Two Time-Scale Joint Service Caching and Task Offloading for UAV-assisted Mobile Edge Computing

Ruiting Zhou[†], Xiaoyi Wu[†], Haisheng Tan[‡]*, Renli Zhang[†]
[†]School of Cyber Science and Engineering, Wuhan University, China
[‡]School of Computer Science and Technology, University of Science and Technology of China, China
Email: {ruitingzhou, xiaoyiwu, zhang_rl}@whu.edu.cn, hstan@ustc.edu.cn

IEEE INFOCOM 2022 - IEEE Conference on Computer Communications | 978-1-6654-5822-1/22/$31.00 ©2022 IEEE | DOI: 10.1109/INFOCOM48880.2022.9796714

***Abstract*—The emergence of unmanned aerial vehicles (UAVs) extends the mobile edge computing (MEC) services in broader coverage to offer new flexible and low-latency computing services for user equipment (UE) in the era of 5G and beyond. One of the fundamental requirements in UAV-assisted mobile wireless systems is the low latency, which can be jointly optimized with service caching and task offloading. However, this is challenged by the communication overhead involved with service caching and constrained by limited energy capacity. In this work, we present a comprehensive optimization framework with the objective of minimizing the service latency while incorporating the unique features of UAVs. Specifically, to reduce the caching overhead, we make caching placement decision every T slots (specified by service providers), and adjust UAV trajectory, user equipment or UE-UAV association, and task offloading decisions at each time slot under the constraints of UAV's energy and resource capacity. By leveraging Lyapunov optimization approach and dependent rounding technique, we design an alternating optimization-based algorithm, named TJSO, which iteratively optimizes caching and offloading decisions. Theoretical analysis proves that TJSO converges to the near-optimal solution in polynomial time. Extensive simulations further verify that our proposed solution can significantly reduce the service delay for UEs while maintaining low energy consumption when compared to the three state-of-the-art baselines.**

I. Introduction

The fast development of internet of things (IoT) generates a large amount of data at the edge, which boosts the applications of mobile edge computing (MEC), *e.g.,* smart city and automatic driving [1]. These applications requires low latency and high computation power, which is realized by offloading services to edge servers at local wireless access points (APs) or cellular base stations (BSs) [2]. Unfortunately, the infrastructure-based MEC in wireless networks brings high deployment cost, and may not be able to provide services in rural areas without sufficient infrastructures or urban areas in peak hours [3], [4]. To compensate the drawback of infrastructure-based MEC, unmanned aerial vehicles (UAVs) equipped with MEC servers have emerged as a promising technology to provide flexible and cost-efficient computing services in the era of 5G and beyond [4], [5]. UAVs are easy to control and deploy, and have high mobility, which has encouraged industries in diverse practices in today's life. For example, on July 21, 2021, the Chinese Ministry of Emergency Management dispatched UAVs to build the emergency communication platform in the communication interruption area of Mihe Town. The mobile base station enabled by UAVs achieves a long-term and stable continuous mobile signal coverage of about 50 square kilometers for five hours [6].

However, how to fully utilize UAV-assisted MEC in the wireless network to provide low-latency services is indeed challenging. *First*, many popular applications are data-driven and require to cache some service contents at the MEC servers in UAVs, *e.g.,* library/code and machine learning model [7], [8]. But due to the limited storage space of UAVs, only partial services can be stored. Moreover, frequent downloading service content from the remote cloud brings heavy communication overhead in terms of both time and cost [9], [10]. *Second*, because the amount of energy obtained by each UAV from one charge is limited, it cannot provide continuous computing services for all tasks in a long time [11]. *Third*, although the mobility of UAVs increase the flexibility and elasticity of MEC, it also bring significant difficulties in the UAV trajectory planing [12], [13]. Therefore, where to cache and offload which service to minimize the service delay (*i.e.,* the overall time of serving UEs' tasks), in the UAV-assisted wireless network becomes the fundamental problem.

Existing researches on MEC mainly focus on the service caching or offloading in a fixed edge or cellular network [14]-[15]. Their solution cannot be directly applied to the UAV-assisted MEC, since UAV trajectory optimization is not studied. There is some related work on UAV-enabled MEC [16]-[17], but the optimization objective is either to realize full converge or minimize the overall UEs'/UAVs' energy consumption [18], [19]. We will discuss it in details in Sec. II. To the best of our knowledge, none of them consider the caching overhead and the limited capacity of UAV's battery. In this paper, we study the two time-scale joint service caching and task offloading problem in UAV-assisted wireless networks, with the goal of service delay minimization. In particular, we capture important features of UAVs (*i.e.,* mobility and limited energy) and take caching overhead into consideration. We make caching placement decision every T slots, and adjust

*This work is supported in part by the NSFC Grants (62072344, U20A20177, 62132009 and 61772489), Hubei Science Foundation (2020CFB195) and Compact Exponential Algorithm Project of Huawei (YBN2020035131). The corresponding author is Haisheng Tan.

978-1-6654-5822-1/22/$31.00 ©2022 IEEE

UAV trajectory, UE-UAV association, and task offloading decisions at each time slot. The value of T is specified by the UAV service provider, and can be adjusted based on the size of caching contents. The joint service caching and task offloading problem is extremely hard, as it is a non-convex mixed integer nonlinear program (MINLP), which couples four sets of variables. By exploiting Lyapunov optimization approach and dependent rounding technique, we design an alternating optimization-based algorithm, TJSO, which iteratively optimizes decisions with guaranteed convergence. We highlight our contribution as following.

- We formulate the service caching and task offloading problem in UAV-assisted wireless networks. The goal is to minimize the service delay, which needs to jointly optimize service caching placement, UAV trajectory, UE-UAV association, and task offloading decisions. Different from existing literature, we are the *first* study that makes caching decision in a longer time window (every T slots) to reduce the communication overhead and bounds the long term energy consumption of a UAV under a certain limit. The optimization problem is a non-convex MINLP, which is extremely hard to solve.
- By leveraging Lyapunov optimization approach, the energy constraint can be represented by a virtual energy queue. To predict the future energy consumption and make caching decision, we approximate the future energy queue backlog value as the current value, and derive an upper bound on the objective function. In this way, we can decompose the T-time-slot problem into multiple one-time optimization problems. Next, we propose an alternating optimization-based algorithm, TJSO, that applies the dependent rounding technique to iteratively optimize four decision variables every T slots and three variables (except the caching decision) at each time slot.
- The efficiency of TJSO is verified by both theoretical analysis and large scale simulation studies based on real world data. We rigorously prove that TJSO can converge to a suboptimal solution in polynomial time. Furthermore, extensive simulations show that TJSO can efficiently serve latency-sensitive tasks and achieve near-to-optimal delay performance while keeping the energy consumption low, compared to three state-of-the-art algorithms.

In the rest of the paper, the related work is reviewed in Sec. II. We introduce the system model and problem formulation in Sec. III. The algorithm is present and evaluated in Sec. IV and Sec. V, respectively. Sec. VI concludes the paper.

II. Related Work

UAV-assisted MEC. UAV-assisted wireless communication is mainly adopted to realize full converge and enhance network performance. Ji *et al.* [16] consider the enhancement of the cell-edge communication through a UAV in multi-cell networks. Alzenad *et al.* [18] propose an optimal placement algorithm for UAV-BSs that maximizes the number of covered users. Chen *et al.* [20] use cache-enabled UAVs to optimize the quality of experience (QoE) of wireless devices in a cloud radio access network. Lyu *et al.* [21] jointly optimize UAV's trajectory, bandwidth allocation and user partitioning to maximize the minimum throughput of all mobile terminals. Some researchers consider offloading computing tasks to UAVs [19] [22] [23]. In [19] and [18], the authors try to minimize the energy consumption of UAVs. Yang *et al.* [12] aim to minimize the sum power in a MEC network with UAVs by jointly optimizing user connection, power control and computation capacity allocation. Both Ji *et al.* [17] and Wu *et al.* [24] maximize the minimum throughput among UAV-served users by jointly optimizing cache placement, UAV trajectory, and transmission power. Most of these papers ignore the problem off service caching. Since many existing popular applications are data-intensive and require a non-trivial amount of data stored in UAV's limited storage space, it is essential to study the cache placement on UAVs. What's more, in this paper we propose the two time-scale joint task offloading and service caching algorithm, due to the overhead of caching, which is ignored by all of the above papers.

Service Caching in MEC. Although the concept of MEC has been proposed for over a decade, service caching for MEC is only studied in the last five years. Urgaonkar *et al.* [25] model the migration of services in edge-clouds as a sequential decision making problem to optimize operational costs. Service caching with the routing of user requests is studied in [26] and [27]. Some researchers focus on cooperative caching. Zhang *et al.* [2] propose a vehicle-aided edge caching scheme, where the caching and computing resources at the wireless network edge are jointly scheduled. Xu *et al.* [14] consider service caching in a mobile edge network with multiple network service providers. In [28] and [29], the authors focus on caching virtual reality (VR) services in MEC. Besides, there are many studies that jointly consider service caching and task offloading. Both Xu *et al.* [30] and Ouyang *et al.* [31] study how to minimize the latency via dynamic service placement and task offloading in MEC. Ma *et al.* [15] develop an algorithm based on Gibbs sampling to consider cooperative service caching and workload scheduling in MEC systems. Bi *et al.* [32] jointly consider service caching and computation offloading in MEC systems to minimize the computation delay and energy consumption of mobile users. These studies on service caching in MEC can't be applied to UAV-assisted MEC directly, because of the finite coverage range and mobility of the UAV. In addition to optimizing the task offloading and service caching decisions, we also need to optimize the trajectory of the UAV jointly.

III. Model and Problem Formulation

System Overview. As shown in Fig. 1, we consider a mobile network of U rotary-wing UAVs. These UAVs, integrating with edge servers, have both computing and caching capabilities. The UAVs can provide K types of services for M user equipments (UEs). U rotary-wing UAVs are denoted as $\mathcal{U} = \{1, 2, ..., U\}$, and M UEs are denoted as $\mathcal{M} = \{1, 2, ..., M\}$. Let $\mathcal{K} = \{1, 2, ..., K\}$ denote the service set provided by

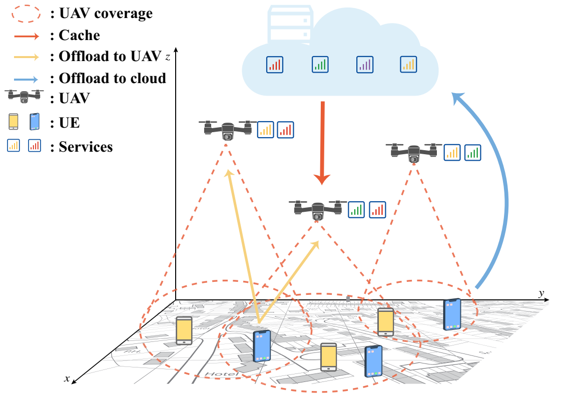

Fig. 1: An illustration of UAV-assisted wireless network.

UAVs. We divide the mission period into several time slots $\mathcal{T} = \left\{1, 2, ..., \widetilde{T}\right\}$. The tasks generated from UEs follow a Poisson process. Let $d^t_{m,k}$ denote the intensity (*i.e.,* Poisson rate) of tasks generated from UE m requiring service k at the time slot t. The tasks can be offloaded to UAVs only when the following two conditions are satisfied: i) the UE is connected with UAVs; ii) the corresponding service contents are cached in these UAVs. Otherwise, UEs will offload the tasks to the remote cloud for processing.

Service Caching. In order to reduce computation latency, we consider the scenario where services will be cached in the UAV's storage. Let c_k represent the storage space required by service k. Considering the limited storage space of a UAV, not all services can be cached at the same time. The total storage space occupied by services should be no more than UAV's storage capacity C_u. Since the communication overhead of caching services from the remote cloud to the UAV is large [9], the cache decisions are updated in every T time slots, where T is defined by the service provider. Therefore, the UAV has to judiciously decide which services to cache.

Task Offloading. Tasks from UEs can be offloaded to UAVs or the remote cloud. The computing capacity (in terms of CPU cycles) of UAV u is f_u. We assume that the average workload (in terms of CPU cycles) of type k tasks is μ_k.

UAV Trajectory. Considering a 3-D Cartesian coordinate system, we assume UAV u flies at a fixed altitude H_u, then $G^t_u = \left(x^t_u, y^t_u\right)$ can be used to represent the horizontal coordinate of UAV u at time slot t. Similarly, UE m has a zero altitude and its horizontal position can be denoted as $G'_m = (x_m, y_m)$. Based on the Euclidean formula, the horizontal distance between UE m and UAV u in slot t is $R^t_{u,m} = \sqrt{(x^t_u - x_m)^2 + (y^t_u - y_m)^2}$. This distance needs to be within the maximum radiation radius of the UAV R^{max}_u. The corresponding achievable uplink data rate of UE m is given by $r^t_{u,m} = B\log_2(1 + \frac{\delta p_{u,m}}{(R^t_{u,m})^2 + H^2_u})$ [13], where B is the spectrum bandwidth of each communication channel, δ is SNR, and $p_{u,m}$ is the transmission power from UE m to UAV u.

Decision Variables. The following decisions need to be made jointly: i) $a^t_{u,k} \in \{0, 1\}$, a binary variable denotes whether service k is cached or not on UAV u at time t, which is updated every T time slots; ii) $x^t_{u,m} \in \{0, 1\}$, whether UE m is connected with UAV u at time slot t; iii) $y^t_{u,m,k} \in [0, 1]$, the fraction of tasks requesting service k from UE m offloaded to UAV u at time slot t; iv) $G^t_u = \left(x^t_u, y^t_u\right)$, the horizontal coordinate of the UAV u at time slot t.

Energy Consumption of UAVs. i) *Computation Cost:* The amount of tasks from UE m requesting service k to UAV u can be calculated as $\lambda^t_{u,m,k} = d^t_{m,k} y^t_{u,m,k}$. Computation consumption of UAV u can be represented as $E^t_{u,c} = \kappa \sum_m \sum_k \lambda^t_{u,m,k}\mu_k$, where κ is the unit energy cost of computation; ii) *Communication Cost:* The communication cost is used to transmit service contents between UAVs and UEs, which is the product of the transmission power and the transmission time. The communication cost can be calculated as $E^t_{u,m} = \sum_m p_{u,m} \frac{\sum_k \lambda^t_{u,m,k} l_k}{r^t_{u,m}}$, where l_k is average size of type k task; iii) *Caching Cost:* The caching cost for UAV u is caused by storing caching contents, which depends on the size of caching contents. The cache cost can be calculated as $E^t_{u,s} = \gamma \sum_k a^t_{u,k} c_k$, where γ is the unit energy cost for storing. Therefore, the total energy consumption* of UAV u is

$$E^t_u = E^t_{u,c} + E^t_{u,m} + E^t_{u,s}.$$

Service Delay. The service delay is the overall service time, consisting of the following three parts at time slot t: i) *Communication delay between UAVs and UEs:* The communication delay of UAV u in time slot t is the communication time for task offloading. It is calculated as $D^t_{u,c} = \sum_m \sum_k \frac{\lambda^t_{u,m,k} l_k}{r^t_{u,m}}$; ii) *Computation delay on UAV:* The computation delay for UAV u can be calculated by $D^t_{u,e} = \sum_m \sum_k \frac{\lambda^t_{u,m,k}\mu_k}{f_u}$. iii) *Communication delay from UEs to the remote cloud:* The unserved tasks would be offloaded to the remote cloud. We assume that the remote cloud has ample computing power. Hence, the delay for tasks offloaded to the cloud is mainly due to the transmission delay. The delay is given by $D^t_l = \frac{\sum_m \sum_k d^t_{m,k} l_k - \sum_u \sum_m \sum_k \lambda^t_{u,m,k} l_k}{\overline{R}}$, where $\overline{R}$ is the average transmission rate between the remote cloud and UEs. Therefore, the total expected service delay at time slot t is:

$$D^t = \sum_u (D^t_{u,c} + D^t_{u,e}) + D^t_l$$

Problem Formulation. Our objective is to minimize the service delay while keeping the UAVs' energy consumption no more than the upper bound Q, because the amount of energy obtained by each UAV from one charge is limited. We jointly optimize the service caching $A = \{a^t_{u,k}\}_{u\in U, k\in K, t=\theta T}$, UE-UAV association $X = \{x^t_{u,m}\}_{u\in U, m\in M, t\in\tilde{T}}$, task offloading $Y = \{y^t_{u,m,k}\}_{u\in U, m\in M, k\in K, t\in\tilde{T}}$, and UAV movement trajectory $G = \{G^t_u\}_{u\in U, t\in\tilde{T}}$. Therefore, the optimization problem is formulated as:

$$(\mathbf{P1}) \quad \min_{A,X,Y,G} \frac{1}{\widetilde{T}} \sum_{t=1}^{\widetilde{T}} \mathbb{E}\{D^t\}$$

$$y^t_{u,m,k} \leq x^t_{u,m}, \forall u, \forall m, \forall k, \forall t \tag{1a}$$

$$y^t_{u,m,k} \leq a^t_{u,k}, \forall u, \forall m, \forall k, \forall t \tag{1b}$$

$$\textstyle\sum_u y^t_{u,m,k} \leq 1, \forall m, \forall k, \forall t \tag{1c}$$

[*] We assume that the flight distance of UAV u between two slots is upper bounded by d^{max}_u. Hence, the flight energy can be considered as a constant and ignored in the total energy consumption.

$$\sum_m \sum_k \lambda_{u,m,k}^t \mu_k \leq f_u, \forall u, \forall t \tag{1d}$$
$$\sum_k a_{u,k}^t c_k \leq C_u, \forall u, \forall t \tag{1e}$$
$$\sum_m x_{u,m}^t \leq N_u^{max}, \forall u, \forall t \tag{1f}$$
$$x_{u,m}^t (R_{u,m}^t)^2 \leq (R_u^{max})^2, \forall u, \forall m, \forall t \tag{1g}$$
$$||G_u^{t+1} - G_u^t||^2 \leq (d_u^{max})^2, \forall u, \forall t \tag{1h}$$
$$E_u^t \leq E^{max}, \forall u, \forall t \tag{1i}$$
$$\frac{1}{\widetilde{T}} \sum_{t=1}^{\widetilde{T}} \mathbb{E}\{E_u^t\} \leq Q, \forall u \tag{1j}$$
$$a_{u,k}^t = a_{u,k}^{t+\Delta}, \forall k, \forall u, \Delta = 1, ..., T-1, t = \theta T \tag{1k}$$
$$a_{u,k}^t \in \{0,1\}, x_{u,m}^t \in \{0,1\}, \forall k, \forall u, \forall m, \forall t \tag{1l}$$
$$y_{u,m,k}^t \in [0,1], \forall k, \forall u, \forall m, \forall t \tag{1m}$$

Constraint (1a) and (1b) ensures that only when UE is connected with UAV and this UAV has the requested cache content, UE will offload tasks to the UAV. Constraint (1c) represents the sum of the fraction of offloaded tasks to all UAVs is no more than 1. Constraint (1d) captures individual UAV's computation capacity. Constraint (1e) captures individual UAV's storage space capacity. The constraint (1f) states that the maximum number of connected UEs at any time slot (N_u^{max}) due to the spectrum limitation. Constraint (1g) guarantees that UEs should be within the coverage area of the corresponding UAVs. Constraint (1h) represents the constraint of UAV's flight distance between two time slots. Constraint (1i) is the energy upper bound of each UAV at every time slot. Constraint (1j) is the long-term energy constraint for UAVs, which requires that the long-term average total energy consumption does not exceed an upper limit Q. Constraint (1k) ensures cache decisions unchanged during T time slots.

Challenges. Due to unknown distributions of task demands in all time slots, the first challenge is the lack of future information. Optimally solving **P1** requires complete offline information. Secondly, **P1** is a mixed integer nonlinear programming and is very difficult to solve even if the future information is known. Finally, the cache decision is updated every T time slots. The offloading decision and UAV trajectory is updated every time slot. **P1** couples two time scales.

TABLE I: Notations

$U/\mathcal{U}$	number/set of UAVs
$M/\mathcal{M}$	number/set of UEs
$\widetilde{T}/\mathcal{T}$	number/set of time slots
$K/\mathcal{K}$	number of types/set of services
T	the interval between cache decisions
$a_{u,k}^t$	whether service k is cached on UAV u at t
$x_{u,m}^t$	whether UE m is associated with UAV u at t
$y_{u,m,k}^t$	the fraction of tasks requesting service k from UE m offloaded to UAV u at t
G_u^t	the horizontal coordinate of UAV u at t
c_k	size of the storage space required by service k
C_u	storage capacity of UAV u
$d_{m,k}^t$	the Poisson rate of tasks generated from UE m requesting service k at t
$\lambda_{u,m,k}^t$	the amount of tasks from UE m requesting service k to UAV u at t
f_u	the computational capability of the UAV u
μ_k	the average workload of type k task
l_k	the average size of type k task

IV. Algorithm Design

A. Algorithm Idea

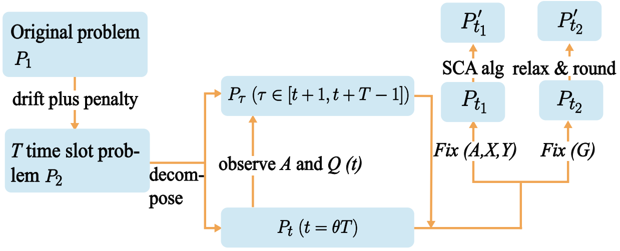

Fig. 2: Main idea of TJSO.

In this section, we leverage Lyapunov optimization approach and dependent rounding technique to design an alternating optimization-based algorithm, TJSO, which iteratively optimizes four decision variables every T slots and three variables (except the caching decision) at each time slot with guaranteed convergence. As shown in Fig. 2, our proposed algorithms consist of following components.

i. We first establish the virtual energy queue $\mathbf{Q}(t)$ to represent the energy constraint (1j). Following the Lyapunov optimization approach, we derive the *T-time-slot drift-plus-penalty* function. Cache decisions are made every T time slot, but the task arrival rate varies during this time period. To overcome this obstacle, we approximate the future energy queue backlog value as the current value, and derive a loose upper bound of the *T-time-slot drift-plus-penalty* function. We obtain the problem **P2** based on the upper bound. Second, we decompose the T-time-slot problem **P2** into multiple one-time optimization problems. We solve the problem $\mathbf{P_t}$ at time $t = \theta T(\theta = 1, 2, ..., \Theta)$, and problem $\mathbf{P}_\tau$ at time $\tau \in [t+1, t+T-1]$.

ii. To obtain a suboptimal solution to $\mathbf{P_t}$, we propose an alternating optimization-based algorithm that iteratively optimizes two different subproblems. First, with the fixed service caching A, UE-UAV association X and task offloading Y, the UAV trajectory subproblem $\mathbf{P_{t1}}$ is a continuous non-convex problem, which can be optimally solved by the successive convex approximation (SCA) technique. Second, we fix the UAV trajectory G, and solve the subproblem $\mathbf{P_{t2}}$ to joint optimize A, X and Y. Subproblem $\mathbf{P_{t2}}$ is a mixed integer linear programming problem. We relax the problem to get the fractional solution by interior point method, and then round the fractional solution to get integer solution by dependent rounding algorithm.

iii. For $\tau \in [t+1, t+T-1]$, with the known cache decision A and predicted virtual energy queue $\mathbf{Q}(t)$, we make decisions on $\{X, Y, G\}$ to solve $\mathbf{P}_\tau$. The algorithm to solve problem $\mathbf{P}_\tau$ is similar to that to solve problem $\mathbf{P_t}$.

B. Two Time-Scale Lyapunov Optimization

We establish a virtual energy queue $Q_u(t)$ to represent the energy constraint (1j):

$$Q_u(t+1) = \max\left\{Q_u(t) + E_u^t - Q, 0\right\}, \tag{2}$$

where $Q_u(t)$ is the queue backlog on UAV u in time slot t, indicating the deviation of current energy consumption from the energy constraint. We next define the Lyapunov function as follows:

$$L(t) = \frac{1}{2}\sum_u Q_u^2(t). \tag{3}$$

Based on these two values, we define the *T-time-slot Lyapunov drift*, $\Delta_T \mathbf{Q}(t)$ as the expected change in the Lyapunov function over T slots, where $\mathbf{Q}(t) = \{Q_u(t)\}_{u \in U}$.

$$\Delta_T(\mathbf{Q}(t)) = \mathbb{E}\left\{L(t+T) - L(t)|\mathbf{Q}(t)\right\}. \tag{4}$$

Following the Lyapunov optimization approach, we add the expected delay over T slots (*i.e.,* a penalty function), $\mathbb{E}\{\sum_{\tau=t}^{t+T-1} D^t\}$, to Eq. (4) to obtain the *T-time-slot drift-plus-penalty* function:

$$\Delta_T(\mathbf{Q}(t)) + V\mathbb{E}\{\sum_{\tau=t}^{t+T-1} D^t|\mathbf{Q}(t)\}, \tag{5}$$

where the control parameter $V > 0$ represents an importance weight on how much we emphasize the task delay minimization. Using the Lyapunov optimization technique, we decompose the long term optimization problem into T-time-slot optimization problems. The theoretical bound on the *T-time-slot drift-plus-penalty* function is given in Theorem 1.

Theorem 1. *Let $V > 0$ and $t = \theta T$ for some nonnegative integer θ. Then, under the decision variables, we have*

$$\Delta_T(\mathbf{Q}(t)) + V\mathbb{E}\{\sum_{\tau=t}^{t+T-1} D^\tau|\boldsymbol{Q}(t)\} \leq B_1 T + V\mathbb{E}\{\sum_{\tau=t}^{t+T-1} D^\tau|\boldsymbol{Q}(t)\}$$
$$+ \sum_u \mathbb{E}\{\sum_{\tau=t}^{t+T-1} Q_u(\tau)(E_u^\tau - Q)|\boldsymbol{Q}(t)\} \tag{6}$$

Here, $B_1 = \frac{1}{2}\sum_u[(E^{max})^2 + Q^2]$.

Proof. See Appendix. A. □

Theorem 1 shows that the *T-time-slot drift-plus-penalty function* is deterministically upper bounded. We need to minimize the right-hand-side (RHS) of the inequality (6). By minimizing the function over T slots, we can realize the suboptimal minimization of the original objective function of **P1**, subject to Constraint (1a) ∼ (1i), (1k)∼(1m).

However, we still need to know the future information or distribution of the uncertain information (*i.e.,* uncertain task arrival) in each T time slots. The future queue backlogs $\mathbf{Q}(t)$ over the time period $[t, t+T-1]$ depends on the task arrival rates $d_{m,k}^t$ and all decisions $\{A, X, Y, G\}$. Therefore, we approximate future queue backlog values as the current value, *i.e.,* $Q(\tau) = Q(t)$ for all $\tau \in [t, t+T-1]$. However, such approximation loosens the upper bound in (6). Theorem 2 shows the loose *T-time-slot drift-plus-penalty* bound.

Theorem 2. *Let $V > 0$, the* T-time-slot drift-plus-penalty *function satisfies*

$$\Delta_T(\mathbf{Q}(t)) + V\mathbb{E}\{\sum_{\tau=t}^{t+T-1} D^\tau|\boldsymbol{Q}(t)\} \leq B_2 T + V\mathbb{E}\{\sum_{\tau=t}^{t+T-1} D^\tau|\boldsymbol{Q}(t)\}$$
$$+ \sum_u \mathbb{E}\{\sum_{\tau=t}^{t+T-1} Q_u(t)(E_u^\tau - Q)|\boldsymbol{Q}(t)\} \tag{7}$$

Here, $B_2 = B_1 + \frac{T-1}{2}\sum_u E^{max}[E^{max} - Q]$.

Proof. See Appendix. B. □

Finally, from the RHS of (7), we can get the simplified optimization problem **P2**:

$$(\mathbf{P2}) \min_{A,X,Y,G} \quad V\mathbb{E}\{\sum_{\tau=t}^{t+T-1} D^\tau|\mathbf{Q}(t)\} + \sum_u \mathbb{E}\{\sum_{\tau=t}^{t+T-1} Q_u(t)E_u^\tau|\mathbf{Q}(t)\} \tag{8}$$

$$\text{s.t.} \quad (1a) \sim (1i), (1k)\sim(1m)$$

We can decompose the T-time-slot problem into multiple one-time problems. At each time slot we need to jointly make online decision variables $\{A, X, Y, G\}$ subject to Constraints (1a) ∼ (1i), (1k)∼(1m) without knowing the future information about the task arrival rates. Note that the service caching decision A is only determined in $t = \theta T$ ($\theta = 1, 2, ..., \Theta$). Our algorithm is summarized in Alg. 1.

Algorithm 1 Two Time-scale Online Control Algorithm (TJSO)

1. Making decisions for service caching, UE-UAV association, task offloading and UAV trajectory

At each time $t = \theta T$ ($\theta = 1, 2, ..., \Theta$), observe the virtual energy queue $\mathbf{Q}(t)$, and then decide the cache decisions A for T slots as well as $\{X, Y, G\}$ for current slot to minimize the following problem $\mathbf{P_t}$:

$$(\mathbf{P_t}) \min_{A,X,Y,G} \sum_u (\kappa \sum_m \sum_k \lambda_{u,m,k}^t \mu_k + \sum_m p_{u,m} \frac{\sum_k \lambda_{u,m,k}^t l_k}{r_{u,m}^t}$$
$$+ \gamma \sum_k a_{u,k}^t c_k) Q_u(t) + V(\sum_u \sum_m \sum_k (\frac{l_k}{r_{u,m}^t} + \frac{\mu_k}{f_u})\lambda_{u,m,k}^t$$
$$+ \frac{\sum_m \sum_k d_{m,k}^t l_k - \sum_u \sum_m \sum_k \lambda_{u,m,k}^t l_k}{\overline{R}}) \tag{9}$$

$$\text{s.t.} \quad (1a) \sim (1i), (1l)\sim(1m)$$

2. Making decisions for UE-UAV association, tasks offloading decisions and UAV trajectory

At each time slot $\tau \in [t+1, t+T-1]$, after observing the energy queue $\mathbf{Q}(t)$ and the cache decisions A, make decisions $\{X, Y, G\}$ to minimize the following problem $\mathbf{P}_\tau$:

$$(\mathbf{P}_\tau) \min_{X,Y,G} \sum_u (\kappa \sum_m \sum_k \lambda_{u,m,k}^\tau \mu_k + \sum_m p_{u,m} \frac{\sum_k \lambda_{u,m,k}^\tau l_k}{r_{u,m}^\tau}) Q_u(t)$$
$$+ V(\sum_u \sum_m \sum_k (\frac{l_k}{r_{u,m}^\tau} + \frac{\mu_k}{f_u})\lambda_{u,m,k}^\tau$$
$$+ \frac{\sum_m \sum_k d_{m,k}^\tau l_k - \sum_u \sum_m \sum_k \lambda_{u,m,k}^\tau l_k}{\bar{R}}) \tag{10}$$

$$\text{s.t.} \quad (1a) \sim (1d), (1f) \sim (1i), (1l)\sim(1m)$$

3. Queue Update

Update the queue $Q_u(\tau)$ using the Eq. (2).

C. Algorithm Design for Time Slot t

In order to obtain a suboptimal solution to the problem $\mathbf{P_t}$, we propose an alternating optimization-based algorithm that iteratively optimizes two subproblems, $\mathbf{P_{t1}}$ and $\mathbf{P_{t2}}$.

Suboptimal UAV Trajectory P_{t1}. We fix the service caching A, UE-UAV association X, and task offloading Y. The UAV trajectory subproblem $\mathbf{P_{t1}}$ is given by

$$(\mathbf{P_{t1}})\min_{G} \quad \sum_u \sum_m \sum_k \frac{\lambda^t_{u,m,k} l_k (V + p_{u,m} Q_u(t))}{B \log_2(1 + \frac{\delta p_{u,m}}{||G^t_u - G'_m||^2 + H^2_u})} \tag{11}$$
$$\text{s.t.} \quad (1g) \sim (1h)$$

To decrease the complexity, we introduce a continuous slack variable $\eta = \{\eta^t_{u,m}\}_{u \in \mathcal{U}, m \in \mathcal{M}, t \in \tilde{\mathcal{T}}}$, where

$$B \log_2(1 + \frac{\delta p_{u,m}}{||G^t_u - G'_m||^2 + H^2_u}) \geq \frac{1}{\eta^t_{u,m}} \tag{12}$$

Then, imposing η in problem $\mathbf{P_{t1}}$, it becomes to

$$(\mathbf{P'_{t1}})\min_{G} \quad V \sum_u \sum_m \sum_k \lambda^t_{u,m,k} l_k \eta^t_{u,m} (V + p_{u,m} Q_u(t)) \tag{13}$$
$$\text{s.t.} \quad (1g) \sim (1h)$$

In the above problem, the objective function is convex in η, the LHS and RHS of inequality (12) are convex in $||G^t_u - G'_m||$ and η, respectively. Now the difficulty lies in the LHS of Eq. (12). Similar to [13], we employ SCA technique to obtain the locally optimal solution of problem $\mathbf{P'_{t1}}$. We approximate the LHS of inequality (12) by its tight lower bound as follows:

$$\begin{aligned} & B \log_2(1 + \frac{\delta p_{u,m}}{||G^t_u - G'_m||^2 + H^2_u}) \\ & \geq \phi^t_{u,m}(j)(||G^t_u - G'_m||^2 - ||G^t_u(j) - G'_m||^2) + \varphi^t_{u,m}(j) \geq \frac{1}{\eta^t_{u,m}} \end{aligned} \tag{14}$$

where

$$\phi^t_{u,m}(j) = \frac{-B \delta p_{u,m} \log_2 e}{(\varsigma^t_{u,m}(j) + H^2_u)(\varsigma^t_{u,m}(j) + H^2_u + \delta p_{u,m})}$$
$$\varphi^t_{u,m}(j) = B \log_2(1 + \frac{\delta p_{u,m}}{\varsigma^t_{u,m}(j) + H^2_u}).$$
$$\varsigma^t_{u,m}(j) = ||G^t_u(j) - G'_m||^2$$

$\mathbf{P'_{t1}}$ in the jth iteration of SCA can be formulated as:

$$(\mathbf{P'_{t1}}(j))\min_{G,\eta} \quad V \sum_u \sum_m \sum_k \lambda^t_{u,m,k} l_k \eta^t_{u,m} (V + p_{u,m} Q_u(t)) \tag{15}$$
$$\text{s.t.} \quad (1g) \sim (1h)$$

It is not difficult to find that the above problem is a convex quadratically constrained quadratic program (QCQP). The SCA-based algorithm for problem $\mathbf{P_{t1}}$ is described in Alg. 2. Finally, the UAV trajectory subproblem is successfully solved.

Suboptimal Service Caching, UE-UAV Association, and Task Offloading P_{t2}. In this part, we fixed UAV trajectory G, and then the subproblem to decide service caching A, UE-UAV association X, and task offloading Y is given by

$$\begin{aligned} (\mathbf{P_{t2}}) \min_{A,X,Y} & \sum_u (\kappa \sum_m \sum_k \lambda^t_{u,m,k} \mu_k + \sum_m p_{u,m} \frac{\sum_k \lambda^t_{u,m,k} l_k}{r^t_{u,m}} \\ & + \gamma \sum_k a^t_{u,k} c_k) Q_u(t) + V(\sum_u \sum_m \sum_k (\frac{l_k}{r^t_{u,m}} + \frac{\mu_k}{f_u}) \lambda^t_{u,m,k} \\ & + \frac{\sum_m \sum_k d^t_{m,k} l_k - \sum_u \sum_m \sum_k \lambda^t_{u,m,k} l_k}{\overline{R}}) \end{aligned} \tag{16}$$
$$\text{s.t.} \quad (1a) \sim (1g), (1i), (1l), (1m)$$

By relaxing the integral decision variables $\{A, X\}$ to be fractional, problem $\mathbf{P_{t2}}$ then becomes a continuous linear program (LP) problem $\mathbf{P'_{t2}}$.

Algorithm 2 SCA Algorithm

Input: Feasible UAV trajectory $G(0) = \{G^t_u(0)\}_{u \in U, t \in \tilde{T}}$, step size sequence $\{\sigma^j\}_j \in (0, 1]$, iteration index $j = 0$, maximum iteration $J^{max} = 100$.
Output: UAV trajectory G

```
while $j < J^{max}$ do
    compute $\hat{G}(G(j))$, the optimal solution of problem $\mathbf{P'_{t1}}$(j).
    Set $G(j+1) \leftarrow G(j) + \sigma^j (\hat{G}(G(j)) - G(j))$;
    if $G(j+1)$ is a stationary solution of Problem $\mathbf{P'_{t1}}$ then
        Break.
    else
        Set $j \leftarrow j + 1$.
    end if
end while
Return $G(j)$.
```

Here, we use interior point method [33] to solve problem $\mathbf{P'_{t2}}$. The optimal fractional solution can be used to obtain an efficient solution to $\mathbf{P_{t2}}$ after appropriate rounding. In the following, we introduce a dependent rounding algorithm that converts the fractional solution into the integer solution with the theoretical analysis of the performance guarantee.

First, we round the fractional $\hat{A}$ to the integral $\bar{A}$, place such $\bar{A}$ into the problem (while keeping the problem feasible), re-solve the problem to obtain the new fractional $\{X^*, Y^*\}$. Then, we round X^* to the integral $\bar{X}$. This ensures Constraint (1e) $\sim$ (1g), (1l) is satisfied. Finally, we place $\{\bar{A}, \bar{X}\}$ into the problem. We can obtain the fractional Y^{**}. This ensures Constraint (1a) $\sim$ (1d), (1i), (1m) is satisfied. Therefore, we get the final solution $\{\bar{A}, \bar{X}, Y^{**}\}$. In every iteration, we always round a pair of fractions altogether, so that one or both of them can become integral while compensating each other and keeping their weighted sum constant before and after rounding [34]. We design Alg. 3 based on this idea.

The rounding procedure is in lines 13-30 of Alg. 3. Consider rounding $\hat{A}$, for example. Every iteration in the loop of lines 15-27 ensures the following: (i) either $\hat{a}^t_{u,i_1}$, or $\hat{a}^t_{u,i_2}$, or both are rounded into integer(s); (ii) we have $Z_{i_1}\theta'_{i_1 t} + Z_{i_2}\theta'_{i_2 t} = Z_{i_1}\theta_{i_1 t} + Z_{i_2}\theta_{i_2 t}$, no matter we choose line 19 or 20; (iii) the expectation of the integral $\bar{a}^t_{u,k}$ equals to the fractional $\hat{a}^t_{u,k}$, *i.e.*, $\mathbb{E}(\bar{a}^t_{u,k}) = \hat{a}^t_{u,k}, \forall i \in I \backslash I'_t$ —for example, if $\hat{a}^t_{u,i_2}$ becomes integral, then $\mathbb{E}(\bar{a}^t_{u,i_2}) = \frac{\omega_2}{\omega_1+\omega_2}(\hat{a}^t_{u,i_2} - \frac{Z_{i_1}}{Z_{i_2}}\omega_1) + \frac{\omega_1}{\omega_1+\omega_2}(\hat{a}^t_{u,i_2} - \frac{Z_{i_1}}{Z_{i_2}}\omega_2) = \hat{a}^t_{u,i_2}$.

Theorem 3. *We analyze the integrality gap incurred by Alg. 3. That is said, the constant r satisfies*

$$\mathbb{E}(\boldsymbol{P'_{t2}}(\{\bar{A}, \bar{X}, Y^{**}\})) \leq r\mathbb{E}(\boldsymbol{P'_{t2}}(\{\hat{A}, \hat{X}, \hat{Y}\})) \tag{17}$$

Proof. See Appendix. C. □

Overall Alternating Optimization-Based Algorithm. The overall alternating optimization-based algorithm is described in Alg. 4. The key idea is to iteratively optimize UAV trajectory (line 3), and service caching, UE-UAV association, task offloading (line 4), respectively. The convergence as well the complexity is theoretically analyzed as follows.

Theorem 4. *Alg. 4 converges with polynomial complexity.*

Algorithm 3 Dependent Rounding Algorithm, $\forall t$

```
Input: Fractional solution $\{\hat{A}, \hat{X}, \hat{Y}\}$.
Output: Solution $\{\bar{A}, \bar{X}, Y^{**}\}$ for $\mathbf{P_{t2}}$.
    ▷ First, round $\hat{A}$.
 1: for $u = 1 : U$ do
 2:     Denote $\bar{a}^t_{u,k}$ as $\bar{z}^t_k$, $\hat{a}^t_{u,k}$ as $\hat{z}^t_k$, and $c_k$ as $Z_k$, $\forall k$;
 3:     $\mathcal{I} = \mathcal{K}$;
 4:     Execute Line 13 through 30, and continue with Line 6
 5: end for
    ▷ Then, based on $\bar{A}$, obtain $\{X^*, Y^*\}$, and round $X^*$.
 6: Fix $\bar{A}$, solve $\mathbf{P'_{t2}}$ and get its solution $\{\bar{A}, X^*, Y^*\}$.
 7: for $m = 1 : M$ do
 8:     Denote $\bar{x}^t_{u,m}$ as $\bar{z}^t_u$, $\hat{x}^t_{u,m}$ as $\hat{z}^t_u$, and 1 as $Z_u$, $\forall u$;
 9:     $\mathcal{I} = \mathcal{U}$;
10:     Execute Line 13 through 30, and continue with Line 12
11: end for
    ▷ Finally, based on $\bar{A}$ and $\bar{X}$, obtain $Y^{**}$.
12: Fix $\{\bar{A}, \bar{X}\}$, solve $\mathbf{P_{t2}}$ and return its solution $\{\bar{A}, \bar{X}, Y^{**}\}$.
13: $\theta^t_i \overset{\text{def}}{=} \hat{z}^t_i, \forall i$;
14: $\mathcal{I}'_t \overset{\text{def}}{=} \mathcal{I} \setminus \{i | \theta^t_i \in \{0, 1\}\}$;
15: while $|\mathcal{I}'_t| > 1$ do
16:     Select $i_1, i_2 \in \mathcal{I}'_t$, where $i_1 \neq i_2$;
17:     $\omega_1 \overset{\text{def}}{=} \min\{1 - \theta^t_{i_1}, \frac{Z_{i_2}}{Z_{i_1}} \theta^t_{i_2}\}$;
18:     $\omega_2 \overset{\text{def}}{=} \min\{\theta^t_{i_1}, \frac{Z_{i_2}}{Z_{i_1}} (1 - \theta^t_{i_2})\}$;
19:     With the probability $\frac{\omega_2}{\omega_1 + \omega_2}$,
          Set $\theta^t_{i_1} = \theta^t_{i_1} + \omega_1, \theta^{\prime t}_{i_2} = \theta^t_{i_2} - \frac{Z_{i_1}}{Z_{i_2}} \omega_1$;
20:     With the probability $\frac{\omega_1}{\omega_1 + \omega_2}$,
          Set $\theta^t_{i_1} = \theta^t_{i_1} - \omega_2, \theta^t_{i_2} = \theta^t_{i_2} + \frac{Z_{i_1}}{Z_{i_2}} \omega_2$;
21:     if $\theta^t_{i_1} \in \{0, 1\}$ then
22:         Set $\bar{z}^t_{i_1} = \theta^t_{i_1}, \mathcal{I}'_t = \mathcal{I}'_t \setminus \{i_1\}$;
23:     end if
24:     if $\theta^t_{i_2} \in \{0, 1\}$ then
25:         Set $\bar{z}^t_{i_2} = \theta^t_{i_2}, \mathcal{I}'_t = \mathcal{I}'_t \setminus \{i_2\}$;
26:     end if
27: end while
28: if $|\mathcal{I}'_t| = 1$ then
29:     Set $\bar{z}^t_i = 0$ for the only $i \in \mathcal{I}'_t$;
30: end if
```

Proof. See Appendix. D. □

D. Algorithm Design for Time Slot τ

The algorithm for $\mathbf{P}_\tau$ is similar to algorithm for $\mathbf{P_t}$. For time slot $\tau \in [t+1, t+T-1]$, the energy virtual queue $\mathbf{Q}(t)$ and the service cache decision A are deterministic. Hence, the rest variables are $\{X, Y, G\}$. We change the input and output of Alg. 4 from $\{A, X, Y, G\}$ to $\{X, Y, G\}$. At the same time, we impose the predicted energy virtual queue $\mathbf{Q}(t)$ to the objective function during $[t+1, t+T-1]$.

V. Numerical Evaluation

A. Evaluation Setup

This section provides numerical results to evaluate TJSO. Following the similar setting in [13], we consider a UAV-assisted MEC network with 3 UAVs ($U = 3$) and 20 UEs ($M = 20$), where these UEs are randomly distributed in a square by four coordinates $[0, 0]$, $[500, 0]$, $[0, 500]$, and $[500, 500]$ m. These UAVs can provide $K = 10$ types of services for UEs. For different types of service cache, its required storage size $c_k \in [0.5, 1]$, while the storage capacity of each UAV $C_u = 3$. The demand for service k generated from UE m during slot t is formulated as a Poisson process with arrival rate $d^t_{m,k} \in [0, 6]$. We suppose there are $\tilde{T} = 100$ time slots. The slot length is 1 minute. The update interval of cache decisions is $T = 5$ time slots. Other decision variables are updated every time slot. Each UE generates a computation-intensive task with input data size $l_k \in [10, 100]$ KB, and requires $\mu_k \in [10^8, 10^9]$ number of CPU cycles. For UAVs, the related parameters are set as follows: the fixed altitude $H_u = 100$ m, maximum coverage range $R^{max}_u = 300$ m, maximum moving distance of one slot $d^{max}_u = 30$ m, maximum number of UEs associated to each UAV $N^{max}_u \in [1, 5]$, and computation capacity of each UAV $f_u \in [10, 20]$ GHz. We suppose that the initial coordinates of UAVs are randomly distributed in this square.

Algorithm 4 Iterative Algorithm

```
Input: Feasible $\{A^0, X^0, Y^0, G^0\}$, tolerance $\epsilon > 0$, $i = 1$,
$I^{max} = 100$.
Output: Service cache $A$, task offloading $X$, task offloading $Y$,
UAV trajectory $G$.
Compute $V^0 = Obj(A^0, X^0, Y^0, G^0)$, where $Obj(A, X, Y, G)$
  is the objective function in problem $\mathbf{P_t}$.
while $i < I^{max}$ do
    With fixed $\{A^{i-1}, X^{i-1}, Y^{i-1}\}$, obtain $G^i$ by Alg. 2.
    With fixed $G^i$, obtain $\{A^i, X^i, Y^i\}$ by Alg. 3.
    if $|V^i - V^{i-1}|/V^{i-1} \leq \epsilon$ then
        Break.
    else
        Set $i \leftarrow i + 1$.
    end if
end while
Return $\{A^i, X^i, Y^i, G^i\}$.
```

We compare TJSO to three benchmarks:

- **Random**: the cache decisions and the trajectory decisions are randomly made. UEs offload as many tasks as possible to UAVs. All of decisions are made under constraints.
- **Delay central algorithm**: without considering energy consumption constraints, all decisions are made only to minimize the service delay.
- **Myopic algorithm**: myopic algorithm imposes a stricter constraint to energy consumption during every time slot. The long-term energy constraint can be always satisfied without establishing deficit energy queue. All decisions are made to meet the stricter constraint and minimize service delay.

B. Evaluation Results

Time Average Delay/Energy Consumption. Fig. 3 and Fig. 4 show the time average service delay and time average UAVs' total energy consumption, respectively. It shows that TJSO achieves near-to-optimal delay performance while closely following the long-term energy constraint. TJSO consumes 24% less energy than delay central algorithm after convergence, and slightly sacrifices the delay performance to satisfy the energy

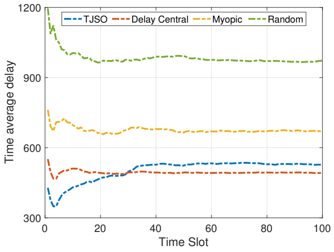

Fig. 3: Time average delay.

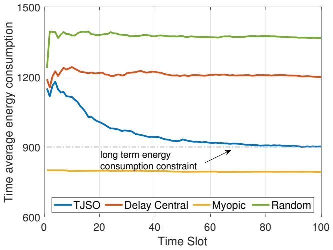

Fig. 4: Time average energy consumption.

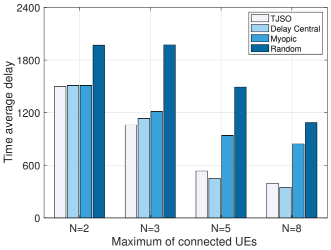

Fig. 5: Time average delay with different communication access ability N_u^{max}.

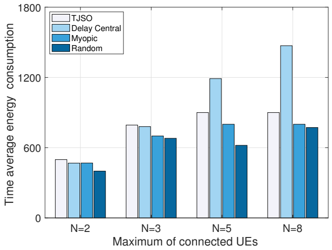

Fig. 6: Time average delay with different communication access ability N_u^{max}.

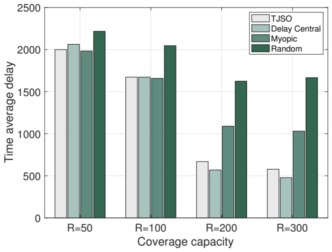

Fig. 7: Time average energy consumption with different communication access ability N_u^{max}.

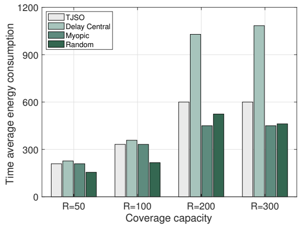

Fig. 8: Time average delay with different coverage range R_u^{max}.

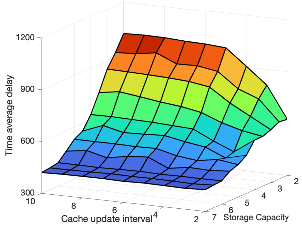

Fig. 9: Time average energy consumption with different coverage range R_u^{max}.

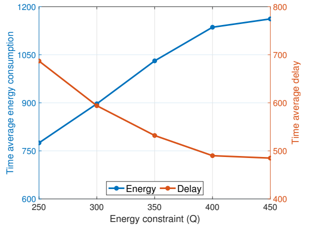

Fig. 10: Time average energy consumption with different coverage range R_u^{max}.

consumption constraint. For the myopic algorithm, due to the low task arrival rate in some time slots, the time average energy consumption can be below the long-term constraint, which leads to inefficient energy usage. The time average delay of myopic algorithm is 28% higher than that of TJSO. For random algorithm, the performance of random algorithm is poor both in service delay and energy consumption. There is a big gap between the random algorithm with the other three algorithms.

Impact of Communication Access Ability. Fig. 5 and Fig. 6 show the impact of communication access ability (N_u^{max}) on the converged service delay and energy consumption after 100 time slots. It can be observed that the service delay decreases when the communication access capability increases since a larger number of UEs can be connected to UAVs. Moreover, when the communication access ability is small, the service delay achieved by TJSO is almost same with the delay central algorithm, because energy queue is zero in most of time. As the communication access ability increases, the service delay of TJSO is larger than that of the delay central algorithm to meet the energy consumption constraint.

Impact of Coverage Range. The analysis of the effect on coverage range is similar to that of communication ability. Fig. 7 and Fig. 8 illustrates the service delay and UAVs' energy consumption versus UE's coverage range. Compared with communication access ability, our algorithm is more sensitive to the coverage range. With the increase of the coverage range of each UAV, the delay performance of the algorithm will be greatly improved.

Impact of Storage Capacity and Cache Update Interval. Fig. 9 illustrates the joint impact of storage capacity and cache update interval on the delay performance. The more kinds of service contents are cached on the UAV, the more likely the task is to be processed on the UAV. If storage capacity C_u is larger than 3, the impact of cache update interval on service delay is very slight. Therefore, the two time-scale cache update algorithm can help to save energy while ensuring the delay performance.

Impact of Energy Constraint. Fig. 10 presents the converged time average service delay and energy consumption of TJSO under different energy consumption constraints Q. When energy consumption constraints Q is small, the energy allocated to UAV is fully utilized. We observe that the TJSO converges to a lower service delay with a larger Q since more tasks are offloaded to UAVs rather than the remote cloud for processing. However, the time average delay decreases slowly when the constraint Q is large enough.

VI. Conclusion

This paper studied the joint service caching and task offloading problem for UAV-assisted wireless networks in a two time-scale system. The most important difference of this problem lies in the caching decision and the limited energy capacity of UAVs, bringing new challenges in the algorithm design. The goal is to minimize the overall service delay, with the constraints of energy consumption, UAV's mobility and network resources, which needs to jointly optimize service caching placement, UAV trajectory, UE-UAV association, and task offloading. Based on Lyapunov optimization approach, we proposed an iterative algorithm with theoretical performance guarantee. Large scale simulations shows that our algorithm can reduce the service delay by up to 70%, compared with three benchmarks, while keeping energy consumption low. There are a few limitations in the current model that will be studied in our future work. First, the switch cost incurred by the replacement of caching contents haven't been incorporated in the total energy consumption. Second, each UAV's computation capacity can be modeled as an decision variable to further optimize the overall performance.

Appendix

A. Proof of Theorem 1

First we prove that if we have the queue dynamics:

$$U(t+1) = \{U(t) - \mu(t), 0\} + A(t)$$

then we have:

$$\begin{aligned}U^2(t+1) =&[\max\{U(t) - \mu(t), 0\}]^2 + A^2(t)\\ &+ 2A(t)\max\{U(t) - \mu(t), 0\}\\ \le& U^2(t) + \mu^2(t) - 2U(t) + \mu(t) + A^2(t)\\ &+ 2A(t)\max\{U(t) - \mu(t), 0\}\\ \le& U^2(t) + \mu^2(t) - 2U(t) + \mu(t) + A^2(t) + 2A(t)U(t)\\ =& U^2(t) + \mu^2(t) + A^2(t) + 2U(t)[A(t) - \mu(t)]\end{aligned}$$

where the first inequity comes from $(\max\{x, 0\})^2 \le x^2$ and the second inequity is always true when $A(t) \ge 0$ and μ_t. So, applying this property to $Q_u(\tau)$, we have:

$$Q_u^2(\tau+1) - Q_u^2(\tau) \le (E_u^t)^2 + Q^2 + 2Q_u(\tau)[E_u^t - Q]$$

Because $E_u^t \le E^{max}$, then

$$Q_u^2(\tau+1) - Q_u^2(\tau) \le (E^{max})^2 + Q^2 + 2Q_u(\tau)[E_u^t - Q]$$

We can derive the drift-plus-penalty term:

$$\begin{aligned}&\Delta_T(\mathbf{Q}(t)) + V\mathbb{E}\{\sum_{\tau=t}^{t+T-1} D^\tau|\mathbf{Q}(t)\}\\ &= \mathbb{E}[L(\mathbf{Q}(t+T)) - L(\mathbf{Q}(t))|\mathbf{Q}(t)] + V\mathbb{E}\{\sum_{\tau=t}^{t+T-1} D^\tau|\mathbf{Q}(t)\}\\ &= \frac{1}{2}\sum_{\tau=t}^{t+T-1}\sum_u[Q_u^2(t+1) - Q_u^2(t)] + V\mathbb{E}\{\sum_{\tau=t}^{t+T-1} D^\tau|\mathbf{Q}(t)\}\\ &\le V\mathbb{E}\{\sum_{\tau=t}^{t+T-1} D^\tau|\mathbf{Q}(t)\}\\ &+ \frac{1}{2}\sum_{\tau=t}^{t+T-1}\sum_u\{(E^{max})^2 + Q^2 + 2Q_u(\tau)[E_u^t - Q]\}\end{aligned}$$

Simplifying the above inequality leads to our theorem 1:

$$\begin{aligned}&\Delta_T(\mathbf{Q}(t)) + V\mathbb{E}\{\sum_{\tau=t}^{t+T-1} D^\tau|\mathbf{Q}(t)\}\\ &\le B_1T + V\mathbb{E}\{\sum_{\tau=t}^{t+T-1} D^\tau|\mathbf{Q}(t)\} + \sum_{\tau=t}^{t+T-1}\sum_u\{Q_u(\tau)[E_u^t - Q]\}\end{aligned}$$

where $B_1 = \frac{1}{2}\sum_u[(E^{max})^2 + Q^2]$.

B. Proof of Theorem 2

For any $\tau \in [t, t+T-1]$, we can get:

$$Q_u(\tau) \le Q_u(t) + (\tau - t)E^{max}$$

Therefore, applying the first inequality to $Q_u(t)$, we have:

$$\begin{aligned}&\sum_{\tau=t}^{t+T-1} Q_u(\tau)[E_u^t - Q]\\ &\le \sum_{\tau=t}^{t+T-1} Q_u(t)[E_u^t - Q] + \sum_{\tau=t}^{t+T-1}(\tau - t)E^{max}[E_u^t - Q]\\ &\le \sum_{\tau=t}^{t+T-1} Q_u(t)[E_u^t - Q] + \sum_{\tau=t}^{t+T-1}(\tau - t)E^{max}[E^{max} - Q]\end{aligned}$$

where $B_2 = B_1 + \frac{T-1}{2}\sum_u E^{max}[E^{max} - Q]$.

C. Proof of Theorem 3

Problem $\mathbf{P_{t2}}$ is a continuous linear program (LP) problem, and can be reformulated as follows:

$$\begin{aligned}\min_{A,X,Y} \quad & \sum_u(\sum_m\sum_k \Phi_{u,m,k}^t y_{u,m,k}^t + \sum_k \Gamma_k a_{u,k}^t)Q_u(t)\\ & + V\sum_u\sum_m\sum_k \Psi_{u,m,k}^t y_{u,m,k}^t\end{aligned}$$

where $\Phi_{u,m,k}^t = d_{m,k}^t(\kappa\mu_k + \frac{p_{u,m}l_k}{r_{u,m}^t}), \forall u, \forall m, \forall k; \Gamma_k = \gamma c_k, \forall k; \Psi_{u,m,k}^t = (\frac{l_k}{r_{u,m}^t} + \frac{\mu_k}{f_u} - \frac{l_k}{R})d_{m,k}^t, \forall u, \forall m, \forall k.$ According to alg. 3, we can derive $\sum_k \bar{a}_{u,k}^t c_k = \sum_{k\in\mathcal{K}\setminus\mathcal{K}'} \bar{a}_{u,k}^t c_k \le \sum_{k\in\mathcal{K}} \hat{a}_{u,k}^t c_k$. From constraint (1b), we can get the inequality $y_{u,m,k}^t \le \bar{a}_{u,k}^t$, and then we derive $\sum_u\sum_m\sum_k y_{u,m,k}^t \le M\sum_u\sum_k \bar{a}_{u,k}^t$. Afterwards, we consider $\sum_u\sum_m\sum_k \Phi_{u,m,k}^t y_{u,m,k}^t$ below. Because $y_{u,m,k}^{t**}$ is not randomized, its "expectation" is itself. $\sum_u\sum_m\sum_k \Phi_{u,m,k}^t y_{u,m,k}^{t**}Q_u(t)$ $\le \max_{u,m,k}\Phi_{u,m,k}^t\sum_u\sum_m\sum_k y_{u,m,k}^{t**}Q_u(t) \le \frac{\max_{u,m,k}\Phi_{u,m,k}^t}{\min_k c_k}\cdot$ $M\sum_u\sum_k \bar{a}_{u,k}^t Q_u(t) \le \frac{\max_{u,m,k}\Phi_{u,m,k}^t}{\min_k c_k}M\sum_u\sum_k \hat{a}_{u,k}^t c_k Q_u(t)$ $\le \delta_{y1}\sum_u\sum_k \hat{a}_{u,k}^t c_k$, where $\delta_{y1} = MQ_u(t)\frac{\max_{u,m,k}\Phi_{u,m,k}^t}{\min_k c_k}$. Similarly, $V\sum_u\sum_m\sum_k \Psi_{u,m,k}^t y_{u,m,k}^{t**} \le \delta_y\sum_u\sum_k \hat{a}_{u,k}^t c_k$, where $\delta_{y2} = MV\frac{\max_{u,m,k}\Psi_{u,m,k}^t}{\min_k c_k}$. Finally, we obtain $\mathbb{E}(\mathbf{P'_{t2}}(\{\bar{A}, \bar{X}, Y^{**}\})) \le r\mathbb{E}(\mathbf{P'_{t2}}(\{\hat{A}, \hat{X}, \hat{Y}\}))$, where $r = \gamma + \delta_{y1}$ $+\delta_{y2}$.

D. Proof of Theorem 4

For the convergence, we first need to prove that, when the sequence (A_i, X_i, Y_i) is updated in the algorithm, the objective function $Obj(A_i, X_i, Y_i, P_i)$ keeps non-increasing. By Alg. 2, we have

$$\begin{aligned}V^{i-1} &= Obj(A^{i-1}, X^{i-1}, Y^{i-1}, P^{i-1})\\ &\ge Obj(A^{i-1}, X^{i-1}, Y^{i-1}, P^i)\\ &\ge Obj(A^i, X^i, Y^i, P^i) = V^i\end{aligned}$$

where the first inequality holds because of the sub-optimality of $\{A^i, X^i, Y^i\}$ by dependent rounding. The third inequality is due to the sub-optimality of UAV trajectory P^i. Moreover, the objective function $Obj(X, Z, G, C)$ is always positive owing to its non-negative expression. Therefore, the objective function is always non-increasing after every iteration, which is also finitely lower-bounded by zero. Furthermore, according to the analysis in [13], in Problem $\mathbf{P'_{t1}}$, it takes $O(J^{max}U^3M^3)$ time, where J^{max} is the number of iterations of the SCA-based algorithm. Problem $\mathbf{P'_{t2}}$ involves the solving of the LP problem, whose computation complexity is about $O(U^3M^3K^3)$ by the well known Interior Point Method [33]. Using the dependent rounding Alg. 3, which $O(UA + UM)$ time. To summarize, the total complexity of Alg. 4 is $O(I^{max}(J^{max}U^3M^3 + U^3M^3K^3 + U(A + M)))$, where I^{max} is the number of outer iterations of Alg. 4. The above complexity is polynomial, which concludes this theorem.

References

[1] M. Li, P. Si, and Y. Zhang, “Delay-tolerant data traffic to software-defined vehicular networks with mobile edge computing in smart city,” *IEEE Transactions on Vehicular Technology*, vol. 67, no. 10, pp. 9073–9086, 2018.

[2] K. Zhang, S. Leng, Y. He, S. Maharjan, and Y. Zhang, “Cooperative content caching in 5g networks with mobile edge computing,” *IEEE Wireless Communications*, vol. 25, no. 3, pp. 80–87, 2018.

[3] Y. Liu, K. Xiong, Q. Ni, P. Fan, and K. B. Letaief, “Uav-assisted wireless powered cooperative mobile edge computing: Joint offloading, cpu control, and trajectory optimization,” *IEEE Internet of Things Journal*, vol. 7, no. 4, pp. 2777–2790, 2019.

[4] J. Wang, K. Liu, and J. Pan, “Online uav-mounted edge server dispatching for mobile-to-mobile edge computing,” *IEEE Internet of Things Journal*, vol. 7, no. 2, pp. 1375–1386, 2019.

[5] B. Li, Z. Fei, and Y. Zhang, “Uav communications for 5g and beyond: Recent advances and future trends,” *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 2241–2263, 2018.

[6] huaxia, “China deploys uav for telecom restoration in rain-hit henan,” http://www.xinhuanet.com/english/2021-07/22/c_1310078337.htm/ Accessed July 22, 2021.

[7] V. Farhadi, F. Mehmeti, T. He, T. F. La Porta, H. Khamfroush, S. Wang, K. S. Chan, and K. Poularakis, “Service placement and request scheduling for data-intensive applications in edge clouds,” *IEEE/ACM Transactions on Networking*, vol. 29, no. 2, pp. 779–792, 2021.

[8] J. Xu, L. Chen, and P. Zhou, “Joint service caching and task offloading for mobile edge computing in dense networks,” in *Proc. of IEEE INFOCOM*, 2018.

[9] T. Ouyang, Z. Zhou, and X. Chen, “Follow me at the edge: Mobility-aware dynamic service placement for mobile edge computing,” *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 10, pp. 2333–2345, 2018.

[10] Y. Guo, Q. Yang, F. R. Yu, and V. C. Leung, “Cache-enabled adaptive video streaming over vehicular networks: A dynamic approach,” *IEEE Transactions on Vehicular Technology*, vol. 67, no. 6, pp. 5445–5459, 2018.

[11] J. Johnson, E. Basha, and C. Detweiler, “Charge selection algorithms for maximizing sensor network life with uav-based limited wireless recharging,” in *2013 IEEE eighth international conference on intelligent sensors, sensor networks and information processing*. IEEE, 2013.

[12] Z. Yang, C. Pan, K. Wang, and M. Shikh-Bahaei, “Energy efficient resource allocation in uav-enabled mobile edge computing networks,” *IEEE Transactions on Wireless Communications*, vol. 18, no. 9, pp. 4576–4589, 2019.

[13] L. Wang, K. Wang, C. Pan, W. Xu, N. Aslam, and A. Nallanathan, “Deep reinforcement learning based dynamic trajectory control for uav-assisted mobile edge computing,” *IEEE Transactions on Mobile Computing*, 2021.

[14] Z. Xu, L. Zhou, S. Chi-Kin Chau, W. Liang, Q. Xia, and P. Zhou, “Collaborate or separate? distributed service caching in mobile edge clouds,” in *Proc. of IEEE INFOCOM*, 2020.

[15] X. Ma, A. Zhou, S. Zhang, and S. Wang, “Cooperative service caching and workload scheduling in mobile edge computing,” in *Proc. of IEEE INFOCOM*, 2020.

[16] Y. Ji, Z. Yang, H. Shen, W. Xu, K. Wang, and X. Dong, “Multicell edge coverage enhancement using mobile uav-relay,” *IEEE Internet of Things Journal*, vol. 7, no. 8, pp. 7482–7494, 2020.

[17] J. Ji, K. Zhu, D. Niyato, and R. Wang, “Joint cache placement, flight trajectory, and transmission power optimization for multi-uav assisted wireless networks,” *IEEE Transactions on Wireless Communications*, vol. 19, no. 8, pp. 5389–5403, 2020.

[18] M. Alzenad, A. El-Keyi, F. Lagum, and H. Yanikomeroglu, “3-d placement of an unmanned aerial vehicle base station (uav-bs) for energy-efficient maximal coverage,” *IEEE Wireless Communications Letters*, vol. 6, no. 4, pp. 434–437, 2017.

[19] J. Zhang, L. Zhou, Q. Tang, E. C.-H. Ngai, X. Hu, H. Zhao, and J. Wei, “Stochastic computation offloading and trajectory scheduling for uav-assisted mobile edge computing,” *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 3688–3699, 2019.

[20] M. Chen, M. Mozaffari, W. Saad, C. Yin, M. Debbah, and C. S. Hong, “Caching in the sky: Proactive deployment of cache-enabled unmanned aerial vehicles for optimized quality-of-experience,” *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 5, pp. 1046–1061, 2017.

[21] J. Lyu, Y. Zeng, and R. Zhang, “Uav-aided offloading for cellular hotspot,” *IEEE Transactions on Wireless Communications*, vol. 17, no. 6, pp. 3988–4001, 2018.

[22] Q. Hu, Y. Cai, G. Yu, Z. Qin, M. Zhao, and G. Y. Li, “Joint offloading and trajectory design for uav-enabled mobile edge computing systems,” *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 1879–1892, 2019.

[23] C. Zhan, H. Hu, X. Sui, Z. Liu, and D. Niyato, “Completion time and energy optimization in the uav-enabled mobile-edge computing system,” *IEEE Internet of Things Journal*, vol. 7, no. 8, pp. 7808–7822, 2020.

[24] Q. Wu, Y. Zeng, and R. Zhang, “Joint trajectory and communication design for multi-uav enabled wireless networks,” *IEEE Transactions on Wireless Communications*, vol. 17, no. 3, pp. 2109–2121, 2018.

[25] R. Urgaonkar, S. Wang, T. He, M. Zafer, K. Chan, and K. K. Leung, “Dynamic service migration and workload scheduling in edge-clouds,” *Performance Evaluation*, vol. 91, pp. 205–228, 2015.

[26] L. Pu, L. Jiao, X. Chen, L. Wang, Q. Xie, and J. Xu, “Online resource allocation, content placement and request routing for cost-efficient edge caching in cloud radio access networks,” *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 8, pp. 1751–1767, 2018.

[27] K. Poularakis, J. Llorca, A. M. Tulino, I. Taylor, and L. Tassiulas, “Joint service placement and request routing in multi-cell mobile edge computing networks,” in *Proc. of IEEE INFOCOM*, 2019.

[28] L. Wang, L. Jiao, T. He, J. Li, and M. Mühlhäuser, “Service entity placement for social virtual reality applications in edge computing,” in *Proc. of IEEE INFOCOM*, 2018.

[29] Y. Zhang, L. Jiao, J. Yan, and X. Lin, “Dynamic service placement for virtual reality group gaming on mobile edge cloudlets,” *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 8, pp. 1881–1897, 2019.

[30] J. Xu, L. Chen, and P. Zhou, “Joint service caching and task offloading for mobile edge computing in dense networks,” in *Proc. of IEEE INFOCOM*, 2018.

[31] T. Ouyang, Z. Zhou, and X. Chen, “Follow me at the edge: Mobility-aware dynamic service placement for mobile edge computing,” *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 10, pp. 2333–2345, 2018.

[32] S. Bi, L. Huang, and Y.-J. A. Zhang, “Joint optimization of service caching placement and computation offloading in mobile edge computing systems,” *IEEE Transactions on Wireless Communications*, vol. 19, no. 7, pp. 4947–4963, 2020.

[33] S. Boyd, S. P. Boyd, and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[34] R. Gandhi, S. Khuller, S. Parthasarathy, and A. Srinivasan, “Dependent rounding and its applications to approximation algorithms,” *Journal of the ACM*, vol. 53, no. 3, pp. 324–360, 2006.

[35] X. Chen, W. Ni, T. Chen, I. B. Collings, X. Wang, and G. B. Giannakis, “Real-time energy trading and future planning for fifth generation wireless communications,” *IEEE Wireless Communications*, vol. 24, no. 4, pp. 24–30, 2017.