) Deploying UAVs: Upon examining $\mathcal{P}'$ in connection with (17), it becomes evident that the deployment of UAVs is intricately tied to constraint (9j), which predominantly influences both the offloading rate from users to UAVs and the transmission rate for migration between UAVs. Furthermore, a more effective deployment of UAVs provides flexibility in designing other variables in subsequent optimizations, ultimately leading to a higher quantity of services for the system.

In practical scenarios, UAVs are strategically deployed in close proximity to users to efficiently address their task-processing needs, ensuring timely service provision. It's evident that a well-planned UAV deployment strategy can significantly enhance user access, accommodating as many tasks as possible and potentially improving system throughput. Furthermore, decisions related to task migration and caching, as indicated by (3) and (4), offer valuable insights into task or user distribution. An increase in the frequency of task migration among UAVs or caching for future computation may indicate a shortage of UAVs to serve areas with high user density, potentially leading to a reduction in system throughput. Additionally, such actions contribute to higher scheduling costs, as indicated by (2). Therefore, optimizing UAV deployment is essential to enhance system performance and reduce scheduling costs. Following this, some meaningful conclusions can be drawn regarding task caching and migration:

- *The impact of task caching on UAV trajectories:* Significant task caching on a UAV indicates its operation in a high-demand area, where the local computational requirements exceed the UAV's immediate processing capabilities. Consequently, a higher concentration of UAVs is necessary in such locations to adequately manage the workload. UAVs tasked with extensive caches should remain within their designated service zones to ensure continuous and effective support to dependent users.
- *The impact of task migration on UAV trajectory:* Frequent task migration by a UAV suggests a strong likelihood of its sustained presence in a particular area to serve local users. Conversely, for users whose tasks are consistently transferred to other UAVs, the reliability of continuous service from the initial UAV is diminished. This dynamic calls for a strategic balance in UAV deployment, ensuring stability in service provision while accommodating the flexible redistribution of tasks.

Motivated by the analyses presented above, our objective is to enhance system throughput through task-scheduling-oriented deployment of UAVs. To achieve this, we introduce I_u^{MO} as an indicator for the quantity of tasks to be migrated and cached, which guides UAV deployment for the next time slot. Leveraging the effectiveness of the K-means clustering algorithm, renowned for its application in user-location-based UAV deployment [37], [38], [39], we propose the task-scheduling-oriented deployment of UAVs, detailed in Algorithm 2. In particular, lines 15-16 of the algorithm suggest that users with cached tasks are less likely to become the next clustering center. This adjustment is made by updating the corresponding weight $\varpi_{i,u}^c$ based on the value of I_u^{MO}, resulting in its reduction as indicated in lines 13 and 16. Conversely, lines 17-19 imply that users with a higher number of migrated tasks are more likely to become the next center. This is because the related weight $\varpi_{i,u}^c$ is multiplied by the number of migration tasks for that user U_i^m within a given time period, increasing the probability of being selected as the next clustering center. Therefore, UAVs will move to specified locations at the next time slot, i.e., the newly generated clustering centers.

2) Determining User-UAV Association Relationship: In this subsection, our primary objective is to formalize the association relationship between users and UAVs. This becomes particularly crucial as UAV positions are initially assigned randomly and are subsequently influenced by user requests according to Section V-A1, directly impacting their service range. Furthermore, since users' task requests are generated randomly in each time slot, a pivotal decision lies in determining user associations within the UAV coverage for the current time slot. Specifically, this decision significantly affects the overall service capacity of the system.

To address this challenge, we propose employing the Distance Probability Rounding Method (DPRM) outlined in Algorithm 3 to determine the association decision $s_{i,u}(t)$. For simplicity, we denote $\boldsymbol{S} = \{s_{i,u}\} \in \mathbb{R}^{I\times U}$. The set $\Pi_i(t)$ is defined as the group of UAVs covering user i during time slot t. Essentially, user i can access any UAV within that set during time slot t. In this context, the probability for user i to be associated with UAV u at time slot t, is defined as $\mathsf{p}_{i,u}^s(t) = [1 - \frac{d_{i,u}(t)}{\tilde{r}_u(t)}]$. This probability is applicable to users within overlapping UAV coverage areas. Here, $\tilde{r}_u(t)$ represents the maximum coverage distance of UAV u at time slot t. The probability reflects the relative likelihood that a user remains within the coverage of a UAV, considering their respective positions.

Notice that constraint (9g) is addressed during the process of establishing the User-UAV association relationship, specifically from Step 6 to Step 9 of Algorithm 3. Additionally, the set $\mathcal{I}_s$, comprising users with tasks associated to UAVs, is derived following the procedure outlined in Step 11 of Algorithm 3.

B. Designing Task Offloading Strategy

Establishing a robust User-UAV association is pivotal for optimal task offloading to UAVs, enhancing overall system service quality. Upon task association, it enters the dynamic waiting queue $\mathcal{I}_s$ satisfying constraint (9i) with $s_{i,u} = 1$ for $\forall i \in \mathcal{I}_s$, its length fluctuating with each time slot. Crucially, if the decision to offload the user's task is determined, it is incorporated into the pending queue $\mathcal{I}_z$. Following this, tasks within $\mathcal{I}_z$ undergo scheduling for computation, migration, and caching. The dynamic nature of $\mathcal{I}_z$ and the imperative for task offloading, guided by constraints (9h), define the task offloading problem $\mathcal{P}_z$ as follows:

$$\mathcal{P}_z : \min_{\boldsymbol{Z}} \quad -VY(z_{i,u}(t))$$

$$\text{s.t. (9c), (9i), (9j).} \tag{19a}$$

This formulation constitutes a binary integer programming problem. It is crucial to note that the task offloading strategy must adhere to constraints (9i) and (9j), ensuring both task accessibility and the completion of offloading and scheduling processing within the current time slot. Constraint (9j) can be further rewritten as

$$z_{i,u}(t) \leq \frac{D_s r_{i,u}(t)}{e_{i,u}(t)}, \forall i \in \mathcal{I}_s, \forall u, \tag{20}$$

where $D_s \triangleq \frac{D}{\tau} - m_{i,uu'}(t)\frac{e_{i,u}(t)}{r_{i,uu'}(t)} - a_{i,u}(t)\frac{e_{i,u}(t)}{w_{i,u}(t)}$. In conformity with constraint (9c), we introduce an intermediary variable denoted as $z_{i,u,1}(t)$. If the condition on the right-hand side (RHS) of (20) holds true, i.e., $\frac{D_s r_{i,u}(t)}{e_{i,u}(t)} \geq 1$, then $z_{i,u,1}(t)$ is set to 1. This signifies that the task is deemed eligible for offloading and can be efficiently processed within a single time slot.

In the pursuit of our primary goal within the original problem $\mathcal{P}'$, which is to minimize costs such as scheduling and caching queues while maximizing system throughput, the value of $z_{i,u}(t)$ holds a pivotal role. It directly influences system throughput and indirectly affects scheduling costs. However, the direct determination of $z_{i,u}(t)$ based solely on $z_{i,u,1}(t)$ does not ensure the optimal achievement of our overarching objective. This is because optimizing $z_{i,u,1}(t)$ alone may not lead to the most

Algorithm 2: Task-Scheduling-Oriented UAV Deployment (TSOUD).

```
Input: Given migration decision $\mathsf{M}(t-1)$, caching decision $\boldsymbol{O}(t-1)$ for the previous time slot, and the positions for users at the horizontal plane for the current time slot $\boldsymbol{q}_i^H(t)$, $\forall i$.
Output: The deployment for UAVs $\boldsymbol{q}_u^{\mathrm{UAV}}(t)$, $\forall u$.
 1: Initialize the set of clustering centers with $\mathbf{0} \in \mathbb{R}^2$ for all entries in $\boldsymbol{Q}^c \triangleq \{\boldsymbol{q}_1^c, \boldsymbol{q}_2^c, \ldots, \boldsymbol{q}_U^c\}$;
 2: Calculate the number of tasks for migration and caching with respect to the users within the coverage of UAV $u$, $I_u^{\mathrm{MO}}$, $\forall u$;
 3: for each UAV $u$ do
 4:   if $I_u^{\mathrm{MO}} == 0$ then
 5:     Randomly select user $i$'s position $\boldsymbol{q}_i^H(t)$, $\forall i$ as the first clustering center $\boldsymbol{q}_1^c$, and add it to $\boldsymbol{Q}^c$;
 6:   else
 7:     Set the position of UAV $u$ with the largest $I_u^{\mathrm{MO}}$ as the first clustering center $\boldsymbol{q}_1^c$, and add it to $\boldsymbol{Q}^c$;
 8:   end if
 9: end for
10: while The number of non-zero entries in set $\boldsymbol{q}_c$ is less than $U$ do
11:   for each user $i$ do
12:     Calculate $d_{i,u}^H = \min_{u \in \mathcal{U}} \|\boldsymbol{q}_i^H(t) - \boldsymbol{q}_u^c\|$;
13:     Set the weight of clustering center as $\varpi_{i,u}^c = d_{i,u}^H$;
14:     for each UAV $u$ do
15:       if $o_{i,u}(t-1) == 1$ then
16:         Set $\varpi_{i,u}^c = \frac{1}{I_u^{\mathrm{MO}}} \varpi_{i,u}^c$;
17:       else $m_{i,uu'}(t-1) == 1$
18:         Set the counter of migration for user $i$ $U_i^m(t) = U_i^m(t-1) + 1$;
19:         Set $\varpi_{i,u}^c = \frac{U_i^m}{I_u^{\mathrm{MO}}} \varpi_{i,u}^c$;
20:       end if
21:     end for
22:   end for
23:   Calculate the probability for user $i$ being the clustering center $P_{i,u}^c = \frac{\varpi_{i,u}^c}{\sum_{i \in \mathcal{I}} \varpi_{i,u}^c}$;
24:   Set the position of the user $\hat{i}$ referring to $\hat{i} = \arg\max_{i \in \mathcal{I}} P_{i,u}^c$ as the next clustering center, and add the corresponding $\boldsymbol{q}_{\hat{i}}$ to $\boldsymbol{Q}^c$;
25: end while
26: repeat
27:   Assign users to their corresponding nearest clustering center according to $\boldsymbol{Q}^c$, forming $U$ clusters;
28:   Update the locations of cluster centroids for each cluster with the mean of associated users' coordinates in that cluster based on the $K$-means method, and set it as $\boldsymbol{q}_u^{\mathrm{UAV}}(t)$;
29: until Convergence
30: return The deployment for UAVs $\boldsymbol{q}_u^{\mathrm{UAV}}(t)$, $\forall u$.
```

Algorithm 3: Distance Probability Rounding Method.

```
Input: Given $\boldsymbol{q}_i$ and $\boldsymbol{q}_u^{\mathrm{UAV}}$ for user $i$ and for UAV $u$, and the distance between them $d_{i,u}(t)$
Output: User-UAV association relationship $\boldsymbol{S}$
 1: Get the set of UAVs covering user $i$, $\Pi_i(t)$, and their corresponding covering radiuses $\tilde{r}_u(t)$, $\forall u \in \Pi_i(t)$;
 2: for each $i \in \mathcal{I}$ do
 3:   for each $u \in \Pi^i(t)$ do
 4:     Set $s_{i,u}(t) = 1$;
 5:   end for
 6:   if user $i$ accesses to more than two UAVs, i.e., $\sum_{u \in \mathcal{U}} s_{i,u}(t) > 1$ then
 7:     if $\hat{u} = \arg\max_u \{\mathsf{p}_{i,u}^s\}$, set $s_{i,\hat{u}}(t) = 1$ then
 8:       Set $s_{i,u}(t) = 0, \forall u \in \mathcal{U} \backslash \{\hat{u}\}$;
 9:     end if
10:   end if
11:   Put $i$ into $\mathcal{I}_s$ with $s_{i,\hat{u}}(t) = 1$;
12: end for
13: return $\boldsymbol{S}$
```

effective solution for minimizing costs and maximizing system throughput across the entire system. Moreover, the subsequent optimizations for other variables cannot further enhance system performance

To overcome this challenge, we introduce an auxiliary variable, $z_{i,u,2}(t)$. This additional variable undergoes iterative optimization alongside the remaining variables, fostering a collaborative effort to effectively serve the overall objective. This variable serves to "partially" determine the value of $z_{i,u}(t)$ and must satisfy $0 \leq z_{i,u,2}(t) \leq 1$, providing more flexibility for optimizing other variables. Additionally, it needs to meet the practical condition of eligibility for upload, expressed as $z_{i,u,2}(t) \leq z_{i,u,1}(t)$. Consequently, $\mathcal{P}_z$ can be further expressed as the following convex problem $\mathcal{P}'_z$ related with $z_{i,u,2}(t)$, given as

$$\mathcal{P}'_z : \min_{\boldsymbol{Z}_2} \; -VY(z_{i,u,2}(t)) \tag{21a}$$

$$\text{s.t.}\; 0 \leq z_{i,u,2}(t) \leq 1, \forall i \in \mathcal{I}_s, \forall u, \tag{21b}$$

$$z_{i,u,2}(t) \leq z_{i,u,1}(t), \forall i \in \mathcal{I}_s, \forall u, \tag{21c}$$

where $\boldsymbol{Z}_2 \triangleq \{z_{i,u,2}\} \in \mathbb{R}^{U \times I}$, and the obtained $z_{i,u,2}^\star$ is within the continuous interval [0,1]. Notice that when $z_{i,u,2}^\star > 0$, $z_{i,u,1}$ will be 1 to satisfy constraint (21c), and thus meet the constraints (9i) and (9j) of $\mathcal{P}_z$. Therefore, we set an expected value for $z_{i,u,2}^\star$ to make constraint (21c) more flexible for the maximization of the system's service quantity. When $z_{i,u,2}^\star$ is greater than or equal to ϵ, we set the value of $z_{i,u}$ to 1. Meanwhile, we examine the influence of varying expected values on system throughput and scheduling cost in Section V of the supporting material. Accordingly, we can design the task offloading strategy based on the obtained $\boldsymbol{Z}_2$, the details of which are given in Algorithm 4.

Notice that the set $\mathcal{I}_z$, comprising users with tasks offloaded to UAVs, is derived following the procedure outlined in Step 15 of Algorithm 4.

Algorithm 4: Designing Task Offloading Strategy.

```
Input: Given association relationship $s_{i,u}(t)$ referring to Section V-A2, the amount of input data for task $e_{i,u}(t)$ and transmission rate $r_{i,u}(t)$ under current time slots
Output: Task offloading strategy $\boldsymbol{Z}$
1: Calculate $\frac{D_s r_{i,u}(t)}{e_{i,u}(t)}$ according to the RHS of (20);
2: for each $i \in \mathcal{I}_s$ do
3:   for each $u \in \mathcal{U}$ do
4:     if $\frac{D_s r_{i,u}(t)}{e_{i,u}(t)} \geq 1$ then
5:       Set $z_{i,u,1}(t) = 1$;
6:     end if
7:   end for
8: end for
9: Optimize $z_{i,u,2}(t)$ according to $\mathcal{P}_z'$;
10: for each $i \in \mathcal{I}_s$ do
11:   for each $u \in \mathcal{U}$ do
12:     if $z^{\star}_{i,u,2}(t) \geq \epsilon$ then
13:       Set $z_{i,u}(t) = 1$;
14:     end if
15:     Put $i$ into $\mathcal{I}_z$ with $z_{i,u}(t) = 1$;
16:   end for
17: end for
18: return $\boldsymbol{Z}$
```

C. Crafting Scheduling Policies

Due to the random generation of users' task requests and potential limitations in UAV computation capabilities within their coverage areas, especially with bursted tasks, the necessity for a scheduling operation arises. This operation significantly influences the system's service quantity performance. The scheduling decisions encompass variables associated with user task computing, migration, and caching, and are determined by solving the outlined subproblem $\mathcal{P}_k$, outlined as

$$\mathcal{P}_k: \min_{\boldsymbol{A},\boldsymbol{O},\mathrm{M}} \sum\nolimits_{u\in\mathcal{U}} G_u(t)C_u(t) + \Omega_u(t)O_u(t) \tag{22a}$$

$$\text{s.t. (9d)–(9f), (9h), (9l),}$$

$$z_{i,u}(t)\frac{e_{i,u}(t)}{r_{i,u}(t)} + \sum\nolimits_{u'\in\mathcal{U}\backslash\{u\}} m_{i,uu'}(t)\frac{e_{i,u}(t)}{r_{i,uu'}} + a_{i,u}(t)\frac{e_{i,u}(t)}{w_{i,u}(t)} \leq \frac{D}{\tau}, \forall i \in \mathcal{I}_z, \forall u, \tag{22b}$$

which is characterized by three distinct decision variables, poses a mixed-integer programming challenge and is acknowledged as NP-hard. Consequently, formulating scheduling policies for the offloaded tasks during time slot t becomes a complex task. To tackle this challenge, we transform $\mathcal{P}_k$ into a quadratically constrained quadratic programming (QCQP) problem.

To facilitate the procedures of solving $\mathcal{P}_k$, we rewrite its objective function as

$$\lambda_u = \sum\nolimits_{i\in\mathcal{I}_z} \lambda_{i,u} \triangleq G_u(t)C_u(t) + \Omega_u(t)O_u(t). \tag{23}$$

Incorporating (6) and (8), (23) can be further expressed as

$$\lambda_u = \sum\nolimits_{i\in\mathcal{I}_z} G_u(t)c_{i,u}(t) + \sum\nolimits_{i\in\mathcal{I}_o} \Omega_u(t)o_{i,u}(t)e_{i,u}(t), \tag{24}$$

where the second term can be reformulated as $\sum_{i\in\mathcal{I}_z} \Omega_u(t)o_{i,u}(t)e_{i,u}(t)$ with $\mathcal{I}_o \subset \mathcal{I}_z$, and some tasks may not be cached with $o_{i,u}(t) = 0$. Following this, we have $\lambda_{i,u}(t) = G_u(t)c_{i,u}(t) + \Omega_u(t)e_{i,u}o_{i,u}(t)$, which can be rewritten as follows with the help of (2)

$$\lambda_{i,u}(t) = \sum\nolimits_{u'\in\mathcal{U}\backslash\{u\}} m_{i,uu'}(t)\rho_{i,uu',1}(t) + a_{i,u}(t)\rho_{i,u,2}(t) + o_{i,u}(t)\rho_{i,u,3}(t), \tag{25}$$

where

$$\rho_{i,uu',1}(t) = \frac{G_u(t)e_{i,u}(t)}{r_{i,uu'}(t)}\left[\varpi_1 + \varpi_2 p_u(t)\right],$$

$$\rho_{i,u,2}(t) = \frac{G_u(t)e_{i,u}(t)}{w_{i,u}(t)}(\varpi_1 + \varpi_2\eta_c),$$

$$\rho_{i,u,3}(t) = G_u(t)\left(\varpi_1\frac{D}{\tau} + \varpi_2\eta_o\frac{D}{\tau}\right) + \Omega_u(t)e_{i,u}(t).$$

To enhance clarity, we introduce the variable vector $\boldsymbol{\phi}_{i,u}(t)$ and the auxiliary vector $\boldsymbol{\upsilon}_{i,u}(t)$, given as

$$\boldsymbol{\phi}_{i,u}(t) \triangleq [m_{i,u1}(t);\ldots;m_{i,uu'}(t);\ldots;m_{i,uU}(t); a_{i,u}(t);o_{i,u}(t)], \forall u' \in \mathcal{U}\backslash\{u\}, \forall i, \tag{26}$$

and

$$\boldsymbol{\upsilon}_{i,u}(t) \triangleq [\rho_{i,u1,1}(t);\ldots;\rho_{i,uu',1}(t);\ldots;\rho_{i,uU,1}(t); \rho_{i,u,2}(t);\rho_{i,u,3}(t)], \forall u' \in \mathcal{U}\backslash\{u\}, \forall i, \tag{27}$$

where $\boldsymbol{\phi}_{i,u}(t) \in \mathbb{R}^{U+1}$ and $\boldsymbol{\upsilon}_{i,u}(t) \in \mathbb{R}^{U+1}$.

Following this, we have

$$\boldsymbol{\phi}_i(t) = \left[\boldsymbol{\phi}_{i,1}(t);\ldots;\boldsymbol{\phi}_{i,u}(t);\ldots;\boldsymbol{\phi}_{i,U}(t)\right], \forall i \in \mathcal{I}_z, \tag{28}$$

$$\boldsymbol{\upsilon}_i(t) = \left[\boldsymbol{\upsilon}_{i,1}(t);\ldots;\boldsymbol{\upsilon}_{i,u}(t);\ldots;\boldsymbol{\upsilon}_{i,U}(t)\right], \forall i \in \mathcal{I}_z, \tag{29}$$

where $\boldsymbol{\phi}_i(t), \boldsymbol{\upsilon}_i(t) \in \mathbb{R}^{(U+1)U}, \forall t$. With the aid of these introduced vectors, the second term of (22a) is rewritten as

$$\sum\nolimits_{u\in\mathcal{U}} G_u(t)C_u(t) + \Omega_u(t)O_u(t) = \sum\nolimits_{i\in\mathcal{I}_z} \boldsymbol{\phi}_i^T(t)\boldsymbol{\upsilon}_i(t). \tag{30}$$

We then replace the integer constraints (9d)–(9f) with quadratic forms

$$a_{i,u}(t)(a_{i,u}(t) - 1) = 0, \forall i \in \mathcal{I}_z, u, \tag{31}$$

$$m_{i,uu'}(t)(m_{i,uu'}(t) - 1) = 0, \forall i \in \mathcal{I}_z, \forall u, \forall u' \in \mathcal{U}\backslash\{u\}, \tag{32}$$

$$o_{i,u}(t)(o_{i,u}(t) - 1) = 0, \forall i \in \mathcal{I}_z, u, \tag{33}$$

and reformulate these constraints as

$$\boldsymbol{\phi}_i^T(t)\mathrm{diag}(\boldsymbol{\chi}_n)\boldsymbol{\phi}_i - \boldsymbol{\chi}_n^T\boldsymbol{\phi}_i = 0, \forall i \in \mathcal{I}_z, \tag{34}$$

where $\boldsymbol{\chi}_n$ is a $(U+1)U \times 1$ standard unit vector with the n-th entry being 1, and $n \in \mathcal{N}$ with $\mathcal{N} = \{1, 2, \ldots, (U+1)U\}$. Subsequently, we transform the sum-to-one constraint (9h) into the following constraint

$$\boldsymbol{\phi}_i^T(t)\boldsymbol{\chi}_u^e(t) = 1, \forall i \in \mathcal{I}_z, \forall u, \tag{35}$$

where $\boldsymbol{\chi}_u^e(t) \in \mathbb{R}^{(U+1)U}$, and

$$\boldsymbol{\chi}_u^e(t) \triangleq \left[\underbrace{\mathbf{0}_{U+1};\ldots;\mathbf{0}_{U+1}}_{u-1};\mathbf{1}_{U+1};\underbrace{\mathbf{0}_{U+1};\ldots}_{U-(u-2)}\right], \quad (36)$$

where $\mathbf{0}_{U+1},\ \mathbf{1}_{U+1} \in \mathbb{R}^{U+1}$.

Similarly, the auxiliary vector $\boldsymbol{\chi}_{i,u}^d(t)$ is given as

$$\boldsymbol{\chi}_{i,u}^d(t) \triangleq \left[\frac{e_{i,u}(t)}{r_{i,u1}(t)};\ldots;\frac{e_{i,u}(t)}{r_{i,uu'}(t)};\ldots;\frac{e_{i,u}(t)}{r_{i,uU}(t)};\frac{e_{i,u}(t)}{w_{i,u}(t)};0\right], \quad (37)$$

where $\boldsymbol{\chi}_{i,u}^d(t) \in \mathbb{R}^{U+1}$, and $u' \in \mathcal{U}\backslash\{u\}$, and thus we can further replace constraint (22b) with

$$\boldsymbol{\phi}_i^T(t)\boldsymbol{\chi}_u^d(t) \leq D_{i,u}^z(t),\ \forall i \in \mathcal{I}_z,\ \forall u, \quad (38)$$

where $D_{i,u}^z(t) = \frac{D}{\tau} - z_{i,u}(t)\frac{e_{i,u}(t)}{r_{i,u}(t)}$, and $\boldsymbol{\chi}_u^d(t)$ is given as

$$\boldsymbol{\chi}_u^d(t) = \left[\underbrace{\mathbf{0}_{U+1};\ldots;\mathbf{0}_{U+1}}_{u-1};\boldsymbol{\chi}_{i,u}^d(t);\underbrace{\mathbf{0}_{U+1};\ldots}_{U-(u-2)}\right], \quad (39)$$

and $\boldsymbol{\chi}_u^d(t) \in \mathbb{R}^{(U+1)U}$.

Moreover, another auxiliary vector $\boldsymbol{\chi}_{i,u}^a(t)$ is introduced as

$$\boldsymbol{\chi}_{i,u}^a(t) \triangleq [\mathbf{0}_{U-1}; w_{i,u}(t); 0],\ \forall i \in \mathcal{I}_z,\ \forall u, \quad (40)$$

where $\mathbf{0}_{U-1} \in \mathbb{R}^{U-1}$ and $\boldsymbol{\chi}_{i,u}^a(t) \in \mathbb{R}^{U+1}$. Thus, we have

$$\boldsymbol{\chi}_u^a(t) = \left[\underbrace{\mathbf{0}_{U+1};\ldots;\mathbf{0}_{U+1}}_{u-1};\boldsymbol{\chi}_{i,u}^a(t);\underbrace{\mathbf{0}_{U+1};\ldots}_{U-(u-2)}\right], \quad (41)$$

where $\boldsymbol{\chi}_u^a(t) \in \mathbb{R}^{(U+1)U}$. Following this, constraint (9l) can be expressed as

$$\sum\nolimits_{i\in\mathcal{I}_z} \boldsymbol{\phi}_i^T(t)\boldsymbol{\chi}_u^a(t) \leq W_u^a(t),\ \forall u. \quad (42)$$

Introducing the further definition $\boldsymbol{\kappa}_i \triangleq [\boldsymbol{\phi}_i; 1] \in \mathbb{R}^{U(U+1)+1}$, in conjunction with the matrix expressions mentioned earlier, allows for a simplification of the problem. With some mathematic manipulations, we can rewrite $\mathcal{P}_k'$ into the following equivalent homogeneous separable QCQP formulation, given as

$$\mathcal{P}_k': \min_{\boldsymbol{\kappa}} \sum\nolimits_{i\in\mathcal{I}_z} \boldsymbol{\kappa}_i^T \boldsymbol{J}_i \boldsymbol{\kappa}_i \quad (43a)$$

$$\text{s.t. } \boldsymbol{\kappa}_i^T \boldsymbol{J}_n^1 \boldsymbol{\kappa}_i = 0,\ \forall i \in \mathcal{I}_z, \quad (43b)$$

$$\boldsymbol{\kappa}_i^T \boldsymbol{J}_u^e \boldsymbol{\kappa}_i = 1,\ \forall i \in \mathcal{I}_z, \forall u, \quad (43c)$$

$$\sum\nolimits_{i\in\mathcal{I}_z} \boldsymbol{\kappa}_i^T \boldsymbol{J}_u^d \boldsymbol{\kappa}_i \leq D_{i,u}^z(t),\ \forall i \in \mathcal{I}_z, \forall u, \quad (43d)$$

$$\sum\nolimits_{i\in\mathcal{I}_z} \boldsymbol{\kappa}_i^T \boldsymbol{J}_u^a \boldsymbol{\kappa}_i \leq W_u^a(t),\ \forall u, \quad (43e)$$

$$\boldsymbol{K}_i \succeq 0,\ \forall i \in \mathcal{I}_z \cup \{0\}, \forall u \in \mathcal{U} \cup \{0\}, \quad (43f)$$

where

$$\boldsymbol{J}_i \triangleq \begin{bmatrix} \mathbf{0} & \frac{1}{2}\boldsymbol{v}_i(t) \\ \frac{1}{2}\boldsymbol{v}_i^T(t) & 0 \end{bmatrix}, \boldsymbol{J}_n^1 \triangleq \begin{bmatrix} \text{diag}(\boldsymbol{\chi}_n) & -\frac{1}{2}\boldsymbol{\chi}_n \\ -\frac{1}{2}\boldsymbol{\chi}_n^T & 0 \end{bmatrix}, \forall n,$$

$$\boldsymbol{J}_u^e \triangleq \begin{bmatrix} \mathbf{0} & \frac{1}{2}\boldsymbol{\chi}_u^e \\ \frac{1}{2}(\boldsymbol{\chi}_u^e)^T & 0 \end{bmatrix}, \boldsymbol{J}_u^d \triangleq \begin{bmatrix} \mathbf{0} & \frac{1}{2}\boldsymbol{\chi}_u^d \\ \frac{1}{2}(\boldsymbol{\chi}_u^d)^T & 0 \end{bmatrix}, \forall u,$$

$$\boldsymbol{J}_u^a \triangleq \begin{bmatrix} \mathbf{0} & \frac{1}{2}\boldsymbol{\chi}_u^a \\ \frac{1}{2}(\boldsymbol{\chi}_u^a)^T & 0 \end{bmatrix}, \forall u.$$

Upon scrutinizing $\mathcal{P}_k'$, it becomes apparent that each constraint in $\mathcal{P}_k$ is directly associated with a specific matrix representation in $\mathcal{P}_k'$. It is essential to emphasize that all variables under consideration in this study are inherently positive, thereby naturally satisfying the constraint (43f). Consequently, $\mathcal{P}_k'$ is essentially equivalent to the original problem $\mathcal{P}_k$.

However, it is crucial to note that the optimization problem $\mathcal{P}_k'$ encapsulates a non-convex separable QCQP challenge, typically falling under the NP-hard category. To tackle this, we can employ a separable Semidefinite Relaxation (SDR) method, transforming $\mathcal{P}_k'$ into a separable semidefinite programming (SDP) problem. With the help of $\boldsymbol{K}_i \triangleq \boldsymbol{\kappa}_i\boldsymbol{\kappa}_i^T$, we have the following equivalent formulations for the left-hand side of equations in constraints of $\mathcal{P}_k'$, given as

$$\boldsymbol{\kappa}_i^T \boldsymbol{J} \boldsymbol{\kappa}_i = \text{Tr}(\boldsymbol{J}\boldsymbol{\kappa}_i\boldsymbol{\kappa}_i^T) = \text{Tr}(\boldsymbol{J}\boldsymbol{K}_i),\ \forall i \in \mathcal{I}_z, \quad (44)$$

$$\text{rank}(\boldsymbol{K}_i) = 1,\ \forall i \in \mathcal{I}_z, \quad (45)$$

where $\boldsymbol{K}_i \in \mathbb{R}^{[U(U+1)+1]\times[U(U+1)+1]}$. Moreover, $\boldsymbol{J}$ can be $\boldsymbol{J}_i$, $\boldsymbol{J}_n^1$, $\boldsymbol{J}_u^e$, $\boldsymbol{J}_u^d$, and $\boldsymbol{J}_u^a$. By omitting the rank-one constraint $\text{rank}(\boldsymbol{K}_i) = 1$, we transform $\mathcal{P}_k'$ into the following separable SDP problem

$$\mathcal{P}_k'': \min_{\mathcal{K}} \sum\nolimits_{i\in\mathcal{I}_z} \text{Tr}(\boldsymbol{J}_i\boldsymbol{K}_i) \quad (46a)$$

$$\text{s.t. (43f)},$$

$$\text{Tr}(\boldsymbol{J}_n^1\boldsymbol{K}_i) = 0,\ \forall i \in \mathcal{I}_z, \quad (46b)$$

$$\text{Tr}(\boldsymbol{J}_u^e\boldsymbol{K}_i) = 1,\ \forall i \in \mathcal{I}_z, \forall u, \quad (46c)$$

$$\text{Tr}(\boldsymbol{J}_u^d\boldsymbol{K}_i) \leq D_{i,u}^z(t),\ \forall i \in \mathcal{I}_z, \forall u, \quad (46d)$$

$$\sum\nolimits_{i\in\mathcal{I}_z} \text{Tr}(\boldsymbol{J}_u^a\boldsymbol{K}_i) \leq W_u^a(t),\ \forall u, \quad (46e)$$

$$\boldsymbol{K}_i((U+1)U+1, (U+1)U+1) = 1, \forall i \in \mathcal{I}_z, \quad (46f)$$

where $\mathcal{K} = \{\boldsymbol{K}_i\} \in \mathbb{R}^{I_z\times[U(U+1)+1]\times[U(U+1)+1]}$. Constraint (46f) comes from the equation $\boldsymbol{K}_i \triangleq \boldsymbol{\kappa}_i\boldsymbol{\kappa}_i^T$, $\forall i \in \mathcal{I}_z$. It is worth noting that $\mathcal{P}_k''$ can be efficiently addressed using solvers such as CVX and MOSEK, and we denote $\mathcal{K}^\star$ as the optimal solution of $\mathcal{P}_k''$. However, this solution may not serve as a feasible solution for the original optimization problem $\mathcal{P}_k$. In this scenario, the task at hand is to recover the scheduling decision variable $\boldsymbol{\phi}_i$ from $\boldsymbol{K}_i^\star$, $\forall i \in \mathcal{I}_z$. To achieve this, we propose a method based on the high probability selection to determine our binary decision variable.

To obtain binary decision variables through randomization, we adopt a methodological approach as outlined in [40] to achieve an integer solution. Upon examining $\boldsymbol{K}_i$, it becomes apparent that its last row adheres to the condition $\boldsymbol{K}_i((U+1)U+1, n) = \boldsymbol{\kappa}_i(n), \forall i \in \mathcal{I}_z$. Consequently, we can leverage the values of $\boldsymbol{K}_i{}^\star((U+1)U+1, n)$ to reconstruct the binary variables

$\boldsymbol{\kappa}_i(n), \forall n$. Regarding the values of $\boldsymbol{K}_i{}^\star((U+1)U+1, n)$, the following lemma is presented as

Lemma 2: For the optimal solution $\mathcal{K}^\star$ of $\mathcal{P}_k''$, the values $\boldsymbol{K}_i{}^\star((U+1)U+1, n)$ lie within the interval [0,1], $\forall i \in \mathcal{I}_z$, $\forall n$.

Proof: The proof of Lemma 2 is provided in Section VI of the supplementary material, available online, due to space constraints. □

Building on *Lemma 2*, we employ a probabilistic mapping method to ascertain $\boldsymbol{\phi}_i(t)$. The values of $\boldsymbol{K}_i{}^\star((U+1)U+1, :)$ are interpreted as the probability that $\mathbf{1}^T\boldsymbol{\phi}_i(t) = 1$. Define $\boldsymbol{\nu}_{i,u}(t)$ as

$$\boldsymbol{\nu}_{i,u}(t) \triangleq \left[\nu^m_{i,u1}(t); \nu^m_{i,u2}(t); \ldots; \nu^m_{i,uu'}(t); \ldots; \nu^m_{i,uU}(t); \nu^a_{i,u}(t); \nu^o_{i,u}(t)\right], \; u' \in \mathcal{U}\backslash\{u\} \quad (47)$$

where $\boldsymbol{\nu}_{i,u} \in \mathbb{R}^{U+1}$, and

$$\boldsymbol{\nu}_i(t) = \left[\boldsymbol{\nu}_{i,1}(t); \ldots; \boldsymbol{\nu}_{i,u}(t); \ldots; \boldsymbol{\nu}_{i,U}(t)\right], \forall i \in \mathcal{I}_z, \quad (48)$$

where $\boldsymbol{\nu}_i(t) \in \mathbb{R}^{(U+1)U}$. Following this, we have

$$\boldsymbol{\nu}_i(t) \triangleq [\boldsymbol{K}_i{}^\star((U+1)U+1, 1); \boldsymbol{K}_i{}^\star((U+1)U+1, 2); \cdots; \boldsymbol{K}_i{}^\star((U+1)U+1, (U+1)U-2); \boldsymbol{K}_i{}^\star((U+1)U+1, (U+1)U-1); \boldsymbol{K}_i{}^\star((U+1)U+1, (U+1)U)]. \quad (49)$$

We reconstruct $\boldsymbol{\phi}_{i,u}(t)$ using $\boldsymbol{\nu}_{i,u}(t)$ as marginal probabilities, ensuring compliance with constraint (9h). This gives rise to our proposed probabilistic randomization method, outlined as follows. To ensure coherence between scheduling decision and offloading strategy, we set the probability of scheduling decision $\boldsymbol{\Xi}_{i,u}$ for user i's task as

$$\begin{aligned}
\Xi^m_{i,u1} &= z_{i,u}\nu^m_{i,u1}(1-\nu^m_{i,u2})\cdots(1-\nu^m_{i,uU})(1-\nu^a_{i,u})(1-\nu^o_{i,u}),\\
\Xi^m_{i,u2} &= z_{i,u}(1-\nu^m_{i,u1})\nu^m_{i,u2}\cdots(1-\nu^m_{i,uU})(1-\nu^a_{i,u})(1-\nu^o_{i,u}),\\
&\vdots\\
\Xi^m_{i,uU} &= z_{i,u}(1-\nu^m_{i,u1})(1-\nu^m_{i,u2})\cdots\nu^m_{i,uU}(1-\nu^a_{i,u})(1-\nu^o_{i,u}),\\
\Xi^a_{i,u} &= z_{i,u}(1-\nu^m_{i,u1})(1-\nu^m_{i,u2})\cdots(1-\nu^m_{i,uU})\nu^a_{i,u}(1-\nu^o_{i,u}),\\
\Xi^o_{i,u} &= z_{i,u}(1-\nu^m_{i,u1})(1-\nu^m_{i,u2})\cdots(1-\nu^m_{i,uU})(1-\nu^a_{i,u})\nu^o_{i,u},
\end{aligned} \quad (50)$$

where the probability $\boldsymbol{\Xi}_{i,u}$ is non-zero only when $z_{i,u} = 1$, signifying that the related scheduling decision is activated. Specifically, the scheduling decision is made by UAV u only if user i offloads its task to UAV u. Following this condition, we normalize the probability associated with the corresponding scheduling decision as $P^{sd}_{i,su} = \frac{\Xi^{sd}_{i,su}}{\Xi^m_{i,uu'}+\Xi^a_{i,u}+\Xi^o_{i,u}}$, $sd \in \{m, a, o\}$, $su \in \{uu', u\}$, satisfying $P^m_{i,u1} + P^m_{i,u2} + \cdots + P^m_{i,uU} + P^a_{i,u} + P^o_{i,u} = 1$.

Notice that we apply the probability reduction method, setting the element with the highest probability $P^{sd}_{i,su}$ to 1 to guide scheduling decisions. Specifically, if $o_{i,u}(t) = 1$, it signifies that user i's task is directed to the caching queue $\mathcal{I}_o$. Likewise, if $m_{i,uu'}(t) = 1$, the task is assigned to both the waiting queue $\mathcal{I}_z$ and the pending forwarding queue $\mathcal{I}^u_m$ at UAV u. *For a clear understanding, a numerical example detailing this process is provided in Section VI of the supplementary material, available online, due to space constraints.*

D. Optimizing Bandwidth Allocation for Migration

Given the recently established scheduling policies, allocating bandwidth to migrate redundant tasks to idle UAVs for subsequent processing within their latency requirements becomes paramount. This approach expands the system's service capacity. Consequently, the associated subproblem concerning bandwidth allocation can be formulated as follows

$$\mathcal{P}_b : \min_{\mathsf{B}} \sum\nolimits_{u\in\mathcal{U}} G_u(t)C_u(t) \quad \text{s.t. (9k), (22b).} \quad (51a)$$

which is a convex problem. While solvers like CVX and MOSEK can readily handle $\mathcal{P}_b$, we leverage the primal-dual method to gain deeper insights into the optimal bandwidth allocation $\mathsf{B}^\star$, drawing inspiration from the pioneering works [36], [41]. However, a direct transformation of the primal domain of $\mathcal{P}_b$ into the dual domain is not feasible due to the presence of $r_{i,uu'}(t)$ in the denominator, and its form renders the problem more intractable. Therefore, we introduce a new non-negative auxiliary variable $\boldsymbol{\varphi} \triangleq \{\boldsymbol{\varphi}_1; \ldots; \boldsymbol{\varphi}_u; \ldots; \boldsymbol{\varphi}_U\}$, where $\boldsymbol{\varphi}_u \triangleq \{\varphi_{i,uu'}\}, \forall u'$, transforming $\mathcal{P}_b$ into the following problem

$$\mathcal{P}_b' : \min_{\mathsf{B},\boldsymbol{\varphi}} \sum_{u\in\mathcal{U}} \tilde{G}_u(t) \sum_{i\in\mathcal{I}_z} \sum_{u'\in\mathcal{U}\backslash\{u\}} \frac{m_{i,uu'}(t)e_{i,u}(t)}{\varphi_{i,uu'}(t)} \quad (52a)$$

$$\text{s.t.} \sum_{u'\in\mathcal{U}\backslash\{u\}} \frac{m_{i,uu'}(t)e_{i,u}(t)}{\varphi_{i,uu'}(t)} \le D^a_{i,u}(t), \forall i \in \mathcal{I}^u_m, \forall u, \quad (52b)$$

$$\sum\nolimits_{i\in\mathcal{I}^u_m} b_{i,uu'}(t) \le 1, \forall u, u', \quad (52c)$$

$$0 < \varphi_{i,uu'}(t) \le r_{i,uu'}(t), \forall i \in \mathcal{I}^u_m, \forall u, u', \quad (52d)$$

where $\tilde{G}_u(t) = G_u(t)[\varpi_1 + \varpi_2 p_u(t)]$, and $D^a_{i,u}(t) = \frac{D}{\tau} - z_{i,u}(t)\frac{e_{i,u}(t)}{r_{i,u}(t)} - a_{i,u}(t)\frac{e_{i,u}(t)}{w_{i,u}(t)}$. The Lagrange function for this problem is given at the top of next page, where $\boldsymbol{\gamma} \triangleq \{\gamma_{iu}\}$, $\boldsymbol{\delta} \triangleq \{\delta_{uu'}\}$, and $\boldsymbol{\varsigma} \triangleq \{\varsigma_{i,uu'}\}$, $\forall u, u'$ are the non-negative Lagrange multipliers corresponding to the relevant constraints (53) shown at the bottom of this page.

Consider $\mathcal{B}$ as the set of all possible B satisfying the constraint $b_{i,uu'} > 0$, and let $\boldsymbol{\Phi}$ be the set of all possible $\boldsymbol{\varphi}$ satisfying the constraint $\varphi_{i,uu'} > 0$. The corresponding Lagrange dual function can be defined as

$$\mathcal{G}(\boldsymbol{\gamma}, \boldsymbol{\delta}, \boldsymbol{\varsigma}) = \min_{\mathsf{B}\in\mathcal{B}, \boldsymbol{\varphi}\in\boldsymbol{\Phi}} \mathcal{L}(\mathsf{B}, \boldsymbol{\varphi}, \boldsymbol{\gamma}, \boldsymbol{\delta}, \boldsymbol{\varsigma}), \quad (54)$$

and consequently, the dual problem can be formulated as

$$\max \mathcal{G}(\boldsymbol{\gamma}, \boldsymbol{\delta}, \boldsymbol{\varsigma}) \quad \text{s.t. } \boldsymbol{\delta} \succeq 0, \boldsymbol{\varsigma} \succeq 0, \boldsymbol{\gamma} \succeq 0. \quad (55a)$$

In the following, our objective is to determine the optimal bandwidth allocation $\mathsf{B}^\star$, given a set of auxiliary variables represented by $\boldsymbol{\varphi}$ and Lagrange multipliers $(\boldsymbol{\gamma}, \boldsymbol{\delta}, \boldsymbol{\varsigma})$. Initially, we seek to find the optimal B. Subsequently, we update the

Lagrange multipliers using the gradient descent method. Finally, we iterate through the process to update the auxiliary variables.

1) Bandwidth Allocation Optimization: Employing the Karush-kuhn-Tucker (KKT) conditions, the following condition is both necessary and sufficient for bandwidth allocation optimality, given as

$$\frac{\partial \mathcal{L}(\mathsf{B},\boldsymbol{\varphi},\boldsymbol{\gamma},\boldsymbol{\delta},\boldsymbol{\varsigma})}{\partial b_{i,uu'}(t)} = \delta_{uu'} - \varsigma_{i,uu'}\left[B\log_2\left(1+\frac{p_u(t)g^{\mathrm{LoS}}_{uu'}(t)}{N_0 b_{i,uu'}(t)B}\right) - \frac{p_u(t)g^{\mathrm{LoS}}_{uu'}(t)}{N_0 b_{i,uu'}(t)\left(1+\frac{p_u(t)g^{\mathrm{LoS}}_{uu'}(t)}{N_0 b_{i,uu'}(t)B}\right)\ln 2}\right] = 0, \quad (56)$$

which can be solved by MATLAB. Due to the inherent complexity in obtaining the closed-form solution for $b_{i,uu'}(t)$, we propose an alternative approach to derive such solutions for all i, u, u' by making the assumption of a high Signal-to-Noise Ratio (SNR). This assumption allows us to explore and gain additional insights into the interplay between bandwidth allocation and various other variables.

Proposition 1: In the high SNR regime, the optimal bandwidth allocation can be directly inferred from (56) under the assumption $\left(1+\frac{p_u(t)g^{\mathrm{LoS}}_{uu'}(t)}{N_0 b_{i,uu'}(t)B}\right) \approx \frac{p_u(t)g^{\mathrm{LoS}}_{uu'}(t)}{N_0 b_{i,uu'}(t)B}$, yielding

$$b^{\star}_{i,uu'}(t) = \frac{p_u(t)g^{\mathrm{LoS}}_{uu'}(t)}{N_0 B}\cdot 2^{-\frac{\delta_{uu'}}{\varsigma_{i,uu'}B}-\frac{1}{\ln 2}}. \quad (57)$$

Remark 1: The high SNR approximation, a widely employed technique in the literature [36], [42], has been previously highlighted. It is evident that in such scenarios, UAV u would logically allocate a greater bandwidth $b^{\star}_{i,uu'}(t)$ for user i to facilitate the seamless migration of its data to UAV u' in cases where the inter-UAV channel quality is superior.

Finding a closed-form solution for the optimal $b^{\star}_{i,uu'}(t)$ from (56) is challenging. Fortunately, we can resort to the following proposition to obtain $b^{\star}_{i,uu'}(t)$.

Proposition 2: $\mathcal{L}$ is a convex function of $b_{i,uu'}(t)$.

Proof: The analysis and proof of this proposition are provided in Section VII of the supplementary material, available online, due to space constraints. □

Since $\mathcal{L}$ is a convex function of $b_{i,uu'}$, we can determine the optimal bandwidth allocation using a bisection method [36]. Due to the convex nature of $\mathcal{L}$ and the monotonically increasing behavior of $\frac{\partial\mathcal{L}}{\partial b_{i,uu'}(t)}$ with respect to $b_{i,uu'}(t)$, the bisection method proves effective within the range $0 \le b_{i,uu'}(t) \le 1$ for finding the optimal solution. Algorithm 5 provides a detailed outline of this approach.

2) Auxiliary Variable Update: Subsequently, with the newly determined $\mathsf{B}^{\star}$ in the previous parts, we try to find the optimal auxiliary variable $\varphi_{i,uu'}{}^{\star}$, which can be solved via the following optimization problem

Algorithm 5: Bisection Method.

```
Input: Given task offloading strategies $\boldsymbol{Z}$, scheduling
  policies $\{\boldsymbol{A}, \mathsf{M}, \boldsymbol{O}\}$, and the trajectories of UAVs $\boldsymbol{Q}$.
Output: Bandwidth allocation for migration $\mathsf{B}^{\star}$.
1: for each $u \in \mathcal{U}$ do
2:   for each $i \in \mathcal{I}^u_m$ do
3:     Initialize $b^{\mathrm{LB}}_{i,uu'} = 0$ and $b^{\mathrm{UB}}_{i,uu'} = 1$;
4:     repeat
5:       Set $b_{i,uu'}(t) = \frac{1}{2}(b^{\mathrm{LB}}_{i,uu'} + b^{\mathrm{UB}}_{i,uu'})$;
6:       Compute $\frac{\partial\mathcal{L}}{\partial b_{i,uu'}(t)}$ according to (56);
7:       if $\frac{\partial\mathcal{L}}{\partial b_{i,uu'}(t)} > 0$ then
8:         Set $b^{\mathrm{UB}}_{i,uu'} = b_{i,uu'}(t)$;
9:       else
10:        Set $b^{\mathrm{LB}}_{i,uu'} = b_{i,uu'}(t)$.
11:      end if
12:    until $\|\frac{\partial\mathcal{L}}{\partial b_{i,uu'}(t)}\| \le \psi_1$.
13:  end for
14: end for
15: return the optimal $b^{\star}_{i,uu'}$.
```

$$\mathcal{P}''_b : \min_{\boldsymbol{\varphi}} \sum_{u\in\mathcal{U}} \tilde{G}_u(t) \sum_{i\in\mathcal{I}_z}\sum_{u'\in\mathcal{U}\setminus\{u\}} \frac{m_{i,uu'}(t)e_{i,u}(t)}{\varphi_{i,uu'}(t)} \quad (58a)$$

$$\text{s.t. (52b)},$$

$$0 < \varphi_{i,uu'}(t) \le r^{\star}_{i,uu'}(t), \forall i \in \mathcal{I}^u_m, \forall u, u', \quad (58b)$$

where

$$r^{\star}_{i,uu'}(t) = b^{\star}_{i,uu'}(t)B\log_2\left(1+\frac{p_u(t)g^{\mathrm{LoS}}_{uu'}(t)}{N_0 b^{\star}_{i,uu'}(t)B}\right). \quad (59)$$

It is evident that $\mathcal{P}''_b$ can be further decomposed into UI^u_m independent subproblems, each corresponding to a User-UAV pair, $\forall i \in \mathcal{I}^u_m$, $\forall u$, given by

$$\mathcal{P}'''_b : \min_{\boldsymbol{\varphi}} \tilde{G}_u(t)\sum\nolimits_{u'\in\mathcal{U}\setminus\{u\}} \frac{m_{i,uu'}(t)e_{i,u}(t)}{\varphi_{i,uu'}(t)} \quad (60a)$$

$$\text{s.t.} \sum\nolimits_{u'\in\mathcal{U}\setminus\{u\}} \frac{m_{i,uu'}(t)e_{i,u}(t)}{\varphi_{i,uu'}(t)} \le D^a_{i,u}(t), \quad (60b)$$

$$0 < \varphi_{i,uu'}(t) \le r^{\star}_{i,uu'}(t), \forall u', \quad (60c)$$

where the optimal auxiliary variable $\varphi^{\star}_{i,uu'}(t)$ can be obtained by the following theorem.

$$\mathcal{L}(\mathsf{B},\boldsymbol{\varphi},\boldsymbol{\gamma},\boldsymbol{\delta},\boldsymbol{\varsigma}) = \sum\nolimits_{u\in\mathcal{U}}\sum\nolimits_{i\in\mathcal{I}^u_m}\left[\tilde{G}_u(t)+\gamma_{i,u}\right]\sum\nolimits_{u'\in\mathcal{U}\setminus\{u\}}\frac{m_{i,uu'}(t)e_{i,u}(t)}{\varphi_{i,uu'}(t)} - \sum\nolimits_{u\in\mathcal{U}}\sum\nolimits_{i\in\mathcal{I}_z}\gamma_{i,u}D^a_{i,u}(t)$$
$$+\sum\nolimits_{u\in\mathcal{U}}\sum\nolimits_{u'\in\mathcal{U}\setminus\{u\}}\delta_{uu'}\left[\sum\nolimits_{i\in\mathcal{I}^u_m} b_{i,uu'}(t)-1\right] + \sum\nolimits_{u\in\mathcal{U}}\sum\nolimits_{u'\in\mathcal{U}\setminus\{u\}}\sum\nolimits_{i\in\mathcal{I}^u_m}\varsigma_{i,uu'}\left[\varphi_{i,uu'}(t)-r_{i,uu'}(t)\right]. \quad (53)$$

Algorithm 6: Optimizing Bandwidth Allocation for Migration.

```
Initialization: Given task offloading strategies $\boldsymbol{Z}$, scheduling policies $\{\boldsymbol{A}, \mathsf{M}, \boldsymbol{O}\}$, and the trajectories of UAVs $\boldsymbol{Q}$.
1: repeat (from 1 to 7)
2:   repeat (from 2 to 5)
3:     Execute Algorithm 5 to determine $\mathsf{B}^\star$;
4:     Update $\boldsymbol{\varphi}^\star$ based on (61);
5:   until Lagrange function converges.
6:   Update $\boldsymbol{\gamma}$, $\boldsymbol{\delta}$, and $\boldsymbol{\varsigma}$;
7: until $\boldsymbol{\gamma}$, $\boldsymbol{\delta}$, $\boldsymbol{\varsigma}$ converge.
```

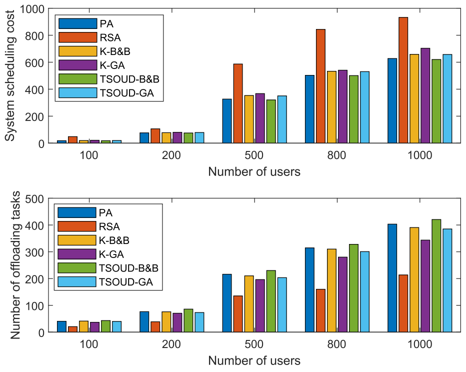


Fig. 3. Comparative analysis of different algorithms on scheduling costs and system throughput.

Theorem 1: With the obtained $b^\star_{i,uu'}(t)$, $\forall u'$, the optimal auxiliary variable $\varphi^\star_{i,uu'}(t)$ can be given as

$$\varphi^\star_{i,uu'}(t) = \begin{cases} \varphi^o_{i,uu'}(t), & \text{if } 0 < \varphi^o_{i,uu'}(t) \le r^\star_{i,uu'}(t), \\ r^\star_{i,uu'}(t), & \text{otherwise,} \end{cases} \quad (61)$$

Proof: The analysis and proof of this theorem are provided in Section VII of the supplementary material, available online, due to space constraints. □

3) Lagrange Multipliers Update: In this part, with $\mathsf{B}^\star$ and $\boldsymbol{\varphi}^\star$ acquired, we proceed to address the dual problem presented in (55). This problem, being convex in nature, allows for the updating of $\boldsymbol{\gamma}$, $\boldsymbol{\delta}$, and $\boldsymbol{\varsigma}$ through the subgradient method.

VI. Comprehensive Algorithm Strategy

In this section, we employ the Lyapunov framework to address individual time slot challenges, culminating in the development of the Proposed Algorithm (PA), as delineated in Algorithm 7.

The PA focuses on UAV deployment and scheduling, incorporating the TSOUD deployment algorithm and QCQP & SDR scheduling decision method. We crafted a series of experimental scenarios to evaluate these components against the PA. Results are depicted in Fig. 3 and Table I. *For a detailed discussion, constrained by space, please see Section VIII of the supplementary material, available online.*

TABLE I
Running Time (s)

Number of users	PA	RSA	K-B&B	K-GA	TSOUD -B&B	TSOUD -GA
100	0.22	0.28	0.35	0.28	0.36	0.24
200	0.64	0.28	0.77	0.65	0.79	0.56
500	1.50	0.31	7.48	4.01	6.97	3.24
800	2.81	0.33	12.08	7.12	13.97	7.39
1000	3.72	0.36	18.85	9.34	18.54	8.30

TABLE II
Experiment Parameters

System parameters	Values
Offloading bandwidth $B_i(t)$	1 MHz
Total bandwidth during task migration B	20 MHz
The duration of system	200 s
The number of time slots	200
Transmission power at UAV p_u	0.5 W
Transmission power at user p_i	0.2 W
Noise power spectrum density N_0	-180 dBm/Hz
Channel gain at $d_0 = 1$m	60 dB
Path loss exponent $\tilde{\iota}$	2.3
NLoS attenuation k	0.2
NLoS environmental constants α, β	11.95, 0.14
UAV maximum coverage distance $\tilde{r}_u$	180 m
UAV altitude h_u	60 m
UAV computing coefficient η_c	10^{-12}
UAV caching coefficient η_o	10^{-15}
UAV computing resource W_u	1 GHz
UAV average long-term cost budget	20

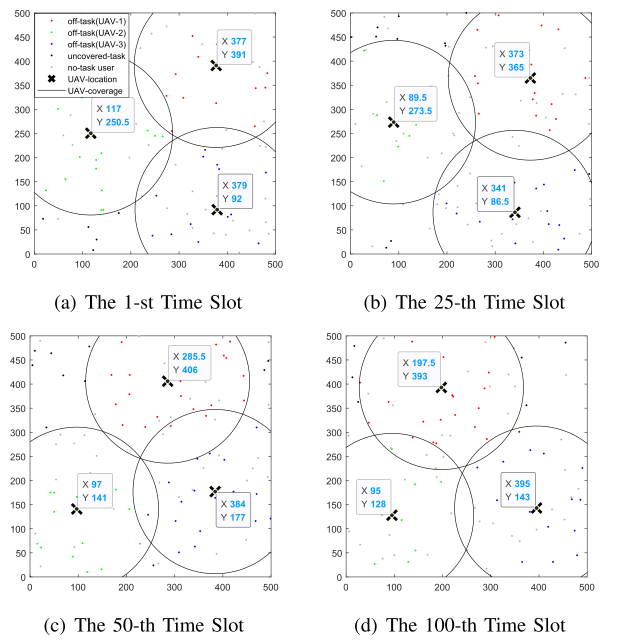
VII. Experiment and Discussion

In this section, we conduct simulation experiments to assess the feasibility of the scenario and the efficacy of the proposed method.

It is important to note that the experiments were conducted on a desktop PC equipped with an Intel(R) Core(TM) i7-12700F 2.1GHz processor and 32GB of RAM. The execution environments used for these experiments included MATLAB R2021a with CVX and YALMIP, and Pycharm 2023.2.3 with Pytorch 1.12.0. We deploy three UAVs to serve 100 mobile users within a designated 500 $m \times 500$ m area. User movements are random, with speeds ranging from 0.5 m/s to 1 m/s , and tasks, varying between 1 and 3 Mbits in size, are generated at each time slot following a Bernoulli distribution. Each task requires 1 to 3 cycles per bit, with a fixed computational allocation of 1 to 3MHz. The UAV deployment adheres to the methodologies described in Section V, ensuring consistent simulation parameters across different scales, as detailed in Table II. The value of η_c is determined based on the energy consumption factor [30]. Additionally, the scheduling cost weight, c_u, balances delay and energy consumption at a ratio of 4:6, configured as $\varpi_1 = 0.4, \varpi_2 = 0.6$. The magnitude of these weights dynamically adjusts in response to computational, migration, and caching decisions: $\varpi_1 = 10^0, \varpi_2 = 10^1$ for computation; $\varpi_1 = 10^0, \varpi_2 = 10^{12}$ for migration; and $\varpi_1 = 10^0, \varpi_2 = 10^{14}$ for caching.

A. UAV Deployment

As depicted in Fig. 4, users randomly generate tasks in each time slot, and the UAVs cater to these tasks within their service ranges. To enhance task service coverage, UAVs adjust their



(a) The 1-st Time Slot (b) The 25-th Time Slot

(c) The 50-th Time Slot (d) The 100-th Time Slot

Fig. 4. UAV deployment and task offloading across various time slots.

positions following user movements. Given the users' limited speed, their relative positions within a span of 100 time slots do not exceed a 100m square area, indicating that each UAV continues to serve most of its initial users during movement. Additionally, the UAVs' speed allows them to reach the deployment locations determined by the algorithm for each time slot.

Furthermore, to assess the impact of UAV trajectory optimization on system objectives, we introduce two alternative schemes for comparison with our proposed algorithm:

- *K-means++ Location:* This method employs the K-means++ algorithm to cluster user locations for each time slot. The resulting cluster centers serve as the deployment locations for UAVs, defining their trajectories.
- *Random Location:* This approach sets the UAVs' initial positions and assigns their movement speeds randomly between 10m/s and 20m/s. Consequently, this method determines the UAVs' trajectories and deployment positions for each time slot.

According to Fig. 5, our proposed algorithm boosts system throughput by 10%-45% and decreases scheduling costs by 15%-30%, surpassing the K-means++ and random deployment methods. *Furthermore, to examine how variations in user characteristics and network conditions affect system performance, we conducted a series of experiments, detailed in Section IX of the supplementary material, available online, due to space constraints.*

B. Task Scheduling

In our experiments, we analyze task dynamics for three users over eleven discrete time slots, as illustrated in Fig. 6. User-1 initiates five tasks throughout this period, all successfully offloaded and processed. The tasks from the first time slot are offloaded to the third UAV, while those from the second time slot are uploaded to the second UAV and processed in the third slot.

Algorithm 7: Proposed Algorithm.

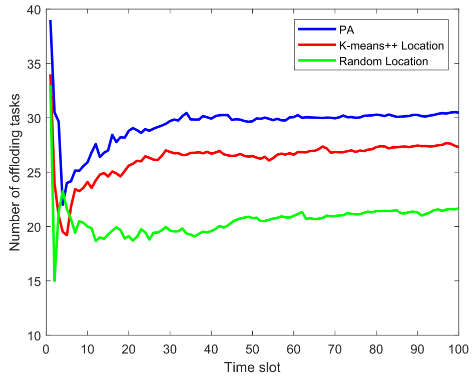
```
Input: Initial queue status $G_u(0)$ and $\Omega_u(0)$, we set
  $L(t) = 0$, and denote $L_{\max}$ as the maximum number of
  iterations.
Output: Solutions $\{\boldsymbol{Q}, \boldsymbol{S}, \boldsymbol{Z}, \boldsymbol{A}, \mathsf{M}, \boldsymbol{O}, \mathsf{B}\}$.
1: Initialize the number of time slot, user count, UAV count;
2: for each time slot $t$ do
3:    Generate random user positions;
4:    Generate task requests for users;
5:    Execute Algorithm 1 update Lyapunov queue $G_u(t)$
      and $\Omega_u(t)$;
6:    Execute Algorithm 2 for UAV deployment $\boldsymbol{Q}$;
7:    Execute Algorithm 3 to determine $\boldsymbol{S}$;
8:    repeat (from 8 to 15)
9:       Execute Algorithm 4 to optimize $\boldsymbol{Z}$ in the $\mathcal{I}_z(t)$;
10:      Utilize QCQP method to optimize $\boldsymbol{A}, \mathsf{M}, \boldsymbol{O}$ for
         offloaded tasks;
11:      for each user $i \in \mathcal{I}_m^u$ do
12:         Determine transmission bandwidth between UAVs
            using Algorithm 6;
13:      end for
14:      $L(t) = L(t) + 1$;
15:   until the difference of successive values of the
      objective function is less than $\epsilon_1$ or $L(t) > L_{\max}$;
16:   Store the feasible solution for the current time slot;
17: end for
18: returnFeasible solutions for all time slots.
```

Tasks initially assigned to the third UAV in the fourth time slot are later migrated to the second UAV for processing in the fifth slot. Tasks directed to the third UAV in the seventh time slot are subsequently moved to the second UAV and, due to processing constraints, further migrated to the first UAV in the eighth time slot. These tasks are cached in the ninth time slot and processed in the eleventh slot. *For clarity and due to space limitations, a detailed table of task dynamics is included in Section IX of the supplementary material, available online.*

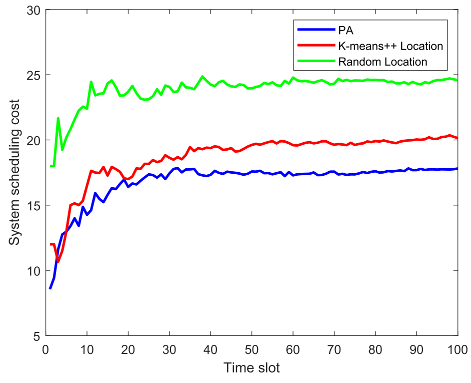
Fig. 7 offer insights into the system's efficiency regarding task association, offloading, migration, computing, and caching. The task completion rate initially increases, then decreases, and finally stabilizes, reflecting the system's early resource abundance, which prevents task queuing. However, the necessity to process cached tasks consumes computational resources, reducing the processing capacity in subsequent time slots. After queue backlog occurs, the system adjusts its scheduling decisions from the 40th time slot to maintain stability. The baseline in the figure represents the average task processing rate by UAVs dedicated solely to computation, without migration or caching. The analysis reveals that our proposed approach significantly boosts the system's long-term average task access frequency.

C. Lyapunov Framework

Fig. 8 demonstrates the inverse relationship between the Lyapunov penalty factor V and system throughput: as V increases, system throughput declines and the queue backlog grows, underscoring V's role as a regulatory mechanism in the



(a) System Throughput



(b) System Scheduling Cost

Fig. 5. Comparative analysis of different UAV deployment methods on scheduling costs and system throughput.

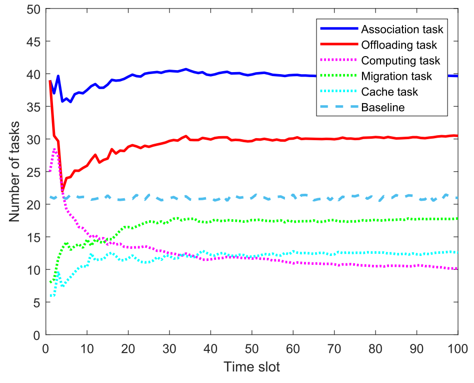
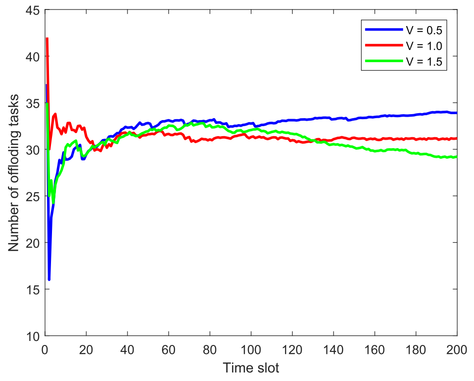
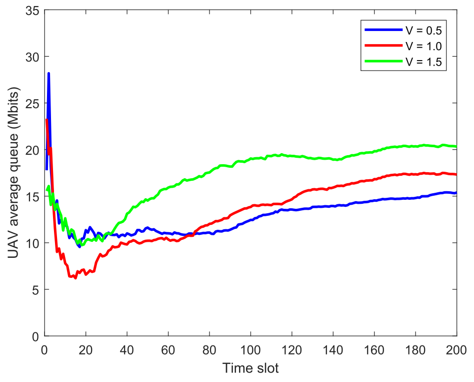


Fig. 7. Task association and scheduling volume.



(a) Impact of Lyapunov Penalty Factors (V) on System Throughput



(b) Impact of Lyapunov Penalty Factors (V) on System Queue Backlog

Fig. 8. Comparative analysis of the impact of Lyapunov penalty factors (V) on system throughput and system queue backlog.

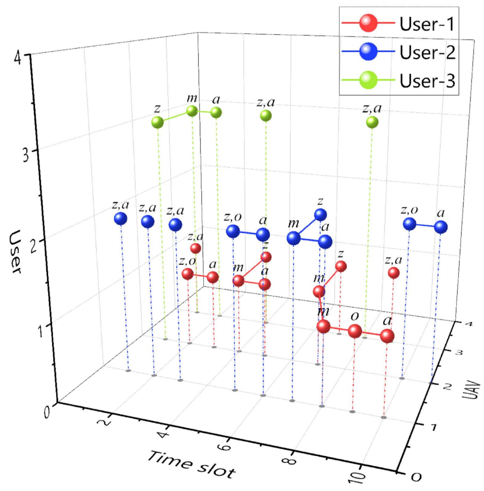


Fig. 6. The state of user tasks.

objective function to balance system accessibility with queue congestion. The relationship between system access and queue backlog stabilizes within the range $[O(1/V), O(V)]$, typically after approximately 100 time slots, indicating a harmonization in system dynamics. Additionally, the Lyapunov online optimization framework is well-suited for maintaining queue stability. *Due to space constraints, corresponding experimental results are detailed in Section X of the supplementary material, available online.*

To evaluate the stability and effectiveness of the Lyapunov framework for online optimization problems, we compared it with reinforcement learning frameworks such as DDPG, A2C, and PPO. DDPG is utilized as an offline learning framework, whereas A2C and PPO are applied in online settings. Nevertheless, certain limitations were observed due to the intrinsic properties of these reinforcement learning frameworks during experimentation, summarized as follows.

- *Dimension Fixing*: The reinforcement learning framework imposes a fixed output dimension per total time slot, where each decision variable remains static regardless of queue dynamics. Thus, each queue requires a preset maximum dimension of $1 \times I$, resulting in high space complexity.

- *Multi-hop Migration Limitations*: The fixed dimensions of the action space in our reinforcement learning framework, particularly the $1 \times I$ migration queue, limit multi-hop migrations. Our setup includes two $1 \times I$ migration queues, allowing up to two migrations per task. Tasks requiring more migrations are discarded, which negatively impacts system throughput.
- *Constraint Satisfaction Issues*: In the reinforcement learning framework, all constraints are transformed into rewards, which does not effectively ensure their satisfaction. Increasing penalty terms has proven insufficient for guaranteeing constraints like (9j) and (9l). Consequently, tasks that do not meet these constraints are discarded from throughput calculations.
- *Queue Stability Concerns*: Due to the minimal caching costs imposed by constraint (19 m), the cache queue experiences significant variability, which the reinforcement learning framework cannot stabilize, potentially leading to an infinite backlog.
- *Convergence Challenges*: The original problem's complexity, featuring five interdependent binary variables, long-term infinite constraints, and UAV deployment linked to scheduling decisions, hinders convergence within the reinforcement learning framework. Despite numerous trials across various models, outcomes remain suboptimal. *Due to space constraints, detailed experimental results are provided in Section X of supplementary material, available online.*

VIII. Conclusion and Future Work

In this paper, we harness UAV-assisted MEC to enhance the real-time data processing capabilities of edge computing within infrastructure deployments effectively. Our methodology addresses the often-overlooked synergistic effects between strategies by co-optimizing the UAV's trajectory, task offloading, caching, and migration strategies. Utilizing the Lyapunov optimization framework and BCD method, we decompose the problem into subproblems: UAV deployment and User-UAV association, task offloading optimization, scheduling optimization, and bandwidth optimization within a single time slot. The first three subproblems are tackled using specialized algorithms, while the complex scheduling problem is transformed into a non-convex constrained quadratic programming problem, efficiently solved by semidefinite relaxation and probabilistic mapping methods. Experimental results indicate that our approach enhances system throughput by 10%-45%, reduces scheduling costs by 15%-30%, and cuts algorithm execution time by 8%-37% compared to conventional algorithms, providing a deeper insight into the dynamic interplay between UAV trajectory planning and task management.

While our research has primarily focused on optimizing operational efficiencies under standard conditions, it has not addressed network disruptions or communication failures. Moving forward, we plan to tackle two significant challenges identified in our studies. First, we will explore scenarios where UAVs unexpectedly run low on battery or exit the network due to mechanical failures, to enhance their operational reliability and resilience. Second, we aim to develop solutions that improve UAV communication in dense urban areas, particularly addressing the challenges posed by high-rise buildings and mixed LoS/NLoS conditions. Our efforts will concentrate on advancing UAV coordination and ensuring robust communication in complex urban environments.

References

[1] X. Diao, X. Guan, and Y. Cai, "Joint offloading and trajectory optimization for complex status updates in UAV-assisted Internet of Things," *IEEE Internet Things J.*, vol. 9, no. 23, pp. 23 881–23 896, Dec. 2022.

[2] Y. K. Tun, T. N. Dang, K. Kim, M. Alsenwi, W. Saad, and C. S. Hong, "Collaboration in the sky: A distributed framework for task offloading and resource allocation in multi-access edge computing," *IEEE Internet Things J.*, vol. 9, no. 23, pp. 24 221–24 235, Dec. 2022.

[3] B. Liu, Y. Wan, F. Zhou, Q. Wu, and R. Q. Hu, "Resource allocation and trajectory design for MISO UAV-assisted MEC networks," *IEEE Trans. Veh. Technol.*, vol. 71, no. 5, pp. 4933–4948, May 2022.

[4] H. Guo, Y. Wang, J. Liu, and C. Liu, "Multi-UAV cooperative task offloading and resource allocation in 5G advanced and beyond," *IEEE Trans. Wireless Commun.*, vol. 23, no. 1, pp. 347–359, Jan. 2024.

[5] L. Bao, J. Luo, H. Bao, Y. Hao, and M. Zhao, "Cooperative computation and cache scheduling for UAV-enabled MEC networks," *IEEE Trans. Green Commun. Netw.*, vol. 6, no. 2, pp. 965–978, Jun. 2022.

[6] D. Wang, J. Tian, H. Zhang, and D. Wu, "Task offloading and trajectory scheduling for UAV-enabled MEC networks: An optimal transport theory perspective," *IEEE Wireless Commun. Lett.*, vol. 11, no. 1, pp. 150–154, Jan. 2022.

[7] P. Qin, Y. Fu, Y. Xie, K. Wu, X. Zhang, and X. Zhao, "Multi-agent learning-based optimal task offloading and UAV trajectory planning for AGIN-power IoT," *IEEE Trans. Commun.*, vol. 71, no. 7, pp. 4005–4017, Jul. 2023.

[8] Y. He, Y. Gan, H. Cui, and M. Guizani, "Fairness-based 3-D Multi-UAV trajectory optimization in multi-UAV-assisted MEC system," *IEEE Internet Things J.*, vol. 10, no. 13, pp. 11 383–11 395, Jul. 2023.

[9] B. Xu, Z. Kuang, J. Gao, L. Zhao, and C. Wu, "Joint offloading decision and trajectory design for UAV-enabled edge computing with task dependency," *IEEE Trans. Wireless Commun.*, vol. 22, no. 8, pp. 5043–5055, Aug. 2023.

[10] X. Zhou, L. Huang, T. Ye, and W. Sun, "Computation bits maximization in UAV-assisted MEC networks with fairness constraint," *IEEE Internet Things J.*, vol. 9, no. 21, pp. 20 997–21 009, Nov. 2022.

[11] Z. Yang, S. Bi, and Y.-J. A. Zhang, "Online trajectory and resource optimization for stochastic UAV-enabled MEC systems," *IEEE Trans. Wireless Commun.*, vol. 21, no. 7, pp. 5629–5643, Jul. 2022.

[12] L. Liu, X. Yuan, N. Zhang, D. Chen, K. Yu, and A. Taherkordi, "Joint computation offloading and data caching in multi-access edge computing enabled Internet of Vehicles," *IEEE Trans. Veh. Technol.*, vol. 72, no. 11, pp. 14 939–14 954, Nov. 2023.

[13] Z. Chen, W. Yi, A. S. Alam, and A. Nallanathan, "Dynamic task software caching-assisted computation offloading for multi-access edge computing," *IEEE Trans. Commun.*, vol. 70, no. 10, pp. 6950–6965, Oct. 2022.

[14] B. Liu, C. Liu, and M. Peng, "Computation offloading and resource allocation in unmanned aerial vehicle networks," *IEEE Trans. Veh. Technol.*, vol. 72, no. 4, pp. 4981–4995, Apr. 2023.

[15] J. Li, S. Chu, F. Shu, J. Wu, and D. N. K. Jayakody, "Contract-based small-cell caching for data disseminations in ultra-dense cellular networks," *IEEE Trans. Mobile Comput.*, vol. 18, no. 5, pp. 1042–1053, May 2019.

[16] V. Sharma, I. You, D. N. K. Jayakody, D. G. Reina, and K.-K. R. Choo, "Neural-blockchain-based ultrareliable caching for edge-enabled UAV networks," *IEEE Trans. Ind. Inform.*, vol. 15, no. 10, pp. 5723–5736, Oct. 2019.

[17] Y. Wu, S. Tang, L. Zhang, L. Fan, X. Lei, and X. Chen, "Resilient machine learning based semantic-aware MEC networks for sustainable next-G consumer electronics," *IEEE Trans. Consum. Electron.*, vol. 70, no. 1, pp. 2188–2199, Feb. 2024.

[18] S. Yang, J. Liu, F. Zhang, F. Li, X. Chen, and X. Fu, "Caching-enabled computation offloading in multi-region MEC network via deep reinforcement learning," *IEEE Internet Things J.*, vol. 9, no. 21, pp. 21 086–21 098, Nov. 2022.

[19] J. Chen, H. Xing, X. Lin, A. Nallanathan, and S. Bi, “Joint resource allocation and cache placement for location-aware multi-user mobile-edge computing,” *IEEE Internet Things J.*, vol. 9, no. 24, pp. 25 698–25 714, Dec. 2022.

[20] Y. Bai, D. Wang, and B. Song, “A knowledge graph-based cooperative caching scheme in MEC-enabled heterogeneous networks,” in *Proc. IEEE Glob. Commun. Conf.*, Rio de Janeiro, Brazil, 2022, pp. 5959–5964.

[21] T. N. Dang, A. Manzoor, Y. K. Tun, S. M. A. Kazmi, Z. Han, and C. S. Hong, “A contract-theory-based incentive mechanism for UAV-enabled VR-based services in 5G and beyond,” *IEEE Internet Things J.*, vol. 10, no. 18, pp. 16 465–16 479, Sep. 2023.

[22] Y. Chen, M. Liu, B. Ai, Y. Wang, and S. Sun, “Adaptive bitrate video caching in UAV-assisted MEC networks based on distributionally robust optimization,” *IEEE Trans. Mobile Comput.*, vol. 23, no. 5, pp. 5245–5259, May 2024.

[23] Z. Bai, Y. Lin, Y. Cao, and W. Wang, “Delay-aware cooperative task offloading for multi-UAV enabled edge-cloud computing,” *IEEE Trans. Mobile Comput.*, vol. 23, no. 2, pp. 1034–1049, Feb. 2024.

[24] X. Huang et al., “Joint interdependent task scheduling and energy balancing for multi-UAV-enabled aerial edge computing: A multiobjective optimization approach,” *IEEE Internet Things J.*, vol. 10, no. 23, pp. 20 368–20 382, Dec. 2023.

[25] X. Chen et al., “Dynamic service migration and request routing for microservice in multicell mobile-edge computing,” *IEEE Internet Things J.*, vol. 9, no. 15, pp. 13 126–13 143, Aug. 2022.

[26] I. Labriji et al., “Mobility aware and dynamic migration of MEC services for the Internet of Vehicles,” *IEEE Trans. Netw. Service Manag.*, vol. 18, no. 1, pp. 570–584, Mar. 2021.

[27] Z. Liao, Y. Ma, J. Huang, J. Wang, and J. Wang, “HOTSPOT: A UAV-assisted dynamic mobility-aware offloading for mobile-edge computing in 3-D space,” *IEEE Internet Things J.*, vol. 8, no. 13, pp. 10 940–10 952, Jul. 2021.

[28] J. Li et al., “Joint optimization on trajectory, altitude, velocity, and link scheduling for minimum mission time in UAV-aided data collection,” *IEEE Internet Things J.*, vol. 7, no. 2, pp. 1464–1475, Feb. 2020.

[29] H. Xiao, Z. Hu, K. Yang, Y. Du, and D. Chen, “An energy-aware joint routing and task allocation algorithm in MEC systems assisted by multiple UAVs,” in *Proc. Int. Wireless Commun. Mobile Comput.*, Limassol, Cyprus, 2020, pp. 1654–1659.

[30] M. Zhao, W. Li, L. Bao, J. Luo, Z. He, and D. Liu, “Fairness-aware task scheduling and resource allocation in UAV-enabled mobile edge computing networks,” *IEEE Trans. Green Commun. Netw.*, vol. 5, no. 4, pp. 2174–2187, Dec. 2021.

[31] H. Mei, K. Yang, Q. Liu, and K. Wang, “Joint trajectory-resource optimization in UAV-enabled edge-cloud system with virtualized mobile clone,” *IEEE Internet Things J.*, vol. 7, no. 7, pp. 5906–5921, Jul. 2020.

[32] M. Guo, W. Wang, X. Huang, Y. Chen, L. Zhang, and L. Chen, “Lyapunov-based partial computation offloading for multiple mobile devices enabled by harvested energy in MEC,” *IEEE Internet Things J.*, vol. 9, no. 11, pp. 9025–9035, Jun. 2022.

[33] Z. Tong, J. Cai, J. Mei, K. Li, and K. Li, “Dynamic energy-saving offloading strategy guided by Lyapunov optimization for IoT devices,” *IEEE Internet Things J.*, vol. 9, no. 20, pp. 19 903–19 915, Oct. 2022.

[34] J. Su, Z. Liu, Y.-A. Xie, Y. Li, K. Ma, and X. Guan, “UEE-delay balanced online resource optimization for cooperative MEC-enabled task offloading in dynamic vehicular networks,” *IEEE Internet Things J.*, vol. 11, no. 7, pp. 11496–11507, Apr. 2024.

[35] H. Hu, Q. Wang, R. Q. Hu, and H. Zhu, “Mobility-aware offloading and resource allocation in a MEC-enabled IoT network with energy harvesting,” *IEEE Internet Things J.*, vol. 8, no. 24, pp. 17 541–17 556, Dec. 2021.

[36] M. Zhao et al., “Energy-aware task offloading and resource allocation for time-sensitive services in mobile edge computing systems,” *IEEE Trans. Veh. Technol.*, vol. 70, no. 10, pp. 10 925–109 403, Oct. 2021.

[37] F. Li, C. He, X. Li, J. Peng, and K. Yang, “Geometric analysis-based 3D anti-block UAV deployment for mmWave communications,” *IEEE Commun. Lett.*, vol. 26, no. 11, pp. 2799–2803, Nov. 2022.

[38] J. Luo, J. Song, F.-C. Zheng, L. Gao, and T. Wang, “User-centric UAV deployment and content placement in cache-enabled multi-UAV networks,” *IEEE Trans. Veh. Technol.*, vol. 71, no. 5, pp. 5656–5660, May 2022.

[39] J. Wang et al., “Multiple unmanned-aerial-vehicles deployment and user pairing for nonorthogonal multiple access schemes,” *IEEE Internet Things J.*, vol. 8, no. 3, pp. 1883–1895, Feb. 2021.

[40] M.-H. Chen, M. Dong, and B. Liang, “Resource sharing of a computing access point for multi-user mobile cloud offloading with delay constraints,” *IEEE Trans. Mobile Comput.*, vol. 17, no. 12, pp. 2868–2881, Dec. 2018.

[41] M. Zhao, H. Bao, L. Yin, J. Yao, and T. Q. Quek, “Secrecy offloading rate maximization for multi-access mobile edge computing networks,” *IEEE Commun. Lett.*, vol. 25, no. 12, pp. 3800–3804, Dec. 2021.

[42] D. Tse and P. Viswanath, *Fundamentals of Wireless Communication*. Cambridge, U.K.: Cambridge Univ. Press, 2005.

Mingxiong Zhao (Member, IEEE) received the BS degree in electrical engineering and the PhD degree in information and communication engineering from the South China University of Technology (SCUT), Guangzhou, China, in 2011 and 2016, respectively. He was a visiting PhD student with the University of Minnesota (UMN), Twin Cities, MN, USA, from 2012 to 2013 and the Singapore University of Technology and Design (SUTD), Singapore, from 2015 to 2016, respectively. Currently, he is a full professor and the Donglu Young Scholar with Yunnan University (YNU), Kunming, China, where he also serves as the director of Cybersecurity Department with the School of Software. His current research interests include network security, mobile edge computing, and edge AI techniques.

Rongqian Zhang received the MS degree in software engineering from Yunnan University (YNU), Kunming, China, in 2024. She is currently working toward the PhD degree in information and communication engineering with the School of Information Science and Engineering, YNU. Her current research interests center on mobile edge computing, focusing on UAV-assisted MEC networks, MEC network security, and task scheduling strategies in MEC networks.

Zhenli He (Senior Member, IEEE) received the MS degree in software engineering and the PhD degree in systems analysis and integration from Yunnan University (YNU), Kunming, China, in 2012 and 2015, respectively. He was a postdoctoral researcher with Hunan University (HNU), Changsha, China, from 2019 to 2021. He is currently an associate professor and the deputy head with the Department of Software Engineering, School of Software, YNU. His current research interests include edge computing, energy-efficient computing, heterogeneous computing, and edge AI techniques.

Keqin Li (Fellow, IEEE) received the BS degree in computer science from Tsinghua University, in 1985, and the PhD degree in computer science from the University of Houston, in 1990. He is currently a SUNY distinguished professor with the State University of New York and a National distinguished professor with Hunan University (China). He has authored or co-authored more than 950 journal articles, book chapters, and refereed conference papers. He received several best paper awards from international conferences including *PDPTA-1996*, *NAECON-1997*, *IPDPS-2000*, *ISPA-2016*, *NPC-2019*, *ISPA-2019*, and *CPSCom-2022*. He holds nearly 70 patents announced or authorized by the Chinese National Intellectual Property Administration. He is among the world’s top five most influential scientists in parallel and distributed computing in terms of single-year and career-long impacts based on a composite indicator of the Scopus citation database. He was a 2017 recipient of the Albert Nelson Marquis Lifetime Achievement Award for being listed in Marquis Who’s Who in Science and Engineering, Who’s Who in America, Who’s Who in the World, and Who’s Who in American Education for more than twenty consecutive years. He received the Distinguished Alumnus Award from the Computer Science Department, University of Houston in 2018. He received the IEEE TCCLD Research Impact Award from the IEEE CS Technical Committee on Cloud Computing in 2022 and the IEEE TCSVC Research Innovation Award from the IEEE CS Technical Community on Services Computing in 2023. He is a member of the SUNY Distinguished Academy. He is an AAAS fellow, and an AAIA fellow. He is a member of Academia Europaea (Academician of the Academy of Europe).