IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 8 May 2024, accepted 3 June 2024, date of publication 7 June 2024, date of current version 17 June 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3411022

RESEARCH ARTICLE

Toward Optimal Resource Allocation: A Multi-Agent DRL Based Task Offloading Approach in Multi-UAV-Assisted MEC Networks

MUHAMMAD NAQQASH TARIQ[1], JINGYU WANG[1], (Senior Member, IEEE), SALMAN RAZA[2], MOHAMMAD SIRAJ[3], (Senior Member, IEEE), MAJID ALTAMIMI[3], (Member, IEEE), AND SAIFULLAH MEMON[4]

[1]State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China
[2]Department of Computer Science, National Textile University, Faisalabad 37610, Pakistan
[3]Department of Electrical Engineering, College of Engineering, King Saud University, Riyadh 11543, Saudi Arabia
[4]Department of Information Technology, Quaid-e-Awam University of Engineering Science and Technology, Nawabshah, Sindh 67450, Pakistan

Corresponding author: Salman Raza (salmanraza@ntu.edu.pk)

This work was supported by the Deputyship for Research and Innovation, 433 Ministry of Education, Saudi Arabia, under Grant IFKSUOR3-462.

ABSTRACT The application of UAV-aided MEC well-suited for the execution of the data-intensive and latency-sensitive tasks in the infrastructure-deprived regions. However, the growing number of UAVs and smart devices causing a major difficulty in the devising an effective scheme for the task offloading and resource allocation in multi-UAV-aided MEC networks. Furthermore, the resource deficient environments unable to sustain prolonged resource-intensive activities, additional complexities are posed on the optimum utilization of the resources. In this paper, we introduced a multi-agent deep reinforcement learning scheme for the task offloading in the multi-UAV-assisted networks (MUAVDRL). In this configuration, the mobile users fetch computational resources from the UAVs with the goal of minimizing the computation cost which incorporates both the energy consumption and the computation delay. Initially, we start with the optimization problem which is defined as the minimizing the computational costs. Through modelling it as MDP, we aim to reduce the computational costs for mobile users. Leveraging the dynamic and high-dimensional nature of the challenge, the MUAVDRL algorithm solves this problem efficiently. Comprehensive simulation results exhibit the efficacy and superiority of our projected framework when compared to existing state-of-the-art methods, illustrating its potential in the practice.

INDEX TERMS DRL, MEC, resource allocation, task offloading, UAV.

I. INTRODUCTION

The swift progression of the Internet of Things (IoTs) and information and communication technology has propelled smart applications like virtual reality (VR), augmented reality(AR), and autonomous driving [1] into the spotlight. As a result, the demand for minimal latency and considerable computational power has risen with the emergence of these applications, thereby posing substantial challenges for IoT mobile devices. These problems can be solved by Mobile edge computing (MEC) which provides the resources like the cloud at the network edge.. However, traditional MEC servers are costly to deploy and may not be viable in areas lacking infrastructure. To reinforce MEC's adaptability, UAV-assisted MEC has emerged as a promising paradigm [2]. However, prevalent solutions for optimization problems lean on centralized frameworks, reliant on the computational capacity of a central controller. With rising IoT and UAV

The associate editor coordinating the review of this manuscript and approving it for publication was Jingxian Wu.

numbers, complexity, and computational costs surging, they are urging the need for distributed decision-making methods to alleviate this computational burden in UAV-assisted MEC networks.

While mobile devices within UAV-assisted networks increasingly run real-time applications, their operational scope remains significantly restricted due to limited computational resources [3]. Introducing task-offloading strategies has expanded their capabilities by enabling application execution task delegation to nearby computational nodes [4]. However, this enhancement faces challenges, notably the potential hindrance caused by insufficient mobile battery energy [5]. This limitation often leads to service disruptions, as mobile applications may prematurely terminate [6]. The uninterrupted function of contemporary wireless networks necessitates prolonged battery life for sustained energy availability, alongside effective energy utilization by edge devices [7]. These issues collectively impede the network's ability to process real-time applications within designated time slots.

Moreover, [8] focuses on optimizing caching in a cellular network where mobile devices are powered by harvested energy. It aims to minimize the proportion of file segments retrieved from the base station (BS) by optimizing file placement on devices. In another study, Lu et al. [9] investigate the placement of unmanned aerial vehicle base stations (UAV-BSs) in mobile UAV networks to minimize UAV-recall-frequency (UAV-RF) and enhance energy efficiency. Considering various power consumptions and ground user density, they employ a pattern formation system to capture the unstable nature of user density. Additionally, [10] delves into trajectory planning for UAVs in UAV-aided IoT networks with mobile sensor nodes (SNs), focusing on priority-oriented time-sensitive data collection. By assigning different delay sensitivities to each SN, the aim is to minimize UAV energy consumption and SN average delay through trajectory optimization. A proposed heuristic trajectory planning algorithm leverages RL but may introduce oscillations in convergence models, potentially impacting stability and effectiveness in specific scenarios.

Deep Reinforcement Learning (DRL) has gained significant traction, especially within network resource allocation problem domains. Its success in video games [11], continuous action control [12], and now in network resource allocations is noteworthy. Researchers are increasingly exploring DRL's potential in these domains, recognizing that optimal resource allocation is essentially an optimal decision-making challenge. When network dynamics exhibit certain regularities, DRL agents can discern these patterns and learn effective policies accordingly [13]. Chen et al. [14] delved into designing computation offloading policies in Edge Computing (EC) through deep Q-networks, integrating task queue state, energy queue state, and channel state information. Meanwhile, Ye et al. [15] focused on resource allocation a DRL-based scheme within V2V communications, treating V2V links as agents making informed decisions regarding optimal power levels and sub-band for transmissions. Sun et al. [16] examined the computation migration as a crucial issue in vehicular cloud scenario, employing polynomial estimation for value functions and SARSA algorithm training to address missions with linear inter-dependency topology. Conversely, Zhu et al. [17] explored research on UAV-supported edge computing for missions with different inter-dependence topology structures using only single agent Actor-Critic (AC) algorithm for selecting target UAVs, albeit without accounting for parallel transmission and energy constraints. Additionally, Min et al. [18] came up with another offloading algorithm based on RL for IoT devices with energy harvesting, considering factors like current battery levels, previous radio transmission rates, and predicted harvested energy. However, the authors overlooked the time delay factor concerning energy consumption, potentially limiting the comprehensive nature of their approach.

There has been substantial research into UAVs aiding mobile edge computing, covering aspects like energy efficiency [19], trajectory design [20], and the joint optimization of communication and computation [21]. However, few of these studies account for scenarios involving multiple UAVs. A single UAV, constrained by a limited payload, possesses restricted computing resources. Moreover, several recent studies have explored multi-UAV-assisted MEC. specifically, Huang et al. [22] delve into offloading scenarios within multi-UAV-enabled aerial edge computing. The authors addressed a scenario where users have interdependent tasks, and multiple UAVs process these tasks centrally. Using a multi-objective optimization approach, they aim to optimize completion time and energy balance among UAVs. However, this study overlooked computing resource allocation, and practical constraints, like limited resources, could indirectly affect its optimization objectives. Consequently, UAV-assisted mobile edge computing necessitates a collective effort among UAVs to share the computation workload. Diverse computing capabilities among UAVs result in varying processing and communication times. Additionally, differences in energy conditions between UAVs and mobile users directly impact offloading outcomes. Within the dynamic network environment, efficiently offloading tasks for mobile users during ground network emergencies poses a significant challenge. The local ground MEC server might become overwhelmed as mobile users demand varied resources within the UAV network. Moreover, the mobile nature of some users causes them to fall outside the network coverage. Considering UAV-assisted networks' complex and dynamic conditions and resources, the task offloading policy should be carefully designed to minimize computation costs.

Due to the limitations of preceding works and the challenges stated above, in this article, we introduced a multi-agent DRL into a UAV-enabled network to propose intelligent task offloading. We offer task offloading and resource allocation, where mobile users offload their tasks to

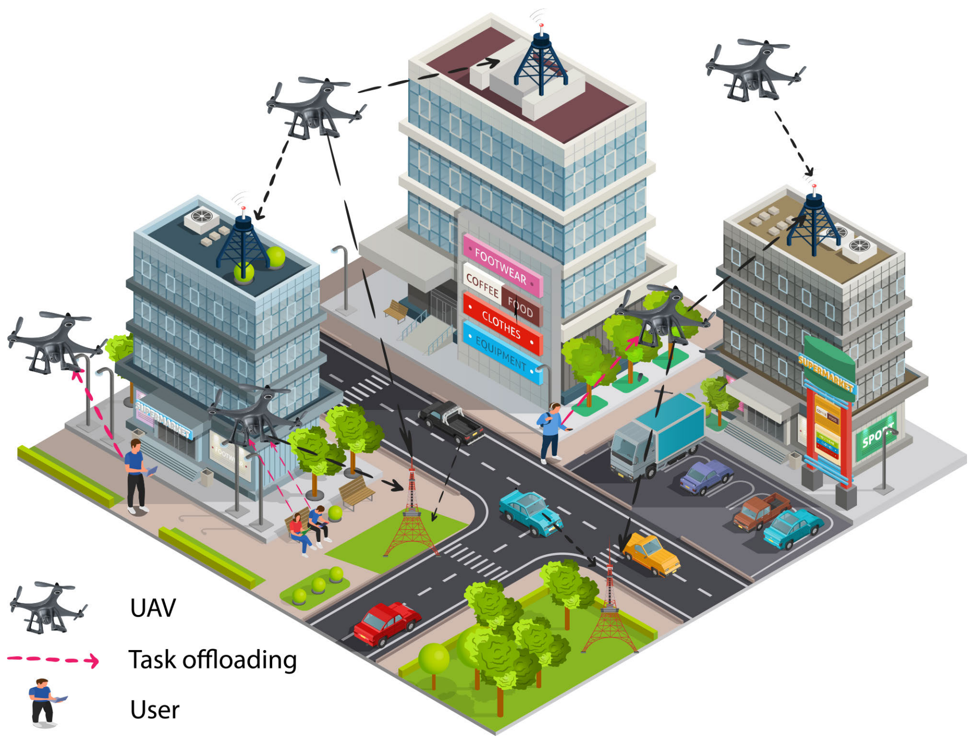


FIGURE 1. System model showcasing multi-UAV-assisted MEC network.

UAVs to perform task offloading. We develop the problem and transform it into a Markov decision process (MDP) model to attain optimum offloading decision policies. Given the presence of multi-objective problems and the dynamic nature inherent in UAV-enabled network environments, we incorporate the capabilities of advanced DRL techniques. This approach allows us to effectively grasp the dynamic network conditions, paving the way to devise an optimal task offloading algorithm that efficiently manages interactions between mobile users and UAVs.

This work's pivotal contributions can be encapsulated as follows:

1) We proposed a MUAVDRL scheme for task offloading among mobile devices within a multi-UAV-assisted network. The primary aim is to minimize computation costs, specifically targeting energy consumption and computation delay. This scheme empowers mobile devices to acquire environmental insights, enabling them to empower optimal task-offloading decisions within the network.
2) We developed a resource allocation and task offloading problem for mobile devices into a Markov MDP model and utilize a multi-agent model-free deep deterministic policy gradient algorithm for multi-UAV-assisted networks (MUAVDDPG). This method manages the expansive and continuous action space to maximize long-term rewards and determine the most efficient task-offloading strategy.
3) We extensively simulate and compare the performance of our proposed algorithm against three benchmark algorithms. These simulations demonstrate that our proposed algorithm significantly minimizes computation costs within a dynamic network setting.

The rest of this paper is arranged as follows. In Section II, the system model is presented. In Section III, the problem is formulated is presented. The MUAVDRL scheme and the algorithm are presented in Section IV. The simulation results are given in Section V. Finally, Section VI presents the conclusion.

II. SYSTEM MODEL

The illustration in Fig. 1 delves into a comprehensive analysis of a Multi-UAV-assisted MEC network, constituting a collective of M UAVs and N mobile users. The UAVs are concisely labeled within this network structure in the set $M = \{1, 2, \ldots, m, \ldots, M\}$. At the same time, the mobile users are categorised explicitly in $N = \{1, 2, \ldots, n, \ldots, N\}$, which provides a structured and scalable representation, allowing for systematic coordination and efficient communication among the various entities present in the network. Owing to limited computational capabilities, mobile devices must transfer their computation tasks to UAVs for processing. Employing a comprehensive offloading model is essential to minimize the time and energy consumption of the user's equipment. Each UAV has a maximum capacity to connect with up to N_{max} mobile users concurrently, while each user is limited to offloading tasks to only one UAV during a specific time.

Each mobile user $n \in N$ contributes a computational task described by $D_n = (S_n, C_n, \tau_n)$, here S_n, C_n, and τ_n represent the task size, necessry for CPU-cycle frequencies, and for the imposed delay constraint, respectively. The parameter $\alpha_n \in \{1, 2, \ldots, M\}$ serves as the association policy for user n, indicating the selection of a UAV by user n for offloading purposes. This association policy is crucial in optimizing resource allocation, enabling each user to select a UAV strategically based on factors such as proximity, resource availability, and computation efficiency, ensuring efficient task offloading and timely processing within the multi-UAV environment. The notation used in the system model are illustrated in Table 1.

TABLE 1. List of notations.

Notations	Description
M	Set of UAVs
N	Set of mobile users
D_n	Set of computation task demand
S_n	Task Size
C_n	Required CPU cycles
τ_n	Delay Constraint
W	Bandwidth
d_{mn}	Distance between UAV m and user n
G_{mn}	Channel gain from user n to UAV m
r_{mn}	Transmission rate between mobile user n and UAV m
$T^{off}_{n,m}$	Transmission delay of user n offloading tasks to UAV m
$E^{off}_{n,m}$	Energy consumption of user n for offloading to UAV m
$T^{re}_{n^{(i)},m}$	Queuing Delay
$T^{tot}_{n,m}$	Total Delay
C^{tot}	Total system cost

A. CHANNEL MODEL

The overall system bandwidth, denoted as W, is distributed among M unmanned aerial vehicles (UAVs), each utilizing a bandwidth of W_m. To prevent interference among these UAVs, the combined bandwidth across all UAVs must not surpass the overall system bandwidth, i.e., $\sum_{m=1}^{M} W_m \leq W$. Moreover, an equal distribution of bandwidth allocation is assumed among users connected to a particular UAV. This means that users connected with UAV m evenly divide the overall bandwidth W_m allocated to that specific UAV m. To ensure optimal utilization of bandwidth, the collective bandwidth used by all UAVs should not fall below the total system bandwidth W [23], which can be represented as follows:

$$\sum_{m \in \mathcal{M}} W_m = W, 0 < W_m < W. \tag{1}$$

The coordinates defining the position of UAV $m \in M$ and user $n \in N$ are specified as $L_m = (x_m, y_m, h)$ and $L_n = (x_n, y_n, 0)$, respectively. To determine the Euclidean distance in a three-dimensional space between two points UAV m and user n, an approach detailed in [2] is adopted, which can be expressed as:

$$d_{mn} = \sqrt{(x_m - x_n)^2 + (y_m - y_n)^2 + h^2} \tag{2}$$

Under the assumption, communication link among UAVs and mobile users is primarily Line of Sight (LoS), as highlighted in [24], the channel gain between mobile user n and UAV m is defined as per [25]:

$$G_{mn} = (d_{mn})^{-\lambda}, \tag{3}$$

where, the symbol λ illustrates the path loss exponent, serving as a parameter in the characterization of path loss for the communication link between the mobile user and the UAV.

B. OFFLOADING MODEL

The expression for the transmission rate between UAV m and mobile user n for the offloaded task D_n to UAV m is represented as:

$$r_{mn} = \frac{W_m}{|\mathcal{N}_m|} \log_2 \left(1 + \frac{p_n G_{mn}}{N_0}\right), \tag{4}$$

where, W_m stands for the total bandwidth allocated to UAV m. $\mathcal{N}_m$ signifies the count of users associated to UAV m, with N_m representing this set of users as $N_m = n1, n2, \ldots, n|N_m|$. ensures an equitable distribution of available bandwidth resources among the users connected to UAV m. Additionally, the variable p_n refers to the transmission power of user n, while N_0 encompasses the background noise within the system.

The time taken for user n to transfer tasks to UAV m, referred to as the transmission delay, can be represented as:

$$T^{off}_{n,m} = \frac{S_n}{r_{mn}}. \tag{5}$$

The energy consumption related to user n during the process of offloading tasks to its respective UAV can be formulated as:

$$E^{off}_{n,m} = p_n T^{off}_n. \tag{6}$$

C. QUEUING MODEL

Once all computation tasks are received from their corresponding users, each UAV m organizes these tasks based on their order of arrival within the set of user $\mathcal{N}_m = \{n^1, n^2, \ldots, n^{|\mathcal{N}_m|}\}$. Consequently, the computation delay for user n at MEC m includes both queuing and computation delay. The queuing delay, crucially dependent on the order of task arrival, aggregates the computational delays of all tasks handled prior to the current task in the sequence. Tasks arriving earlier experience a longer queuing delay as they wait for previously arrived tasks to be processed. For instance, considering user $n^{(i)} \in \mathcal{N}_m$, processing delay pertaining to $n^{(i)}$ is outlined in [26].

$$T^{re}_{n^{(i)},m} = \begin{cases} \dfrac{C_{n^{(i)}}}{f_m}, i = 1 \\ T^{re}_{n^{(i-1)},m} + \dfrac{C_{n^{(i)}}}{f_m}, i > 1, \end{cases} \tag{7}$$

Therefore, accounting for the available computation frequency of UAV m, denoted as f_m, the comprehensive latency incurred in processing tasks from user n encompasses both the queuing delay and the computational time. As defined in (7) the remaining processing time $T^{re}_{n^{(i)},m}$ for tasks from user $n^{(i)}$ at UAV m. It comprises two cases: firstly, when $i = 1$, indicating the first task from user n, the remaining processing time is straightforwardly computed as the computational time of the initial task $C_{n^{(i)}}$ divided by the computation frequency of UAV m as f_m. Subsequently, for tasks beyond the first $i > 1$, the remaining processing time is determined by adding the computational time of the current task $C_{n^{(i)}}$ divided by f_m to the remaining processing time of the previous task $T^{re}_{n^{(i-1)},m}$. This formulation enables tracking of the time required for task completion, considering both the computational demands of individual tasks and the processing capabilities of the UAV. Hence, the total latency is calculated as the sum of the queuing delay and the time required for computation on UAV m, which is defined as follows.

$$T^{tot}_{n,m} = T^{off}_{n,m} + T^{re}_{n,m}, \tag{8}$$

The variable $\beta_n = \{0, 1\}$ functions as a binary indicator, signifying whether user n gratifies the stipulated delay requirement. When $T^{tot}_{n,m} \leq \tau_n \cdot \beta_n = 1$, it indicates that the user meets the specified delay constraints. This condition ensures that the total time taken for task processing by user n at UAV m does not exceed the designated maximum time threshold, thus adhering to the latency requirements.

Consequently, the total amount of the system cost amalgamates the energy consumption and time utilization from all users whose computational tasks are completed successfully. The representation of the total amount of the system cost is formulated as the summation of energy and time costs across all users in the system whose task computations are within the stipulated time constraints:

$$\begin{aligned} C^{tot} &= \delta^e E^{tot} + \delta^t T^{tot} \\ &= \delta^e \sum_{n\in\mathcal{N}} \sum_{m\in\mathcal{M}} \beta_n p_n T^{tot}_{n,m} \\ &\quad + \delta^t \sum_{n\in\mathcal{N}} \sum_{m\in\mathcal{M}} \beta_n T^{tot}_{n,m}. \end{aligned} \tag{9}$$

III. PROBLEM FORMULATION

The primary objective is to minimize the cost of the system in multi-UAV-assisted network scenarios. The system cost is expressed as energy consumption and computation delay. For this, we formulate an optimization problem to minimize the system cost of the system through optimizing task offloading strategy and resource allocation to eventually reduce the overall system cost and is mathematically expressed as:

$$P1 : \min_{T,E} C^{tot}$$

$$s.t. \sum_{m\in\mathcal{M}} W_m = W, \forall m, \tag{10a}$$

$$\alpha_n \in \{1, 2, \ldots, m, \ldots, M\}, \forall n, m, \tag{10b}$$

$$|\mathcal{N}_m| \leq N_{\max}, , \forall m, \tag{10c}$$

$$p_n \in \{1, 2, \ldots, n, \ldots, N\}, \forall n. \tag{10d}$$

$$E^{off}_n \leq E^{max}_n, \quad \forall n. \tag{10e}$$

$$T^{tot}_n \leq T^{max}_n, \quad \forall n \tag{10f}$$

The optimization problem (*P1*) revolves around key variables: the user association vector, transmit power vector, and UAV bandwidth allocation vector. These variables are defined as follows:

1) User Association Vector (α): The $\alpha = [\alpha_1, , \alpha_n, \ldots, \alpha_N]$ represents the user association vector. Each α_n denotes the choice of a UAV by user n for task offloading.
2) Transmit Power Vector (p): The $p = [p_1, \ldots, p_n, \ldots, p_N]$ represents the transmit power vector. The p_n corresponds to the transmission power of individual users.
3) UAV Bandwidth Allocation Vector (W): The $W = [W_1, \ldots, W_m, \ldots, W_M]$ denotes the UAV bandwidth allocation vector. W_m stands for the total bandwidth allocated to each specific UAV m in the system.

These variables set the foundation for the optimization problem, which is further defined by the following constraints:

1) Bandwidth Restriction ($\sum_{m\in\mathcal{M}} W_m = W, \forall m$): Constraint (10a) ensures that the aggregate bandwidth usage across all UAVs does not exceed the total system bandwidth W.
2) User Association ($\alpha_n \in \{1, 2, \ldots, m, \ldots, M\}, \forall n, m$): Constraint (10b) represents the user association indicator governed by the user association vector α. It imposes a limit on the maximum number of users a UAV can associate with, denoted as $N_{\max}$.
3) Maximum User Association ($|\mathcal{N}_m| \leq N_{\max}, , \forall m$): Constraint (10c) regulates the number of users connected to each UAV, aiding in workload management and optimization for individual UAVs within the network.
4) Transmit Power Range ($p_n \in \{1, 2, \ldots, n, \ldots, N\}$, $\forall n$): Constraint (10d) signifies the permissible range for users' transmit power.
5) Energy Limitation ($E^{off}_n \leq E^{max}_n$, $\forall n$): Constraint (10e) outlines mobile users' energy limitation, ensuring that each computing device's battery energy remains adequate.
6) Time Constraint ($T^{tot}_n \leq T^{max}_n$, $\forall n$): Constraint (10f) sets the time constraint for mobile users, stipulating that the total time T^{tot}_n should not surpass the maximum delay constraint for the task, denoted as T^{max}_n.

The proliferation of mobile devices is experiencing rapid exponential growth, coupled with the dynamic network scalability, thereby elevating the complexity of problem P1. This complexity stems from its classification as a mixed-integer and non-convex optimization problem. It revolves around the offloading decision vector α and features a nonconvex objective function, a characteristic that renders it NP-hard and poses challenges in directly deriving an optimal solution. An algorithm in polynomial time would not be able to unveil the best decision.Hence, we are providing a simplified DRL

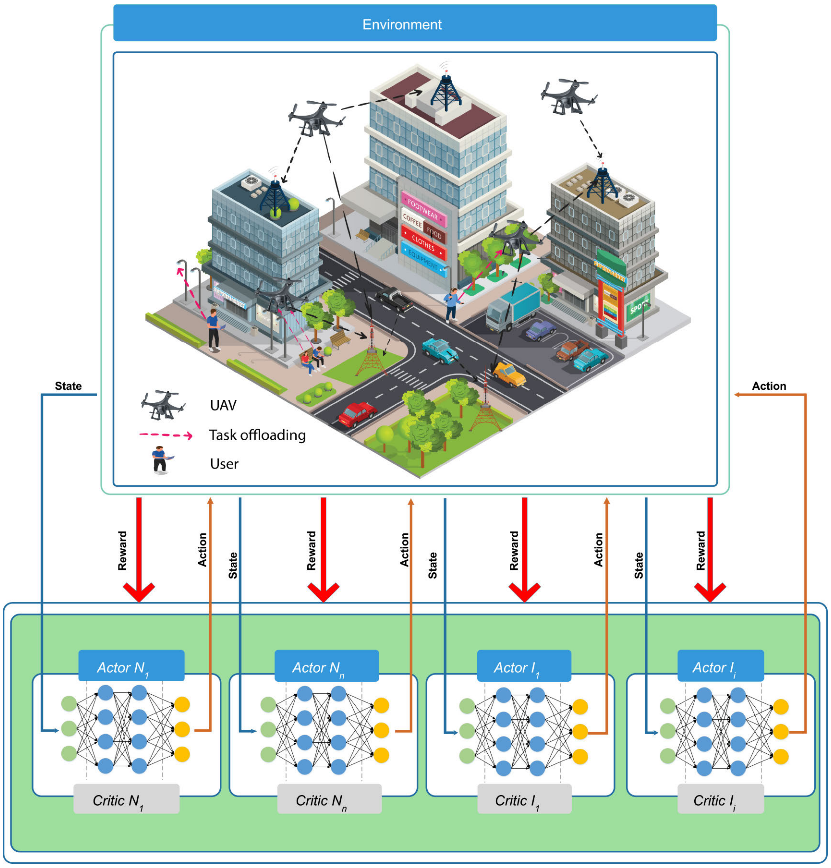


FIGURE 2. MUAVDRL scheme for multi-UAV-assisted MEC network.

algorithm that minimizes the system offloading cost while maximizing resource allocation efficiency.

IV. MULTI-AGENT DRL SCHEME

A. MUAVDRL BASED TASK OFFLOADING

In this paper, we explore the task offloading paradigm within a multi-UAV-supported network operating within a dynamic environment where multiple agents are actively involved. The complexities of optimizing solutions in such a setting using traditional single-agent approaches have been acknowledged as challenging [27]. To overcome these hurdles, our contribution introduces the MUAVDRL algorithm, tailored explicitly to address multiple problems within a multi-agent system. This algorithm is a strategic response to the complexities of optimizing task offloading across a network supported by multiple UAV agents.

We have formulated the problems within a Markov decision process framework in a multi-agent setting. Our approach introduces a MUAVDRL-based task offloading scheme aimed at minimizing the computation costs of mobile users while concurrently maximizing the rewards accrued by UAVs. Within this scheme, mobile devices and UAVs function as agents, engaging in a competitive interplay to accomplish their objectives. As illustrated in Fig. 2, agents compete during each stage, gaining experiential insights from others by observing the environmental state. The rewards acquired by UAVs and mobile users constitute the system's overall reward. To execute the MUAVDRL method effectively, we define the state space, action space, and rewards as follows:

- **State space:** The learning agents within the system acquire experiential knowledge and refine their decision-making policies by observing the environmental states. Their decisions are selected by optimization objectives within the environment, relying on the state space $s(t)$. This state space $s(t) = (r_{mn}, E^{off}_{n,m}, f_m, p_n)$, comprising elements such as the transmission rate r_{mn} between user n and UAV m, user energy consumption $E^{off}_{n,m}$ during task offloading, UAV computation frequency f_m, and user transmit power p_n, serves as the cornerstone for these agents' adaptive decision-making, enabling them to optimize strategies based on the current environmental conditions.
- **Action space:** The vector of offloading decision $B = [\beta_1, \beta_2, \ldots, \beta_N]$, vector of transmit power $p = [p_1, \ldots, p_n, \ldots, p_N]$, and computation resource allocation is represented as $F = [f_1, f_2, \ldots, f_N]$. The complete action vector at time 't' is indicated as $a(t) = [\beta_1, p_1, f_1, \ldots, \beta_N, p_N, f_N]$.
- **Reward function:** The overarching aim of the reward function is to enhance offloading decisions, mitigating computation costs encompassing delays and

energy consumption. This optimization is crucial for ensuring QoS while extending the battery lifespan of mobile users. The reward function for mobile users is formulated as:

$$r_{n,m}(t) = -(\delta^e E^{tot} + \delta^t T^{tot}). \tag{11}$$

The reward function for UAVs is defined as:

$$r_m(t) = R_m(t). \tag{12}$$

The comprehensive reward function for the system is formulated as:

$$r(t) = \sum_{n \in \mathcal{N}} \sum_{m \in \mathcal{M}} \Big(r_{n,m}(t) + r_m(t) \Big). \tag{13}$$

B. MUAVDRL ALGORITHM

Fig. 2 illustrates the MUAVDRL framework within a multi-UAV-supported network. The MUAVDDPG algorithm, an adapted actor-critic network incorporating the DQN method, demonstrates its efficacy in managing dynamic environments. Particularly suited for cooperative policy learning among multiple agents operating within a continuous action space, this algorithm stands as a valuable tool. It is essential to highlight that this system facilitates centralized training of the action-value function, known as the critic network while promoting decentralized execution. The critic function utilizes action policies from other agents, enabling each agent to decide its actions autonomously based on individual strategies and observations. The update of policy parameters is carried out individually by each agent within the framework.

The scheme involves a collection of J agents, encompassing both UAVs denoted as M_m and mobile users denoted as N_n, each equipped with distinctive observations, actions, and reward mechanisms. Within this context, each agent, indexed as J among the J agents, possesses a distinct set of states denoted as $\mathcal{S} = \{S_1, S_2, \ldots, S_J\}$, an array of actions denoted as $\mathcal{A} = \{\mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_J\}$, and a series of observations represents as $\mathcal{O} = \{\mathcal{O}_1, \mathcal{O}_2, \ldots, \mathcal{O}_J\}$. In this scheme, agents are empowered to select an action utilizing a stochastic policy $\pi_{(\theta_j)} : \mathcal{O}_j \times \mathcal{A}_j \to [0, 1]$ to select an action based on its own observations and available actions. The transition to the subsequent state is facilitated by the state transition function $\mathcal{T} : \mathcal{S} \times \mathcal{A}_j \times, \ldots, \times \mathcal{A}_J \to S$. Every agent receives rewards centered on their current states and the actions taken, depicted as $r_j : \mathcal{S} \times \mathcal{A}_j \to \mathcal{R}$. Additionally, each agent privately acquires their specific observation $\mathcal{O}_j : \mathcal{S} \to \mathcal{O}_j$. Each agent j operates with the primary goal of maximizing its expected cumulative rewards, denoted as $\mathbb{E}\left\{\sum_{j=0}^{\infty} \gamma^j r_j, t+k\right\}$. Here, $r_j, t+k$ signifies the reward obtained by agent j at a time k episodes into the future, considering the discount factor γ. Within the MUAVDDPG framework, the fundamental concept revolves around learning a centralized critic function, namely the action-value function, designated as $Q_j^{\pi}(\mathcal{J}, \alpha_1, \ldots, \alpha_J)$ for the j-th agent. This function involves representations of actions from all agents $\{\alpha_1, \ldots, \alpha_J\} \in \mathcal{A}$, and each agent possesses comprehensive information regarding global training weights.

Moreover, the collective policies' set utilized by agents is denoted as $\pi = \pi_1, \ldots, \pi_J$, each accompanied by its respective policy parameter represented as $\theta = \theta_1, \ldots, \theta_J$. This centralized learning approach underpins the MUAVDDPG framework, facilitating coordinated learning among multiple agents within the system. Within the MUAVDDPG algorithm, updates are performed on both the actor and critic-network upon completing every training episode. More precisely, the actor-network undergoes an update using a gradient method computed in the following manner:

$$\begin{aligned} \nabla_{\theta_j}(\theta_j) = \mathbb{E}_{\mathcal{J},a\sim\mathcal{D}}[\nabla_{a_j}\theta_j^{\mu}(\mathcal{J}, a_1, \ldots, a_J) \\ \nabla_{\theta_j}\mu\theta_j(\mathcal{O}_j)|_{aj} = \mu\theta_j(\mathcal{O}_j)], \end{aligned} \tag{14}$$

where, the variable $\mathcal{D}$ signifies the replay buffer, responsible for storing tuples of experiences encountered by all agents, consisting of $(\mathcal{J}, \mathcal{J}', a_1, \ldots, a_J, r_1, \ldots, r_J)$. The update process for the critic function θ_j^{μ} is represented as follows:

$$\mathcal{L}(\theta_j) = \mathbb{E}_{\mathcal{J},\mathcal{J}',a_1,\ldots,a_J,r_1,\ldots,r_J}[(\theta_j^{\mu}(\mathcal{J}, a_1, \ldots, a_J) - y)^2], \tag{15}$$

where, $y = r_j + \gamma\theta j^{\mu'}(\mathcal{J}', a1', \ldots, a_J')|ak' = \mu k'(\mathcal{O}k)$ denotes the computation for updating the critic function. Here, the target policies' set, characterized by delayed parameters, is represented as θ_j', represented as θ_j', is depicted as $\mu' = \mu\theta_1', \ldots, \mu\theta_J'$. This calculation involves incorporating the rewards received by agent j alongside evaluating the next state-action values utilizing the target policies, considering the delayed parameters.

Algorithm 1 outlines the proposed MUAVDDPG algorithm tailored for task offloading within a multi-UAV-supported network. It comprises two essential procedures: Collecting the observed data and the training process. The algorithm commences with an initialization phase (lines 1-4), establishing the replay buffer and initializing parameters for the actor-network and critic-network along with their respective weights. This phase also sets the groundwork by defining the total number of episodes and training steps for the agents (lines 5-10). This preliminary step ensures that the algorithm starts with a defined structure and clear parameters before the actual training begins. Collection of observed data by agents: Execution of actions, reward acquisition, and generation of new states (lines 11-21). The experiences gained are stored within the experience replay buffer. Training phase (lines 22-28): Policy training involves batch sampling from the replay buffer. Subsequently, the actor-network and critic-network undergo updates based on randomly selected samples. This algorithm orchestrates the collection of data from the environment by agents, storing experiences for later training. During the training phase, samples from these experiences to update the actor and critic networks, facilitating policy learning within the multi-agent system.

Algorithm 1 MUAVDDPG Algorithm for Task Offloading

```
**Initialization:**
Replay buffer $\mathcal{D}_j^{loc}$ at UAV and User.
**Initialization:**
Actor and critic networks parameters are initialized with $\theta$
**for** episode = 1 to 3000 **do**
    Initialize the states $S = \{s_1, s_2, \ldots, s_J\}$
    **for** $t$ = 1 to 200 **do**
        Estimate the resource requirements of the user $(E_{n,m}^{off}, f_n)$
        Estimate the available resource of UAV $(f_m)$
        Each agent receives an initial state. $s(t) = (r_{mn}, E_{n,m}^{off}, f_m, p_n)$
        Every agent $j$ makes a stochastic decision $a_j$ according to the policy $\pi_{\theta_j}$
        given the state $s_j$ with probability $\varepsilon$
        Agents perform the action $a(t) = \{a_1(t), a_2(t), \ldots, a_J(t)\}$, encompassing $a = [\beta_1, p_1, f_1, \ldots, \beta_N, p_N, f_N]$
        The rewards observed are denoted as $r(t) = \{r_1(t), r_2(t), \ldots, r_J(t)\}$
        The new state $s_j(t+1)$ is denoted by $s'_j$
        store the tuples $\{s_j(t), a_j(t), r_j(t), s'_j\}$ in $\mathcal{D}_j^{loc}$
        $s_J \leftarrow s'_j$
        **for** agent $j$ = 1 to $J$ **do**
            Mini-batch of $H$ samples tuples $(s^k, a^k, r^k, s'^k)$ from $D_j^{loc}$
            Set $y^k = r_j^k + \gamma Q_j^{\pi'}\left(S'^k, a'_1, \ldots, a'_J\right)|_{a'_j=\pi'_j(s_j^k)}$
            Update the critic-network as (15)
            Update actor-network as (14)
        **end for**
        Update the target network's parameters for each agent $j$
        $\theta'_j \leftarrow \tau\theta_j + (1-\tau)\theta'_j$
    **end for**
**end for**
```

C. COMPUTATIONAL COMPLEXITY

The proposed MUAVDDPG algorithm outlines the computational complexities for actor and critic networks. For the actor-network with P layers and the critic network with U layers, the computational complexity of each layer is expressed as $\mathcal{O}\left(Z_{p-1}^a Z_p^a + Z_p^a Z_{p+1}^a\right)$ and $\mathcal{O}\left(Z_{u-1}^c Z_u^c + Z_u^c Z_{u+1}^c\right)$ respectively. Consequently, the overall training complexity of the proposed algorithm can be expressed as:

$$\mathcal{O}\left(J * T * E * \left(\sum_{p=2}^{P-1}\left(Z_{g-1}^a Z_p^a + Z_p^a Z_{p+1}^a\right) + \sum_{u=2}^{U-1}\left(Z_{u-1}^c Z_u^c + Z_u^c Z_{u+1}^c\right)\right)\right) \tag{16}$$

Each agent's execution complexity is $\mathcal{O}\left(J * E * \left(\sum_{p=2}^{P-1}\left(M_{p-1}^a Z_p^a + Z_p^a Z_{p+1}^a\right)\right)\right)$. The algorithm's convergence analysis considers UAV agents' interaction, influencing subsequent user agent actions while supporting policy changes among agents.

V. SIMULATIONS RESULTS AND DISCUSSIONS

In this section, we analyze the outcomes of our algorithm's simulations across various parameter configurations. Our simulations were executed in a Python 3.7 environment utilizing Tensorflow 2.0. The computations were performed on a system powered by a Core i7 CPU operating at 2.4GHz and equipped with 16GB RAM. We consider our analysis to involve a range of 2 to 12 UAVs and 10 to 60 mobile users unless specified otherwise. These UAVs are positioned at an altitude of 100m above ground level. Within our devised multi-agent algorithm, we established 3000 episodes during the training phase, each comprising 200 steps. The neural network employed in this algorithm is a fully connected architecture featuring both a critic network and an actor-network. Specifically, every agent within the system operates with a neural network structure comprising two hidden layers in the actor as well as critic neural networks. These layers consist of 256 neurons in the initial hidden layer and 128 neurons in the subsequent layer. A mini-batch size of 256 was employed to manage the learning process. Other simulation parameters are summarized in Table 2.

TABLE 2. Experimental parameters.

Parameters	Value
Task Size	[150-1500] KB
CPU cycles required for task	[150-2000] Mcycles
Maximum processing time delay	[0.1-2] Sec
Mobile Transmission Power of	0.1 W
Task Size	[150-1500] KB
Bandwidth	40 MHz
Learning rate of actor-network	$1e^{-4}$
Learning rate of critic-network	$3e^{-4}$

The proposed MUAVDDPG algorithm is evaluated against the following baseline algorithms.

1) Deep Deterministic Policy Gradient (DDPG) [28]: A DDPG-based scheme wherein each agent learns from prior offloading experiences and dynamically fellows the nearest computational UAV node.
2) A3C [28]: Both the A3C and DDPG algorithms belong to the paradigm of model-free DRL and follow an actor-critic framework. However, their distinct advantages come to light in various applications, especially in achieving efficient convergence during training.
3) DQN [29]: DQN exhibits strengths in dynamically allocating computational tasks among heterogeneous resources. DQN learns from experience and navigates high-dimensional state spaces contributing to effective decision-making in dynamically changing network conditions.

The convergence analysis depicted in Fig. 3 illustrates the performance of various DRL algorithms. Specifically, the proposed MUAVDDPG algorithm demonstrates superior performance compared to the three baseline algorithms. Convergence for the MUAVDDPG algorithm occurs before the 550-episode mark, while the other three algorithms converge around the 700-episode range. This points to the successful alignment of interests between UAVs and mobile users, effectively maximizing rewards and minimizing computation costs through optimal offloading decisions. An observed trend reveals an increase in average rewards proportional to both the number of episodes and the number of agents. Remarkably, the proposed MUAVDDPG algorithm yields an overall reward increase of 16%, 26%, and 33% compared to DDPG, A3C, and DQN, respectively. This notable enhancement in reward values signifies the efficacy of the MUAVDDPG approach in optimizing both UAV and mobile users' objectives within a dynamic network environment.

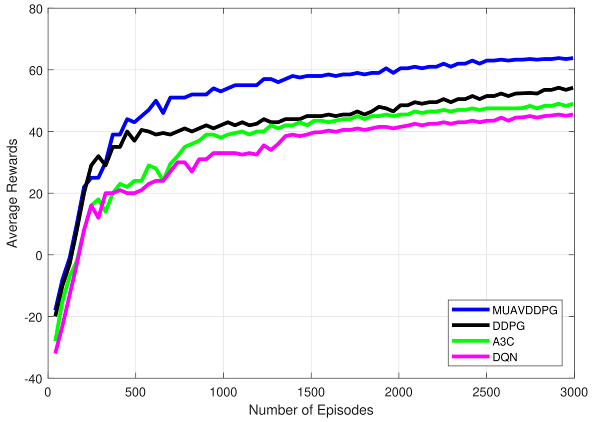
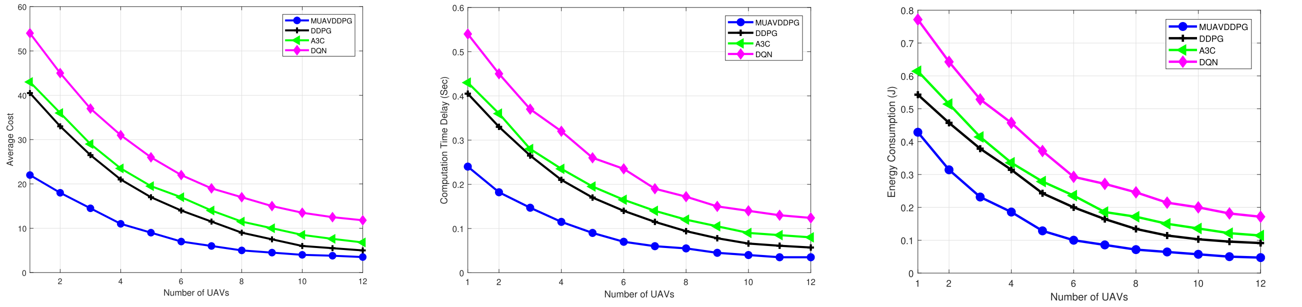


FIGURE 3. Average rewards with training episodes.

In Fig. 4, our focus initially centers on analyzing the performance of the proposed algorithm concerning the escalating number of UAVs. As observed in Fig. 4a, a trend emerges, indicating a decrease in the average computation cost across all algorithms with an increase in the number of UAVs. Notably, the MUAVDDPG algorithm showcases a swifter reduction in average cost, surpassing the performance of the three baseline algorithms. This pattern suggests that under the MUAVDDPG framework, agents collaborate more effectively, leading to reduced resource consumption compared to other algorithms. DDPG and A3C exhibit superior performance to DQN due to their enhanced support for distributed learning. Particularly noteworthy is the MUAVDDPG's remarkable reduction in average cost by 30%, 49%, and 70% compared to DDPG, A3C, and DQN, respectively. This demonstrates the MUAVDDPG algorithm's effectiveness within a multi-agent system, surpassing baselines by optimizing agents' decisions. Additionally, Fig. 4b and 4c portray a notable trend: as the number of UAVs increases within the MUAVDDPG algorithm, optimal time delay and energy consumption decrease. This trend is also observed, to a lesser extent, in DDPG, A3C, and DQN. Thus, across various numbers of UAVs, the MUAVDDPG algorithm consistently outperforms the three baseline algorithms by efficiently minimizing time delay and energy consumption.

In Fig. 5, our investigation delves into the average cost, time delay, and energy consumption across various data sizes from 1 to 16 MB. This analysis aims to assess how offloading data sizes impact system performance. Fig. 5a graphs the average cost across diverse data sizes. The observed gradual increase in the average cost across all algorithms is attributed to larger offloading data sizes demanding more time and energy consumption. Tasks with increased data sizes require higher computational time and resources from UAVs. Once the offloading task size reaches 4 MB, there's a rapid spike in the average cost for the three benchmark algorithms. However, under the proposed algorithm, the average cost escalates significantly slower. The algorithm's collaborative nature facilitates information sharing among agents, enabling efficient cost minimization. Comparatively, the proposed algorithm demonstrates a significant reduction in the average cost, showcasing a decrease of 45%, 52%, and 65% when compared to the DDPG, A3C, and DQN algorithms, respectively. Moving to Fig. 5b and 5c, the analysis focuses on time delay and computation energy consumption, encompassing the transmission and execution of mobile users' tasks. Fig. 5b demonstrates a proportional increase in computation time delay with rising offloading data sizes. However, the proposed algorithm consistently maintains lower time delays compared to the three baseline algorithms. Meanwhile, Fig. 5c reveals a general uptrend in total energy consumption across all algorithms as offloading tasks expand. Remarkably, the proposed algorithm exhibits a notably slower increase in energy consumption compared to the three baseline algorithms. This highlights the algorithm's capacity to effectively minimize computation costs concerning time delay and energy consumption across varying offloading data sizes.

The research analyzes how the computational capabilities of UAVs impact the computation cost mobile users incur when offloading tasks. As indicated in Fig. 6, an evident pattern emerges: as the computational capacity of UAVs rises, there is a gradual decline in average costs observed across all algorithms. This decline in average costs demonstrates a consistent, albeit slow, reduction. To elaborate further, it is important to note that the average cost associated with the three fundamental algorithms exceeds that of the proposed MUAVDDPG algorithm. Upon closer examination by evaluating each algorithm's performance concerning the maximum computation capacity, substantial reductions in average costs are evident. Specifically, when comparing the MUAVDDPG algorithm to the DDPG, A3C, and DQN algorithms, there is a notable decrease in average costs by 40%, 52%, and 66%, respectively. This comparison underscores the significant efficacy of the MUAVDDPG algorithm in significantly reducing costs compared to the established algorithms studied.

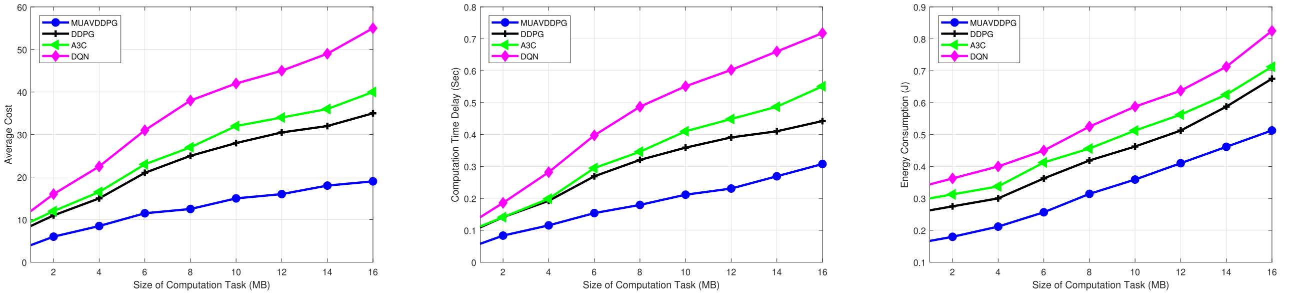


(a) Average Cost

(b) Average Time Delay

(c) Energy Consumption

FIGURE 4. Effects of UAVs on cost, time, and energy consumption.



(a) Average Cost

(b) Average Time Delay

(c) Energy Consumption

FIGURE 5. Effects of data size on cost, time, and energy consumption.

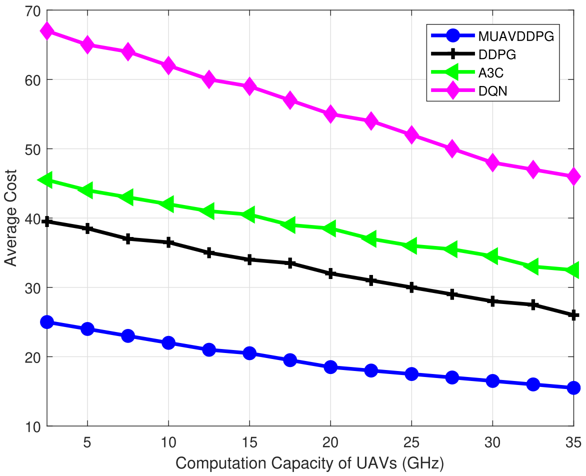


FIGURE 6. UAV computation capacity vs average cost.

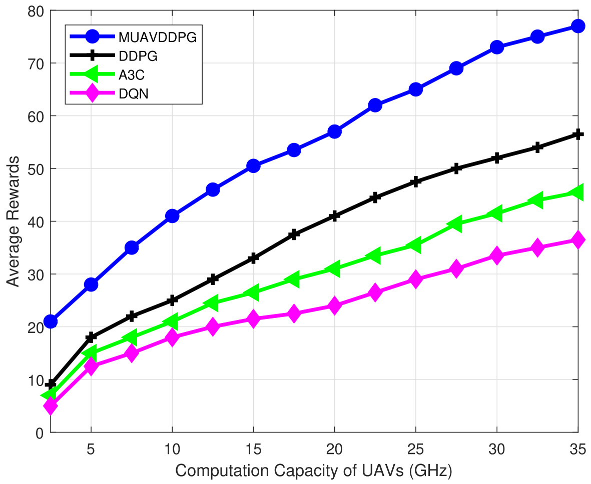


FIGURE 7. UAVs capacity vs average reward.

Fig. 7 and Fig. 8 demonstrate how the computation capacity of UAVs influences the performance metrics of various benchmark algorithms in multi-UAV-assisted MEC environments. Specifically, Fig. 7 illustrates the relationship between UAVs' computation capacity and the average reward achieved by each algorithm, highlighting the increasing trend in average rewards as the computation capacity grows. Notably, the MUAVDDPG algorithm shows the most significant performance enhancement, suggesting that it is particularly effective at utilizing increased computational resources to maximize rewards. This outperformance is consistent across varying capacities, indicating MUAVDDPG's robustness in different computational scenarios. Moreover, Fig. 8 explores the impact of UAVs' computation capacity on the offloading rate, which is a critical measure of how effectively computing tasks are transferred from mobile devices to UAVs. It can be observed from Fig. 8 that all algorithms benefit from increased computation capacities,

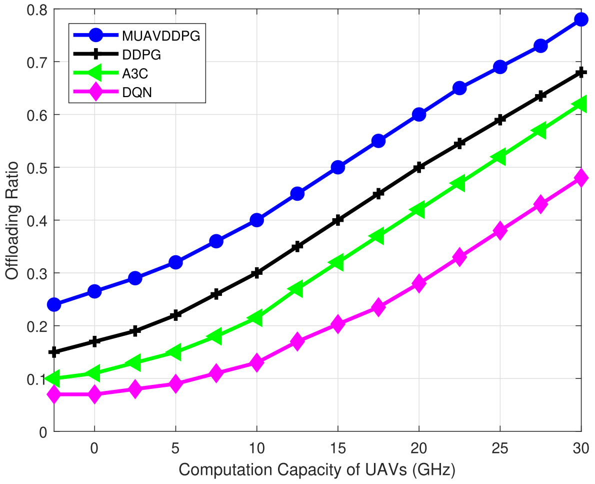


FIGURE 8. UAVs capacity vs offloading rate.

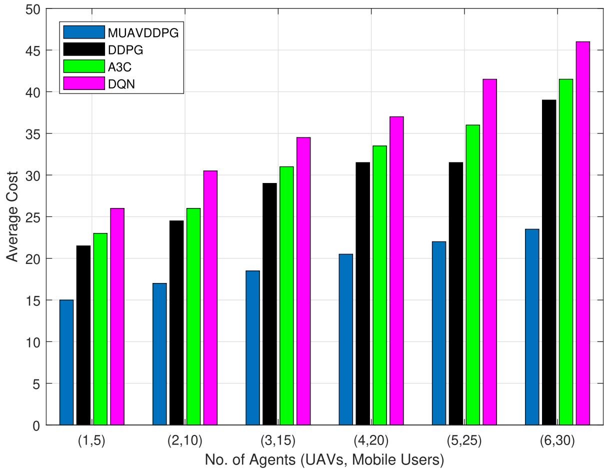


FIGURE 9. Varying No. of UAVs and mobile users vs average cost.

achieving higher offloading rates. The superior performance of MUAVDDPG in achieving higher offloading rates further confirms its enhanced ability to efficiently manage and utilize UAV computational resources, which is crucial for task offloading in complex environments. The consistency of MUAVDDPG's superior performance in both average reward and offloading rates under varying computational capacities supports its stability and robustness, making it a promising option for optimizing UAV-assisted operations in diverse MEC settings.

As the demand from mobile users for resources from UAVs rises, so does the average cost. Collaboration intensifies within the ever-changing network setting when there's a growth in the number of agents - including UAVs and mobile users. This collaboration leads to improved rewards for UAVs and cost reductions for mobile users. To confirm the scalability of our proposed framework, we varied the number of agents in our scenario, ranging from 1 UAV and 5 mobile users to 6 UAVs and 30 mobile users across all algorithms, as depicted in Fig. 9. The proposed MUAVDDPG outperforms the three alternative algorithms. Our observations revealed that average costs also escalate as the number of agents rises. However, the cost associated with the MUAVDDPG algorithm remains lower than the others and sees a more gradual increase. Under MUAVDDPG, agents exhibit higher cooperation and a more favorable experience in contrast to DDPG, A3C, and DQN algorithms as the number of participants grows. Hence, in our proposed scenario, the MUAVDDPG algorithm exhibits greater scalability than the other benchmark algorithms and proves its relevance in multi-agent system scenarios.

A. PRACTICAL IMPLEMENTING AND EVALUATING OF THE PROPOSED ALGORITHM

The evaluation of the proposed MUAVDDPG algorithm against various practical scenarios and benchmark approaches reveals its efficacy in optimizing resource allocation and minimizing computation costs in dynamic network environments. Compared to baseline algorithms such as DQN, A3C, and DDPG, the MUAVDDPG algorithm demonstrates superior performance, as illustrated by convergence analysis and average reward increases. Notably, the MUAVDDPG algorithm converges faster, aligning the interests of UAVs and mobile users to maximize rewards and minimize computation costs through optimal offloading decisions.

Furthermore, across different network scales, including varying numbers of UAVs and offloading data sizes, the MUAVDDPG algorithm consistently outperforms baseline algorithms by efficiently minimizing time delay, energy consumption, and average cost. This performance advantage is attributed to the collaborative nature of the MUAVDDPG framework, which facilitates effective information sharing among agents, reduces resource consumption, and ensures optimal resource allocation.

Additionally, the scalability of the MUAVDDPG algorithm is demonstrated by its ability to maintain lower average computation costs even as the number of agents, including UAVs and mobile users, increases. Overall, these findings underscore the practical utility and generalizability of the proposed algorithm in multi-agent system scenarios, highlighting its potential to address complex task offloading and resource allocation challenges in dynamic network environments.

VI. CONCLUSION

In this paper, we explored task offloading and resource allocation within a resource-constrained multi-UAV-assisted MEC network. Our focus centered on optimizing computation costs, encompassing energy consumption and computation delay, by enabling mobile users to access computational resources from UAVs. Leveraging an optimization problem modeled on dynamic preferences and employing the MDP framework, our goal was to minimize computational demands for mobile users.The MUAVDRL algorithm effectively addresses this challenge by efficiently navigating its dynamic and high-dimensional nature. Rigorous simulation results validate our proposed framework's superior performance and effectiveness when compared to existing state-of-the-art methods, highlighting its substantial potential in practical scenarios. Notably, the simulation outcomes underscore the efficacy of the MUAVDRL algorithm, showcasing its ability to significantly enhance system utility while meeting stringent latency and energy requirements, surpassing the capabilities of other baseline algorithms.

ACKNOWLEDGMENT

The authors are grateful to the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunication (BUPT), Beijing, China, for providing the Laboratory Services.

REFERENCES

[1] S. Raza, S. Wang, M. Ahmed, and M. R. Anwar, "A survey on vehicular edge computing: Architecture, applications, technical issues, and future directions," *Wireless Commun. Mobile Comput.*, vol. 2019, pp. 1–19, Feb. 2019.

[2] S. Raza, W. Liu, M. Ahmed, M. R. Anwar, M. A. Mirza, Q. Sun, and S. Wang, "An efficient task offloading scheme in vehicular edge computing," *J. Cloud Comput.*, vol. 9, no. 1, pp. 1–14, Dec. 2020.

[3] Z. Yu, Y. Gong, S. Gong, and Y. Guo, "Joint task offloading and resource allocation in UAV-enabled mobile edge computing," *IEEE Internet Things J.*, vol. 7, no. 4, pp. 3147–3159, Apr. 2020.

[4] M. Ayzed Mirza, Y. Junsheng, S. Raza, M. Ahmed, M. Asif, A. Irshad, and N. Kumar, "MCLA task offloading framework for 5G-NR-V2X-based heterogeneous VECNs," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 12, pp. 14329–14346, Dec. 2023.

[5] M. A. Mirza, J. Yu, M. Ahmed, S. Raza, W. U. Khan, F. Xu, and A. Nauman, "DRL-driven zero-RIS assisted energy-efficient task offloading in vehicular edge computing networks," *J. King Saud Univ. Comput. Inf. Sci.*, vol. 35, no. 10, Dec. 2023, Art. no. 101837.

[6] M. Ahmed, S. Raza, M. A. Mirza, A. Aziz, M. A. Khan, W. U. Khan, J. Li, and Z. Han, "A survey on vehicular task offloading: Classification, issues, and challenges," *J. King Saud Univ. Comput. Inf. Sci.*, vol. 34, no. 7, pp. 4135–4162, Jul. 2022.

[7] X. Huang, X. Yang, Q. Chen, and J. Zhang, "Task offloading optimization for UAV-assisted fog-enabled Internet of Things networks," *IEEE Internet Things J.*, vol. 9, no. 2, pp. 1082–1094, Jan. 2022.

[8] Z. Chen, Z. Chen, Y. Jia, and L.-C. Wang, "Residual energy-aware caching in energy harvesting-based mobile D2D network," *IEEE Trans. Green Commun. Netw.*, vol. 4, no. 3, pp. 845–857, Sep. 2020.

[9] J. Lu, S. Wan, X. Chen, Z. Chen, P. Fan, and K. B. Letaief, "Beyond empirical models: Pattern formation driven placement of UAV base stations," *IEEE Trans. Wireless Commun.*, vol. 17, no. 6, pp. 3641–3655, Jun. 2018.

[10] H. Cao, W. Zhu, Z. Chen, Z. Sun, and D. O. Wu, "Energy-delay tradeoff for dynamic trajectory planning in priority-oriented UAV-aided IoT networks," *IEEE Trans. Green Commun. Netw.*, vol. 7, no. 1, pp. 158–170, Mar. 2023.

[11] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis, "Mastering the game of go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, pp. 484–489, Jan. 2016.

[12] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, "Asynchronous methods for deep reinforcement learning," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 1928–1937.

[13] M. A. Mirza, J. Yu, S. Raza, M. Krichen, M. Ahmed, W. U. Khan, K. Rabie, and T. Shongwe, "DRL-assisted delay optimized task offloading in automotive-Industry 5.0 based VECNs," *J. King Saud Univ. Comput. Inf. Sci.*, vol. 35, no. 6, Jun. 2023, Art. no. 101512.

[14] X. Chen, H. Zhang, C. Wu, S. Mao, Y. Ji, and M. Bennis, "Performance optimization in mobile-edge computing via deep reinforcement learning," in *Proc. IEEE 88th Veh. Technol. Conf. (VTC-Fall)*, Aug. 2018, pp. 1–6.

[15] H. Ye, G. Y. Li, and B. F. Juang, "Deep reinforcement learning based resource allocation for V2V communications," *IEEE Trans. Veh. Technol.*, vol. 68, no. 4, pp. 3163–3173, Apr. 2019.

[16] F. Sun, N. Cheng, S. Zhang, H. Zhou, L. Gui, and X. Shen, "Reinforcement learning based computation migration for vehicular cloud computing," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2018, pp. 1–6.

[17] S. Zhu, L. Gui, N. Cheng, Q. Zhang, F. Sun, and X. Lang, "UAV-enabled computation migration for complex missions: A reinforcement learning approach," *IET Commun.*, vol. 14, no. 15, pp. 2472–2480, Sep. 2020.

[18] M. Min, L. Xiao, Y. Chen, P. Cheng, D. Wu, and W. Zhuang, "Learning-based computation offloading for IoT devices with energy harvesting," *IEEE Trans. Veh. Technol.*, vol. 68, no. 2, pp. 1930–1941, Feb. 2019.

[19] N. H. Motlagh, M. Bagaa, and T. Taleb, "Energy and delay aware task assignment mechanism for UAV-based IoT platform," *IEEE Internet Things J.*, vol. 6, no. 4, pp. 6523–6536, Aug. 2019.

[20] S. Jeong, O. Simeone, and J. Kang, "Mobile edge computing via a UAV-mounted cloudlet: Optimization of bit allocation and path planning," *IEEE Trans. Veh. Technol.*, vol. 67, no. 3, pp. 2049–2063, Mar. 2018.

[21] T. Zhang, Y. Xu, J. Loo, D. Yang, and L. Xiao, "Joint computation and communication design for UAV-assisted mobile edge computing in IoT," *IEEE Trans. Ind. Informat.*, vol. 16, no. 8, pp. 5505–5516, Aug. 2020.

[22] X. Huang, C. Peng, Y. Wu, J. Kang, W. Zhong, D. I. Kim, and L. Qi, "Joint interdependent task scheduling and energy balancing for multi-UAV-enabled aerial edge computing: A multiobjective optimization approach," *IEEE Internet Things J.*, vol. 10, no. 23, pp. 20368–20382, 2023, doi: 10.1109/JIOT.2023.3288379.

[23] U. Saleem, Y. Liu, S. Jangsher, X. Tao, and Y. Li, "Latency minimization for D2D-enabled partial computation offloading in mobile edge computing," *IEEE Trans. Veh. Technol.*, vol. 69, no. 4, pp. 4472–4486, Apr. 2020.

[24] F. Zhou, Y. Wu, R. Q. Hu, and Y. Qian, "Computation rate maximization in UAV-enabled wireless-powered mobile-edge computing systems," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 9, pp. 1927–1941, Sep. 2018.

[25] J. Chen, Q. Wu, Y. Xu, N. Qi, T. Fang, L. Jia, and C. Dong, "A multi-leader multi-follower Stackelberg game for coalition-based UAV MEC networks," *IEEE Wireless Commun. Lett.*, vol. 10, no. 11, pp. 2350–2354, Nov. 2021.

[26] S. Raza, M. Ahmed, H. Ahmad, M. A. Mirza, M. A. Habib, and S. Wang, "Task offloading in mmWave based 5G vehicular cloud computing," *J. Ambient Intell. Humanized Comput.*, vol. 14, no. 9, pp. 12595–12607, Sep. 2023.

[27] M. A. Ebrahim, G. A. Ebrahim, H. K. Mohamed, and S. O. Abdellatif, "A deep learning approach for task offloading in multi-UAV aided mobile edge computing," *IEEE Access*, vol. 10, pp. 101716–101731, 2022.

[28] A. M. Seid, G. O. Boateng, S. Anokye, T. Kwantwi, G. Sun, and G. Liu, "Collaborative computation offloading and resource allocation in multi-UAV-Assisted IoT networks: A deep reinforcement learning approach," *IEEE Internet Things J.*, vol. 8, no. 15, pp. 12203–12218, Aug. 2021.

[29] I. Khan, S. Raza, W. U. Rehman, R. Khan, K. Nahida, and X. Tao, "A deep learning-based algorithm for energy and performance optimization of computational offloading in mobile edge computing," *Wireless Commun. Mobile Comput.*, vol. 2023, pp. 1–12, May 2023.

MUHAMMAD NAQQASH TARIQ received the bachelor's (B.E.E.) degree from The University of Faisalabad (TUF), Faisalabad, in 2012, and the master's degree in electronics and communication engineering from Beijing University of Posts and Telecommunications, Beijing, China, in 2017. He is currently pursuing the Ph.D. degree with the School of Information and Communication Engineering, Beijing University of Posts and Telecommunications. His research interests include unmanned ariel vehicular communication networks, software defined networking, fog and edge computing, the Internet of Things, machine learning and deep reinforcement learning, and big data analytics.

JINGYU WANG (Senior Member, IEEE) received the Ph.D. degree from Beijing University of Posts and Telecommunications, in 2008. He is currently a Professor with the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications. He has published more than 100 articles in international journals, including IEEE *Communications Magazine*, IEEE Transactions on Cloud Computing, IEEE Transactions on Wireless Communications, IEEE Transactions on Multimedia, and IEEE Transactions on Vehicular Technology. His research interests include the IoV and AIoT, SDN, overlay networks, and traffic engineering.

SALMAN RAZA received the M.C.S. degree (Hons.) in computer science from Bahauddin Zakariya University, Multan, Pakistan, in 2009, the M.S. degree in computer science from G.C. University, Faisalabad, Pakistan, in 2014, and the Ph.D. degree from the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, China. He is currently working as an Assistant Professor with the National Textile University Faisalabad. His current research interests include mobile edge computing, vehicular networks, task offloading, and deep reinforcement learning.

MOHAMMAD SIRAJ (Senior Member, IEEE) received the Bachelor of Engineering degree in electronics and communication engineering from Jamia Millia Islamia, New Delhi, India, the Master of Engineering degree in computer technology and applications from Delhi College of Engineering, New Delhi, and the Ph.D. degree from Universiti Technologi Malaysia. He has worked as a Scientist at the Defense Research and Development Organization, India. Currently he is working as an Assistant Professor of electrical engineering with King Saud University. He has numerous peered publications in well-known international journals and Conferences. His research interests include cognitive wireless networks, wireless mesh networks, sensor networks, the Internet of Things, cloud computing, and telecom optical networks. He is a reviewer of many well-known international journals and conferences.

MAJID ALTAMIMI (Member, IEEE) received the B.Sc.Eng. degree (Hons.) in electrical engineering from King Saud University, Saudi Arabia, in 2004, and the M.Sc. and Ph.D. degrees in electrical and computer engineering from the University of Waterloo, Canada, in 2010 and 2014, respectively. From 2004 to 2006, he worked as a TA at the Department of Electrical Engineering, King Saud University, and the Department of Electrical and Computer Engineering, University of Waterloo, in 2013. In 2015, he joined the Department of Electrical Engineering, King Saud University, as an Assistant Professor. His current research interests include analyzing the energy cost for wireless handheld devices and cloud computing architecture, integrating mobile computing with cloud computing, and studying and designing green ICT solutions.

SAIFULLAH MEMON received the master's degree from the Quaid-e-awam University of Engineering, Science and Technology, (QUEST), in 2015, and the Ph.D. degree in communication engineering from Beijing University of Posts and Telecommunications (BUPT), China, in 2023. He is currently working as an Assistant Professor with QUEST. He is the author of more than 30 research articles. His research interests include wireless communication, sensors, wireless body area networks (WBAN), QoS issues in sensor and ad-hoc networks, routing algorithms, and network security.