Reliability-Aware Offloading in UAV-Aided Mobile Edge Network by Lyapunov Optimization

Jingjing Yao*, Semih Cal*, and Xiang Sun†
*Department of Computer Science, Texas Tech University, Lubbock, TX 79409, USA.
†Department of Electrical and Computer Engineering, University of New Mexico, Albuquerque, NM 87131, USA.

***Abstract*—Unmanned aerial vehicles (UAVs) have become increasingly popular in providing services to mobile users, as they can enhance network capacity and coverage. UAV-aided mobile edge networks leverage UAVs as edge servers to provide computing resources such that mobile users can offload their computation-intensive tasks to the UAVs. To balance the load among multiple UAVs, an intelligent task offloading scheme is required that can distribute tasks from users to different UAVs while guaranteeing network latency. Moreover, as computation-intensive tasks are energy-consuming, an energy-efficient scheme that can minimize the energy consumption of offloading the tasks is essential. Additionally, UAV-aided edge networks may experience disturbances caused by UAV and communication link failures, making reliability performance a critical consideration. Therefore, in this paper, we investigate the energy-efficient task offloading problem in UAV-aided edge networks with guaranteed reliability and latency. We formulate this problem as an integer programming problem and propose a Lyapunov optimization algorithm to efficiently solve the problem. Simulation results are conducted to demonstrate the feasibility and superiority of our proposed algorithm.**

***Index Terms*—Mobile edge computing, Unmanned Aerial Vehicle (UAV), computation offloading, quality of service (QoS), reliability**

I. Introduction

Emerging applications such as virtual reality, augmented reality, and autonomous vehicles are becoming increasingly time-sensitive and energy-consuming, presenting challenges for mobile users [1]. For example, virtual reality applications have strict latency requirements of less than 15 *ms* for rendering images, to prevent users from feeling dizzy or detached [2]. However, the limited resources of mobile devices, including computing capability and battery power, make it challenging to implement these applications. The traditional approach is to rely on remote cloud resources, which can lead to long communication latency. Mobile edge computing (MEC), which distributes computing and storage resources at the network edge (e.g., base stations and wireless access points), is considered one of the promising solutions to address this challenge. Computation-intensive tasks generated by mobile users can be offloaded to proximal MEC servers for processing [3], which reduces network latency and energy consumption of mobile devices.

Unmanned aerial vehicles (UAVs), also known as drones, have garnered significant attention from academia and industry due to their mobility and deployment flexibility [4]. UAVs can function as temporary edge servers in mobile edge networks, particularly in areas with limited communication such as emergency rescue or sports events stadiums, to enhance network capacity [5]. One of the key advantages of UAV-aided mobile edge networks is that UAVs can dynamically adjust their positions based on network traffic to provide better services. This flexibility enables UAVs to serve as mobile edge servers that can move to the locations of mobile users, providing computing resources and reducing communication latency.

To balance the load across different UAVs, an effective task offloading scheme that distributes tasks from different users to their appropriate UAVs is required to meet quality of service (QoS) requirements, such as task completion time [1]. However, intensive computations are energy-intensive and can rapidly drain the limited UAV batteries. Therefore, an energy-efficient task offloading scheme with guaranteed QoS performance is essential to address these challenges.

In practice, UAV-aided edge networks may face inevitable disturbances such as UAV failures due to software or hardware breakdowns and communication link failures between mobile users and UAVs caused by various communication errors. Under such circumstances, achieving reliable workflow becomes a crucial challenge [6]. Therefore, in addition to considering latency guarantees, a reliable mechanism is essential to ensure the reliability of task offloading.

In this paper, we investigate the energy-efficient task offloading with guaranteed reliability and QoS in UAV-aided mobile edge networks. Furthermore, we propose a Lyapunov optimization algorithm to make effective offloading decisions. Simulation results demonstrate the performances of our proposed algorithm.

This paper is structured as follows. In Section II, we review the related work. Section III presents the analysis of our system model. In Section IV, we formulate our problem. Section V proposes a Lyapunov optimization algorithm to effectively address our problem. Section VI presents the simulation results and analysis. Finally, Section VII concludes the paper.

II. Related Work

MEC has been the focus of many research studies. Mao *et al.* [1] conducted a survey of recent MEC research from the perspective of radio and computational resource management and provided insights into the challenges and future research directions of MEC. Sun and Ansari [7] proposed the EdgeIoT architecture to handle data streams at the mobile edge. The

979-8-3503-7099-7/24/$31.00 ©2024 IEEE

architecture adopts a hierarchical fog computing architecture to provide network services while maintaining user privacy. Li *et al.* [8] investigated the cooperative offloading problem in MEC networks, where resource-constrained edge clouds help each other with computation-intensive tasks. They proposed an online learning method based on social trust to minimize system cost.

Several studies have investigated the task offloading problem in UAV-aided edge networks. For example, Wang *et al.* [5] focused on the computation offloading problem in fog computing enabled vehicle networks, aiming to ensure reliable communication and low latency. Wu *et al.* [9] addressed the joint deployment and UAV flight trajectory algorithm in UAV-assisted vehicle edge networks with the objective of minimizing the cost of UAVs, while considering both UAV flying and turning energy costs. They proposed a deep reinforcement learning algorithm to obtain an energy-efficient autonomous deployment strategy. Yang *et al.* [10] investigated the load balancing problem in multi-UAV-aided MEC systems to offload tasks to different UAVs while guaranteeing the coverage constraints and the QoS of mobile devices. However, the reliability performance in UAV-aided MEC networks is not considered in the above works.

Liu and Qi [11] proposed an offloading strategy in mobile edge networks to minimize the task offloading failure probability subject to service latency constraints. Yao and Ansari [12] investigated the tradeoff between maximizing system reliability and minimizing the system cost by designing the fog resource provisioning strategy in fog-aided networks. Huang *et al.* [13] studied the reliability-aware network function virtualization instance provisioning problem in MEC networks where different users request different network services with different reliability requirements with the objective to maximize the network throughput. Hou *et al.* [14] designed a fault-tolerant particle swarm optimization algorithm for task allocation problem in software-defined and edge-computing-aided internet of vehicles to maximize the reliability with latency constraints. However, none of the above works consider the reliability-aware task offloading problem in UAV-aided mobile edge networks. To fill the gap in the existing research, we investigate the energy-efficient task offloading in UAV-aided mobile edge networks with guaranteed reliability and latency constraints in this work.

III. System Model

In the considered UAV-aided mobile edge networks, depicted in Fig. 1, N mobile users offload their computing tasks to M drones, which provide computing resources and are assumed to be stationary in the air. The index sets of mobile devices and UAVs are denoted by $\mathcal{N} = 1, 2, \ldots, N$ and $\mathcal{M} = 1, 2, \ldots, M$, respectively. Each mobile user is associated with a UAV, which processes its computing tasks. To model this association, a binary variable $x_{ij} \in \{0, 1\}$ is defined, where $x_{ij} = 1$ indicates that user i is associated with UAV j, and $x_{ij} = 0$ otherwise. To account for the inevitable disturbances in UAV-aided mobile edge networks, we assume that UAV j may fail with a rate of λ_j while executing tasks due to hardware failures or software errors [6]. Additionally, the wireless communication link between user i and UAV j may fail with a rate of λ'_{ij} due to communication errors.

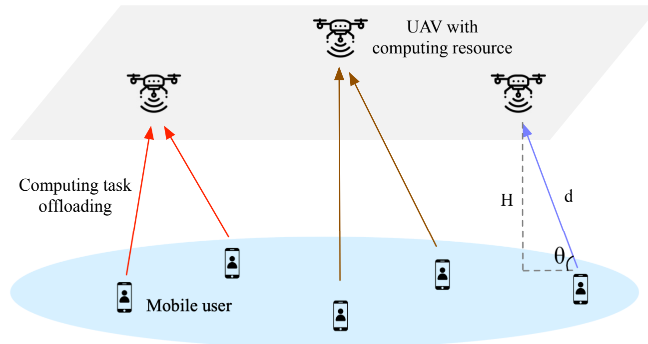

Fig. 1. UAV-aided edge networks.

A. UAV Air-to-Ground Channel Model

To characterize the air-to-ground channel between UAVs and mobile users, we adopt a widely used probability model that accounts for both line-of-sight (LoS) and non-line-of-sight (NLoS) scenarios [15]. Specifically, we assume that the wireless channel can be either LoS or NLoS, and that the probabilities of these scenarios depend on the elevation angle θ (as shown in Fig. 1). The probabilities of the channel being LoS or NLoS can be calculated as $\mathbb{P}(LoS) = \frac{1}{1+\alpha \exp(-\beta[\frac{180}{\pi}\theta-\alpha])}$ and $\mathbb{P}(NLoS) = 1 - \mathbb{P}(LoS)$, respectively, where α and β are constants that depend on the environment (e.g., rural or urban) [15].

The LoS and NLoS pathloss models are characterized by the free space propagation model, where the LoS pathloss is $PL_{LoS} = 20\log_{10}(\frac{4\pi f_c d}{c}) + \xi_{LoS}$ and the NLoS pathloss is $PL_{NLoS} = 20\log_{10}(\frac{4\pi f_c d}{c}) + \xi_{NLoS}$, where f_c is the carrier frequency, d is the distance between a UAV and mobile user, and ξ_{LoS} and ξ_{NLoS} are constants that depend on the environment [15]. By combining the LoS and NLoS pathloss models and their probabilities, the average pathloss model can be calculated as $\overline{PL} = \mathbb{P}(LoS)\cdot PL_{LoS} + \mathbb{P}(NLoS)\cdot PL_{NLoS}$. Then, the wireless channel between mobile user i and UAV j can be expressed as $h_{ij} = 10^{-\frac{\overline{PL}}{10}}$. According to the Shannon Equation, the wireless transmission rate from mobile user i to UAV j can be calculated as

$$r_{ij} = w_{ij}log_2(1 + \gamma_{ij}) = w_{ij}log_2(1 + \frac{p_i h_{ij}}{N_0 w_{ij}}), \quad (1)$$

where w_{ij} is the allocated bandwidth between user i and UAV j, γ_{ij} represent the signal-to-noise ratio (SNR) between user i and UAV j, h_{ij} indicates the wireless channel gain between user i and UAV j, N_0 is the noise power spectrum density, and p_i is the wireless transmission power of user i.

B. QoS Model

In UAV-aided mobile edge networks, the QoS of a user is determined by the network latency experienced by the user.

If the latency exceeds user i's completion deadline D_i, the user's experience is greatly affected. To model computing tasks, we adopt a widely used tuple notation $\langle l_i, v_i, D_i \rangle$, where l_i denotes the input data size in bits, v_i is the computation intensity in CPU cycles per bit, and D_i is the completion deadline in seconds [4]. Computing tasks are first offloaded from a user to a UAV, where the task is processed, and then the computing results are sent back to the user. The network latency experienced by a user usually consists of the task offloading time, the task processing time, and the result transmission time. As the results of tasks from UAVs are relatively small in size and the downlink capacity is typically much larger than the uploading capacity, we neglect the result transmission time in our work. The task offloading time from user i to UAV j can be calculated as $t_{ij}^w = \frac{l_i}{r_{ij}}$, where r_{ij} is the wireless transmission rate from user i to UAV j. UAV j's task processing time to execute a task from user i can be expressed as $t_{ij}^c = \frac{l_i v_i}{f_j}$, where $l_i v_i$ is the required CPU cycles to process the task from user i, and f_j is UAV j's CPU frequency in CPU cycles per second. In summary, the total task completion time experienced by user i, if it is associated with UAV j, can be expressed as

$$t_{ij} = t_{ij}^w + t_{ij}^c = \frac{l_i}{r_{ij}} + \frac{l_i v_i}{f_j}. \tag{2}$$

C. System Energy Model

In our system model, computation-intensive tasks are offloaded from users to UAVs for processing, which implies that the energy consumption of the system is composed of the energy used by users for task offloading and the energy used by UAVs for task processing. As we mentioned before that the results of tasks from the UAVs are usually small, and so we neglect the energy consumption for downloading the results from UAVs back to the users. Specifically, the energy consumption of user i for offloading a task to UAV j is given by $E_{ij}^w = p_i \frac{l_i}{r_{ij}}$, where p_i is the transmission power of user i and $\frac{l_i}{r_{ij}}$ is the wireless transmission time. Meanwhile, we adopt a widely used energy consumption model for a UAV to execute tasks, where the model assumes the energy consumption of a single CPU cycle is proportional to μf_j^2. Here, μ is a scalar and its value depends on the CPU switched capacitance of a UAV [16]. Thus, the energy consumption of UAV j for processing a task from user i is $E_{ij}^c = \mu f_j^2 l_i v_i$. In summary, the total system energy consumption for the task to be offloaded from user i and processed by UAV j is

$$E_{ij} = E_{ij}^w + E_{ij}^c = p_i \frac{l_i}{r_{ij}} + \mu f_j^2 l_i v_i. \tag{3}$$

Note that our analysis focuses solely on the energy consumed by the UAV during wireless transmission and communication, while disregarding the energy expended by the drone while hovering. This hovering energy is typically determined by the UAV's physical attributes like its weight and propellers, making it a constant factor. Hence, it has no impact on the solution to our optimization problem [17].

D. Reliability Model

In our work, we take into account both the reliability of UAVs and the reliability of the communication links in the system. During the processing of computing tasks, UAVs may experience hardware or software failures, while communication links between mobile users and UAVs may be disrupted by severe path loss, fading, or shadowing. We define the reliability of a UAV as the probability that the UAV is operational while processing tasks, and the reliability of a communication link as the probability that the link is functional during task offloading [18]. We adopt the Poisson process to model the failures of UAV j with the failure rate λ_j [19]. Therefore, the reliability of UAV j can be calculated as $R_j^{uav} = e^{-\lambda_j \sum_{i \in \mathcal{N}} x_{ij} t_{ij}^c} = e^{-\lambda_j \sum_{i \in \mathcal{N}} x_{ij} \frac{l_i v_i}{f_j}}$, where x_{ij} is the binary variable which indicates whether user i is associated with UAV j [18]. The failures of each communication link are also assumed to be a Poisson process with the failure rate λ'_{ij} [19]. Hence, the reliability of communication link between user i and UAV j can be expressed as $R_{ij}^{com} = e^{-\lambda'_{ij} x_{ij} t_{ij}^w} = e^{-\lambda'_{ij} x_{ij} \frac{l_i}{r_{ij}}}$ [18]. To evaluate the overall system reliability, we adopt a commonly used model [18] that defines system reliability as the probability that all the UAVs in the system and communication links between the UAVs and users are functional. Thus, the system reliability is equal to the product of the reliabilities of UAVs and communication links. Therefore, the system reliability can be calculated by

$$\begin{aligned} R^{sys} &= \prod_{j \in \mathcal{M}} R_j^{uav} \prod_{i \in \mathcal{N},\ j \in \mathcal{M}} R_{ij}^{com} \\ &= e^{-\sum_{j \in \mathcal{M}} \lambda_j \sum_{i \in \mathcal{N}} x_{ij} \frac{l_i v_i}{f_j} - \sum_{i \in \mathcal{N}} \sum_{j \in \mathcal{M}} \lambda'_{ij} x_{ij} \frac{l_i}{r_{ij}}} \\ &= e^{-\sum_{i \in \mathcal{N}} \sum_{j \in \mathcal{M}} (\frac{\lambda_j l_i v_i}{f_j} + \frac{\lambda'_{ij} l_i}{r_{ij}}) x_{ij}}. \end{aligned} \tag{4}$$

IV. Problem Formulation

We present the formulation of the energy-efficient task offloading problem in UAV-assisted mobile edge networks, which considers both QoS and reliability. The problem can be mathematically formulated as:

$$\textbf{P0:} \min_{x_{ij}} \sum_{i \in \mathcal{N}} \sum_{j \in \mathcal{M}} E_{ij} x_{ij} \tag{5}$$

$$s.t., \sum_{j \in \mathcal{M}} t_{ij} x_{ij} \le D_i, \forall i \in \mathcal{N}, \tag{6}$$

$$R^{sys} \ge R^{th}, \tag{7}$$

$$\sum_{j \in \mathcal{J}} x_{ij} = 1,\ \forall i \in \mathcal{N}, \tag{8}$$

$$x_{ij} \in \{0, 1\},\ \forall i \in \mathcal{N},\ j \in \mathcal{M}. \tag{9}$$

The objective of problem **P0**, as defined in Eq. (5), is to minimize the overall energy consumption of the UAV-assisted mobile edge network by determining the optimal user-to-UAV associations represented by the binary variables x_{ij}. The QoS constraint is given by Eq. (6), which ensures that the completion time of each user's computing task does not exceed its respective deadline D_i. The system reliability constraint, expressed in Eq. (7), mandates that the overall system reliability exceeds the predetermined threshold R^{th}. The association constraint, as formulated in Eq. (8), guarantees that each user is associated with only one UAV. Finally, Eq. (9) constrains the user-to-UAV association variable to be a binary variable.

Plugging Eq. (4) into Eq. (7), we have $e^{-\sum_{i\in\mathcal{N}}\sum_{j\in\mathcal{M}}(\frac{\lambda_j l_i v_i}{f_j}+\frac{\lambda' ijl_i}{rij})x_{ij}} \geq R^{th}$, which can be further simplified as

$$\sum_{i\in\mathcal{N}}\sum_{j\in\mathcal{M}}(\frac{\lambda_j l_i v_i}{f_j}+\frac{\lambda' ijl_i}{rij})x_{ij} \leq \ln\frac{1}{R^{th}}. \quad (10)$$

For ease of notation, let $\Phi_{ij} = \frac{\lambda_j l_i v_i}{f_j}+\frac{\lambda' ijl_i}{rij}$. Hence, Eq. (10) can be transformed into

$$\sum_{i\in\mathcal{N}}\sum_{j\in\mathcal{M}}\Phi_{ij}x_{ij} \leq \ln\frac{1}{R^{th}}. \quad (11)$$

As a result, problem **P0** can be formulated as an integer linear programming (ILP) problem, which can be reduced to the generalized assignment problem (GAP), a well-known NP-hard problem [20]. In the next section, we will present an algorithm to solve this problem.

V. Proposed Algorithm

In this section, we present our algorithm designed to solve **P0** by using Lyapunov optimization. Our algorithm addresses both quality of service (QoS) and reliability by selecting the optimal UAV for each mobile user.

P0 is challenging due to the interdependence introduced by the reliability constraint (11), which couples the user and UAV pairs together. In other words, without the reliability constraint (11), **P0** could be easily solved by transforming it into N independent subproblems, where each subproblem involves selecting the UAV that generates the minimum energy consumption for a single mobile user while satisfying the time constraint (6). Therefore, this coupling makes the optimization problem more challenging and requires a more sophisticated approach to solve. Our algorithm is motivated by the need to break the coupling caused by the reliability constraint and address both the QoS and reliability constraints in a computationally efficient manner.

We utilize Lyapunov optimization which is typically used to optimize time-averaged objective functions in dynamic stochastic networks such as networks with random events and uncertainties [21]. To transform the static problem into a dynamic problem, we choose a user and UAV pair in each time slot while ensuring that the QoS and reliability constraints are satisfied. To do this, we introduce a reliability addition Φ_{ij} whenever we add a new user and UAV pair to our system in each time slot. The sum of all reliability additions over time must not exceed a reliability capacity threshold $\ln\frac{1}{R^{th}}$, as defined by the reliability constraint (11).

In Lyapunov optimization, virtual queues are used to characterize the event arrivals and departures. For our problem, we define the event arrival as the reliability addition Φ_{ij} in each time slot i and the departure as the reliability budget (i.e., the average reliability addition) $\frac{1}{N}\ln\frac{1}{R^{th}}$. The backlog of the virtual queue q_i is then defined as

$$q_{i+1} = \max\left\{0, q_i+\Phi_{ij}-\frac{1}{N}\ln\frac{1}{R^{th}}\right\}, \quad (12)$$

where q_i can also be considered as the reliability deficit because it reflects the deviation of the reliability addition from the reliability budget in each time slot. A large reliability deficit q_i implies that the previous reliability additions exceed the reliability budgets, so a smaller reliability addition is required in time slot i to ensure that the total reliability additions do not surpass the reliability capacity. Therefore, q_i measures the importance of minimizing the reliability addition in time slot i. By combining the objective of minimizing energy consumption, we can reformulate **P0** in time slot i as:

$$\textbf{P1:}\ \min_{x_{ij}}\ V\sum_{j\in\mathcal{M}}E_{ij}x_{ij}+q_i\sum_{j\in\mathcal{M}}\Phi_{ij}x_{ij} \quad (13)$$

$$s.t.,\ (6),\ (8),\ (9),$$

where V is a positive constant to balance the energy consumption objective and the reliability objective. The objective of problem **P1** is to jointly minimize the energy consumption and the weighted reliability addition while satisfying the QoS constraint.

Algorithm 1: Proposed Algorithm

Input : E, Φ, D, R^{th}, $\mathcal{N}$,$\mathcal{M}$,
Output: Task offloading $x_{i,j}$
Initialize $q_i = 0, x_{ij} = 0$;
Initialize current reliability summation $s = 0$;
Set reliability capacity as $\ln\frac{1}{R^{th}}$;
for $i = 1\cdots\mathcal{N}$ **do**
  Sort UAVs in the ascending order of $VE_{ij}+q_i\Phi_{ij}$;
  for $j = 1\cdots\mathcal{M}$ **do**
    if $t_{ij} \leq D_i$ *and* $s+\Phi_{ij} \leq \ln\frac{1}{R^{th}}$ **then**
      $x_{ij} = 1$;
      $s = s+\Phi_{ij}$;
      break;
  end
  Update q_{i+1} based on Eq. (12);
end

P0 can be addressed by solving **P1** in an online fashion for each time slot, which removes the coupling caused by the reliability constraint (11). In time slot i, we choose the optimal UAV for mobile user i. We first sort all UAVs in the ascending order of the objective function $VE_{ij}+q_i\Phi_{ij}$. We then choose the UAV with the smallest objective value and check whether it satisfies both the QoS constraint (6) and the reliability constraint (11). If it satisfies both constraints, we assign the UAV to mobile user i. If not, we proceed to the next UAV with the second smallest objective value and repeat the process until we find a UAV that satisfies both constraints.

Our proposed algorithm is outlined in Alg. 1. Line 1 initializes the reliability deficit $q_i = 0$ and the task offloading decision $x_{ij} = 0$. Lines 2-3 initialize the current reliability summation as 0 and the reliability capacity as $\ln \frac{1}{R^{th}}$. Lines 4-13 choose the allocated UAV for each mobile user. Line 5 sorts the UAVs according to the objective function $VE_{ij} + q_i\Phi_{ij}$. Lines 6-10 choose a UAV that satisfies constraints (6) and (11). Line 12 updates the reliability deficit q_i.

VI. Performance Evaluation

In our simulation, the network is assisted by $N = 25$ UAVs that are also uniformly distributed in the air at a height of $H = 100$ *m*. To model the probability of line-of-sight (LoS) and non-line-of-sight (NLoS) signals, we set the environment-related constants $\alpha = 9.6$ and $\beta = 0.28$, respectively. In the UAV pathloss model [15], we assume the carrier frequency is 2 *GHz*, the speed of light $c = 3 \times 10^8$ *m/s*, and environment-related constants of $\xi_{LoS} = 1$ *dB* and $\xi_{NLoS} = 20$ *dB*. The bandwidth allocated to each user-UAV pair is 1 *MHz*, and the noise power density is $N_0 = -174$ *dBm/Hz*. The task completion deadline is set to 114 *s*. The failure rate of a UAV is 5×10^{-6} and that of a communication link is 5×10^{-6}. The reliability threshold value R^{th} is set to 94%. Each user has a wireless transmission power of 1 *W*. Note that the default values of these parameters can be adjusted to reflect the actual conditions and their impact on the algorithm's performance when we show the simulation results. We adopt three benchmark algorithms for comparison including *ILP*, *Greedy*, and *Random* and we denote our proposed algorithm as *Propose*. *ILP* is the optimum solution obtained by the CPLEX solver. *Greedy* greedily chooses the drone that has the minimum ratio of energy consumption over reliability for each user. *Random* randomly chooses the drone that satisfies the time and reliability constraints for each user.

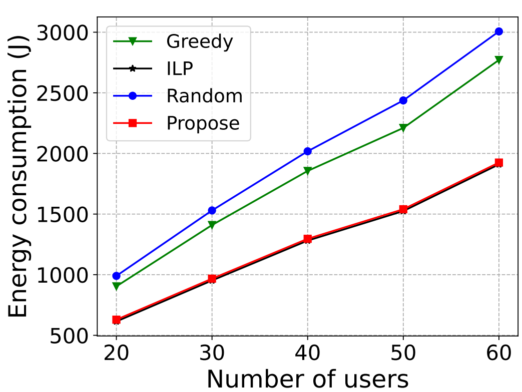

Fig. 2. Energy consumption vs number of users.

Fig. 2 compares four different algorithms in a small-scale problem with user numbers ranging from 20 to 60. The system energy consumption increases with the increasing number of users because more users consume more energy. *Propose* performs close to the optimal solution *ILP*. Besides, *Propose* is better than *Greedy Random* because *Greedy* greedily optimizes the energy consumption for each user without considering the impacts of other users' decisions on the performance.

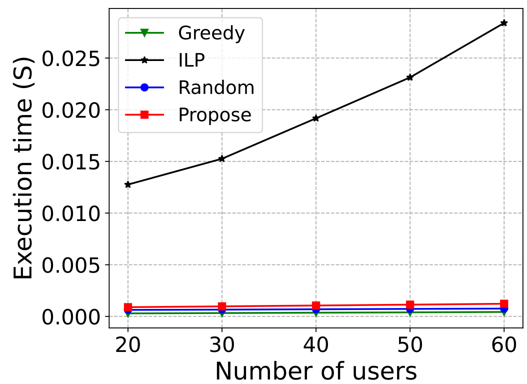

Fig. 3. Execution time vs number of users.

Fig. 3 presents the execution time of four different algorithms with different numbers of users. The execution time increases as the number of users increases because more users lead to larger size problems and so more time is required to get the task offloading decisions. Note that *ILP* incurs an exponentially increasing execution time while *Propose*, *Greedy*, and *Random* are relatively stable. This is because *ILP* usually adopts a branch-and-bound algorithm, thus leading to the exponential time complexity. On the other hand, *Propose*, *Greedy*, and *Random* choose the appropriate UAVs for each user, and thus are executed in polynomial time.

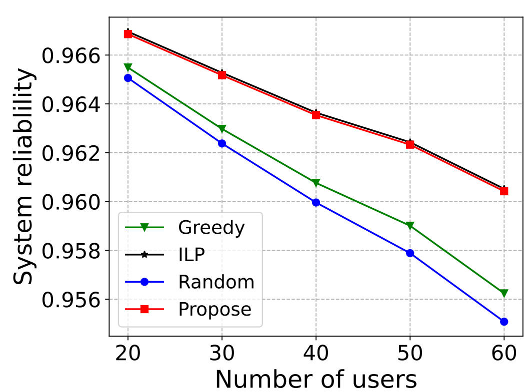

Fig. 4. System reliability vs number of users.

Fig. 4 illustrates the performance of system reliability with different numbers of users ranging from 20 to 60. The system reliability decreases as the number of users increases. This is because more users create more communication links and thus it is more likely to have link failures. Besides, *Propose* performs similar to *ILP* and better than *Greedy* and *Random*.

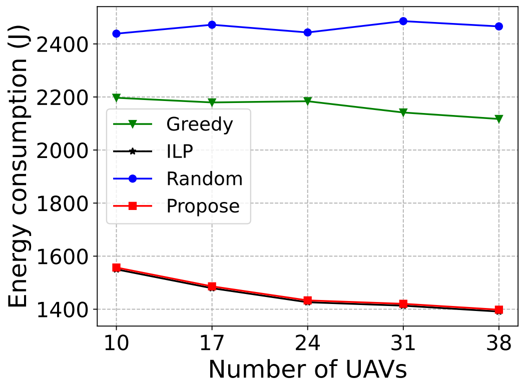

Fig. 5. Energy consumption vs number of UAVs.

Fig. 5 displays the performance of system energy consumption with different numbers of UAVs ranging from 10 to 38. The system energy consumption of *Propose*, *ILP*, and *Greedy* decreases when the number of UAVs increases because more UAVs provide a larger exploration space and so it is more likely to find better UAVs that provides smaller energy consumption in the task offloading problem. Similar to Fig. 2, *Propose* performs similar to *ILP* and better than *Greedy* and *Random*.

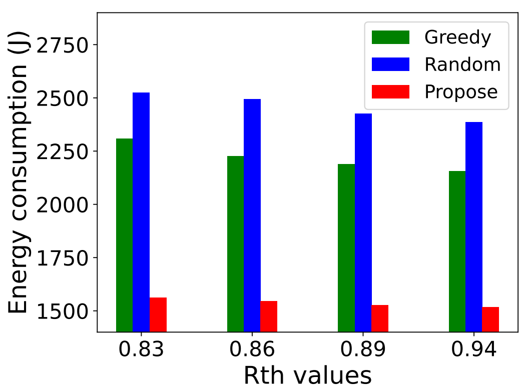

Fig. 6. Energy consumption vs Reliability threshold.

Fig. 6 presents the system energy consumption of *Propose*, *Greedy*, and *Random* with different reliability thresholds R^{th}. It can be observed that the system energy consumption decreases as R^{th} increases. This is because a larger system reliability R^{th} requires a smaller wireless transmission time according to Eq. (4) and hence the energy consumption becomes smaller. Besides, *Propose* consumes less energy than *Greedy* and *Random* with a similar reason in Fig. 2.

VII. Conclusion

In this paper, we have investigated the energy-efficient task offloading problem in UAV-assisted edge networks. We have formulated our problem as an ILP problem with the objective to minimize the system energy consumption while satisfying the task completion time requirement and reliability constraint. We have designed a Lyapunov optimization algorithm to address our problem, which achieves a polynomial time complexity. Extensive simulations have been conducted to demonstrate that our proposed algorithm performs similarly to the optimal solution and better than benchmark algorithms.

References

[1] Y. Mao *et al.*, "A survey on mobile edge computing: The communication perspective," *IEEE Commun. Surv. Tutorials*, vol. 19, no. 4, pp. 2322–2358, 2017.

[2] M. S. Elbamby *et al.*, "Toward low-latency and ultra-reliable virtual reality," *IEEE Netw.*, vol. 32, no. 2, pp. 78–84, 2018.

[3] J. Yao and N. Ansari, "QoS-aware fog resource provisioning and mobile device power control in IoT networks," *IEEE Trans. Netw. Service Manag.*, vol. 16, no. 1, pp. 167–175, Mar. 2019.

[4] ——, "QoS-aware power control in internet of drones for data collection service," *IEEE Trans. Veh. Technol.*, vol. 68, no. 7, pp. 6649–6656, 2019.

[5] J. Wang *et al.*, "Delay-sensitive multi-period computation offloading with reliability guarantees in fog networks," *IEEE Trans. Mob. Comput.*, vol. 19, no. 9, pp. 2062–2075, 2020.

[6] L. Xing, "Reliability in internet of things: Current status and future perspectives," *IEEE Internet Things J.*, vol. 7, no. 8, pp. 6704–6721, 2020.

[7] X. Sun and N. Ansari, "EdgeIoT: Mobile edge computing for the internet of things," *IEEE Commun. Mag.*, vol. 54, no. 12, pp. 22–29, 2016.

[8] Y. Li *et al.*, "Learning-aided computation offloading for trusted collaborative mobile edge computing," *IEEE Trans. Mob. Comput.*, vol. 19, no. 12, pp. 2833–2849, 2020.

[9] Z. Wu *et al.*, "Joint deployment and trajectory optimization in UAV-assisted vehicular edge computing networks," *J. Commun. Networks*, vol. 24, no. 1, pp. 47–58, 2022.

[10] L. Yang *et al.*, "Multi-UAV-enabled load-balance mobile-edge computing for IoT networks," *IEEE Internet Things J.*, vol. 7, no. 8, pp. 6898–6908, 2020.

[11] J. Liu and Q. Zhang, "Reliability and latency aware code-partitioning offloading in mobile edge computing," in *2019 IEEE Wireless Communications and Networking Conference (WCNC)*, 2019, pp. 1–7.

[12] J. Yao and N. Ansari, "Fog resource provisioning in reliability-aware IoT networks," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 8262–8269, 2019.

[13] M. Huang *et al.*, "Reliability-aware virtualized network function services provisioning in mobile edge computing," *IEEE Trans. Mob. Comput.*, vol. 19, no. 11, pp. 2699–2713, 2020.

[14] X. Hou *et al.*, "Reliable computation offloading for edge-computing-enabled software-defined IoV," *IEEE Internet Things J.*, vol. 7, no. 8, pp. 7097–7111, 2020.

[15] M. Mozaffari *et al.*, "Drone small cells in the clouds: Design, deployment and performance analysis," in *Proc. IEEE GLOBECOM 2015*, San Diego, CA, USA, Dec. 2015, pp. 1–6.

[16] Y. Mao, J. Zhang, and K. B. Letaief, "Dynamic computation offloading for mobile-edge computing with energy harvesting devices," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 12, pp. 3590–3605, 2016.

[17] Y. Zeng and R. Zhang, "Energy-efficient UAV communication with trajectory optimization," *IEEE Trans. Wireless Commun.*, vol. 16, no. 6, pp. 3747–3760, Jun. 2017.

[18] S. Shatz and J.-P. Wang, "Models and algorithms for reliability-oriented task-allocation in redundant distributed-computer systems," *IEEE Trans. Reliab.*, vol. 38, no. 1, pp. 16–27, 1989.

[19] L. Xing, G. Levitin, and C. Wang, *Dynamic system reliability: modeling and analysis of dynamic and dependent behaviors*. John Wiley & Sons, 2019.

[20] D. G. Cattrysse and L. N. Van Wassenhove, "A survey of algorithms for the generalized assignment problem," *Eur. J. Oper. Res.*, vol. 60, no. 3, pp. 260–272, 1992.

[21] M. J. Neely, E. Modiano, and C.-P. Li, "Fairness and optimal stochastic control for heterogeneous networks," *IEEE/ACM Trans. Networking*, vol. 16, no. 2, pp. 396–409, 2008.