AI-Based Mobility-Aware Energy Efficient Resource Allocation and Trajectory Design for NFV Enabled Aerial Networks

Mohsen Pourghasemian, Mohammad Reza Abedi, Shima Salar Hosseini, Nader Mokari, Senior Member, IEEE, Mohammad Reza Javan, Senior Member, IEEE, and Eduard A. Jorswieck, Fellow, IEEE

Abstract—In this paper, we propose a novel joint intelligent trajectory design and resource allocation algorithm based on users’ mobility and their requested services for unmanned aerial vehicles (UAVs) assisted networks, where UAVs act as nodes of a network function virtualization (NFV) enabled network. Our objective is to maximize energy efficiency and minimize the average delay on all services by allocating the limited radio and NFV resources. In addition, due to the traffic conditions and mobility of users, we let some virtual network functions (VNFs) migrate from their current locations to other locations to satisfy the Quality of Service requirements. We formulate our problem to find near-optimal locations of UAVs, transmit power, subcarrier assignment, VNF placement, and VNF scheduling over the UAVs and perform suitable VNF migration. Then we propose a novel hierarchical hybrid continuous and discrete action (HHCDA) deep reinforcement learning method to solve the proposed problem. Finally, the convergence and computational complexity of the proposed algorithm and its performance are analyzed for different parameters. Simulation results show that our proposed HHCDA method decreases the average end-to-end delay by 31.5% and increases the energy efficiency by 40% compared to the state-of-the-art Deep Deterministic Policy Gradient method.

Index Terms—UAVs’ trajectory, energy efficiency, VNF placement and scheduling, VNF migration, deep reinforcement learning, hierarchical hybrid actions.

I. INTRODUCTION

NOWADAYS, daily life critically depends on data networks and customers who expect to have services in any circumstances. For instance, gatherings of a large number of people in a short period of time in one place, sports matches, or heavy traffic in a crowded crossover, cause an overload of the local network. Moreover, during a natural disaster in which lots of infrastructures are destroyed and the conventional base stations (BSs) do not work properly, users can hardly access the network. One of the most effective approaches to overcome the aforementioned challenges is to temporarily deploy the unmanned aerial vehicle assisted (UAV-assisted) networks, in which UAVs can play the roles of BSs, in addition to their abilities to move and create an ad-hoc network [1]. In addition to aerial networks, network function virtualization (NFV) is one of the key technologies for the next generation cellular networks, which can reduce the deployment cost of network functions [2]. On the other hand, real time video streaming application is a perfect usage of an NFV-enabled network which needs agile and efficient deployment in huge gatherings such as disasters and sports matches. Using NFV can considerably reduce operational expenditure (OPEX) and capital expenditure (CAPEX) costs [3] of such video streaming applications. Moreover, it provides subnet isolation and network slicing for extreme customizability and flexibility [2] and is crucial to enable quality-of-service (QoS) for the aforementioned applications in the UAV assisted networks. More specifically, when UAVs are integrated to support such applications, they can effectively extend their capacity [1]. Indeed, given their computing, storage, and networking capabilities, UAVs can be considered as a flexible platform to both support cost-effective communications and enable the shared use of resources in the next generation of mobile networks [1]. In this context, the NFV paradigm can be smoothly integrated into UAV networks with the aim of enhancing the performance of traffic processing delivered in these mobile networks [4]. However, despite the huge effort that has been put into enhancing resource orchestration, there still exist issues and hurdles that should be efficiently addressed. We consider an UAV assisted and NFV-enabled video sharing (streaming) application and proposed an agile and energy efficient framework to support such application. The aim of considering the video streaming application is that in huge gathering in disasters, the broadcast system may not be able to deliver the news and other necessary video streams to people. Therefore, an agile and distributed
video streaming framework can be potentially a promising solution to support delivering video streams [5]. Changing the locations of mobile users and their service requests will oblige NFV enabled networks to perform VNF migration in order to satisfy the network QoS requirements [3]. Moreover, new requirements of emerging services like real time video streaming require a high data rates [6]. To provide those features, we can use a new spectral efficient non-orthogonal multiple access (NOMA) technology like power domain-NOMA (PD-NOMA) which can improve the data rate of users. Compared with previous orthogonal multiple access (OMA) technologies like orthogonal frequency-division multiple access (OFDMA), PD-NOMA can support more users offer higher data rates, and improve fairness [7]. The optimal resource allocation, function placement, and scheduling in NFV-enabled networks need more attention since NFV technology provides more flexibility for resource limited wireless networks. On the one hand, most of the resource allocation and VNF placement problems are non-convex and NP-hard, therefore, it is very difficult to obtain the optimal solutions by traditional methods. On the other hand, as the number of optimization variables increases, the traditional methods may not achieve the sub-optimal solution and their performance complexity increases exponentially. Also, the traditional methods must be solved every time the network parameters change. Therefore it is impractical to use traditional methods for such complex problems. To tackle these challenges, artificial intelligence (AI)-based algorithms can be effectively used to solve such optimization problems. One of the best AI-based solutions is reinforcement learning (RL) in which an agent observes the environment, takes an action, and based on the received reward chooses a strategy to reach an efficient policy [8]. However, RL cannot operate well in an environment which has large state space and it needs a huge amount of resources such as memory and processor. In this regard, deep RL (DRL) methods are introduced which utilize deep neural networks to process the learning procedure [8]. DRL methods can provide solutions for modern networks which arise in large environments with high computational complexity. In addition, since the centralized single-agent methods lead to large signaling overhead in large networks, DRL based solutions can be used in a decentralized multi-agent manner which has become more popular in recent networks such as UAV-assisted networks [9]. In this approach, after offline training, each agent takes its own decision independently of the others. Therefore, it is not required to obtain global information and can reduce the signaling overhead in large networks [9].

B. Literature Review

1) VNF Placement and Scheduling: The authors in [10] considered VNF placement for a single service and focus on minimizing the provisioning cost of a single service. However, they do not consider the impact of the VNF placement of one service on other services. Reference [11] employed a queuing-based model and formulated a VNF placement problem to minimize the ratio between the actual and maximum allowed latency. The authors in [12] investigated the NFV enabled network, and developed optimal function scheduling to minimize the processing and transmission delay and they solved their proposed problem based on a Genetic algorithm. However, they do not consider VNF migration as well as users’s mobility in VNF placement. The authors adopted NFV in [13], in which they provided tactile Internet (TI) service. They considered joint VNF placement and radio resource allocation, while guaranteeing the required delay for TI service by finding the optimal transmit power and function scheduling. These works only considered static VNF placement where the migration of VNFs is not allowed. Also, none of them considered VNF mapping on UAVs.

2) Dynamic VNF Migration: In [14], the authors considered the impact of users’ mobility on VNF placement in a cellular network, aiming to minimize the cost of VNF migration. Authors in [15] proposed a dynamic VNF placement and migration in industrial IoT aiming to jointly minimize the age of information and VNF cost using DRL. In [16], the authors investigated the mobility of users that may cause the NFV to enable the network to re-route the VNF placement to handle the requested service. In [17], the authors combined mechanisms for prediction and migration together to optimize the cost of VNFs migration and guarantee the quality of services that may be decreased by resource limitations. They proposed a real-time VNF migration algorithm based on a deep belief network to predict future resources requirements. They investigated bandwidth utilization and migration problems jointly. Nevertheless, all of the existing works in VNF migration considered one threshold for migration cost and migration delay regardless of performing VNF migration, so whenever there is no VNF migration, this threshold results in a local optimal solution. We consider two threshold values for latency, one threshold for times when we do have VNF migration, and another one is for times when we do not have VNF migration.

3) UAV Assisted NFV Enabled Networks: The authors in [18] considered space-air-ground integrated network (SAGIN) and proposed a VNF placement framework based on static network conditions and formulated their problem as integer non-linear programming. Their objective was to jointly minimize the VNF placement cost and maximize the service provisioning. They do not consider VNF migration and hence it may poorly perform in dynamic networks. In [19], a dynamic VNF placement and scheduling framework was proposed in the SAGIN network with the objective of minimizing the VNF placement cost. Although they considered the mobility of users for VNF placement and migration, the trajectory design of UAVs which significantly improves the VNF placement performance was missed. They considered the UAVs as stationary VNF nodes. In addition, the SAGIN network utilizes low orbit satellites as NFV enabled nodes which needs special users to connect with. Also, using BSs as NFV nodes, causes the SAGIN network to be impractical in disaster situation deployments. The authors in [20], proposed a VNF placement and routing traffic scheme by prior sensing the running conditions of satellite user services in satellite ground station networks. The information of all user services is predictable in satellite control centers via satellite mission planning. The location-aware resource allocation algorithms based on Greedy
for evaluating the performance of their proposed resource allocation in small scale networks problems have also been considered. In [21], the authors investigated the VNF placement problem in satellite edge computing with the aim of maximizing the number of allocated IoT users, while minimizing the overall deployment cost, which is composed of energy consumption, bandwidth, and service delay costs. They implemented a decentralized resource allocation algorithm based on a potential game to tackle the VNF placement problem by finding a Nash equilibrium. An aerial platform consisting of multiple UAVs that supports the traffic demand of a ground network is considered in [22]. They concentrated on optimizing the placement and routing decisions of UAVs on a quasi-static terrestrial network, and for a mobile terrestrial network that considers the impact of the speed of the UAVs.

4) Deep Reinforcement Learning Based Solution: AI-based methods are promising solutions for solving problems that are NP-hard and non-convex [23]. However, depending on the problem, the state-of-the-art methods such as deep Q network (DQN) and deep deterministic policy gradient (DDPG) may lose to perform optimally due to their restrictions. For instance, DQN cannot perform optimally in an environment with continuous actions [23], also DDPG cannot be directly used in an environment that contains discrete actions [24]. On the one hand, there are environments that have both discrete and continuous actions simultaneously, and for which the parameterized actions space based DRL [25] methods are perfectly suited. On the other hand, in some situations, like in our system model, the discrete and continuous actions in the environment are dependent on each other. For instance, the discrete actions (like VNF placement) should be taken based on the continuous ones (like transmitting powers) and the existing DRL methods cannot be directly used to solve such a hybrid environment. In [26] a compound action actor critic (CA2C) method based on DDPG and DQN method was proposed to support an UAV-assisted task assignment and trajectory design which has continuous and discrete actions simultaneously. However, the complexity of their CA2C method increases as the number of discrete actions increases, especially in multi-agent deployment. On the other hand, CA2C is not a good candidate for the partially observable environment since it needs full state information of the environment for action selection. From the above literature review, there does not exist an UAV-assisted network which considers UAVs as NFV enabled nodes with UAVs trajectory design so that the VNF placement and migration approach with the limited radio and NFV resources are performed by the high mobility UAVs, aiming to satisfy the QoS of the mobile users. In some situations like a disaster, there is no available BS to be considered as NFV enabled node. Therefore, the SAGIN network cannot be considered a candidate for these kinds of situations. In our proposed VNF migration enabled UAV-assisted network, the locations of UAVs not only should be changed related to the locations of users but also should be determined based on the requested services of users. Moreover, the limitation of power budget and available bandwidths between UAVs may result in an increase in the delay of services. Another challenge for these kinds of networks is the central processing unit (CPU) limitation for UAV nodes, which causes that they can run only a few number of functions.

In some partially observable environments in which multiple agents do not have full observation of the environment, like our proposed UAV-assisted NFV-enabled network, the VNF placement (discrete actions) is dependent on what are the resource allocation and UAVs trajectory design (continuous actions). In other words, the near optimal VNF placement can be performed if the locations of the UAVs and the allocated services are known to the agents. Therefore we proposed a DRL method that not only considers the dependency of discrete and continuous actions but also considers the hierarchy for discrete and continuous action selection. Motivated by the above challenges, we propose a hybrid hierarchical continuous discrete action (HHCDLA) which chooses the best VNF placement and migration strategy based on the allocated resources to the services and UAVs locations.

C. Our Contributions

In this paper, we consider an UAV-assisted network, where the UAVs are utilized as NFV nodes to play BS roles. We formulate our problem as an optimization problem, in which we aim to jointly maximize the energy efficiency (EE) and minimize services’ delay by finding optimal transmit power, subcarrier assignment, UAVs’ location, service function chaining, function placement and scheduling subject to network, and resource constraints. According to the discussion above, the primary contributions of our paper are listed as follows:

- To the best of our knowledge, the existing works in the literature do not consider energy-efficient and low latency VNF placement, scheduling, and migration over UAVs with UAV trajectory design. We assume that UAVs are NFV enabled nodes on which the VNFs can be placed in an optimal way. Also, we consider that VNF migration can take place due to users’ mobility and their heterogeneous service requests. Hence, we propose an energy-efficient UAV trajectory design and resource allocation algorithm based on joint users’ mobility and service requests, in which the EE of the UAVs is maximized and the delay of the requested services is minimized. In addition, by considering two weighting factors, we provide a balance between energy efficiency and service delay in the network.

- Considering our environment as partially observable to the UAVs with both discrete and continuous actions including transmit power allocation, subcarrier assignment, UAV trajectory design, VNF placement, and scheduling, we propose an HHCDLA DRL method and extend it to a decentralized scheme to solve our problem.

- Through the numerical results, we show that our proposed NFV-enabled system model with its novel HHCDLA solution outperforms the state-of-the art DRL methods in decreasing UAV movement, increasing energy efficiency, and decreasing service latency. Moreover, it achieves the highest accumulated reward with the lowest convergence time.
D. Organization and Notation

The remainder of this paper is organized as follows: In Section II, the system model is presented. In Section III, the optimization problem is formulated, and the elements of the learning-based algorithm are determined in Section IV. Simulation results are discussed in Section V, and finally, in Section VI, the conclusion is presented.

Notations: Vectors and matrices are denoted by boldface small and big letters, respectively. ∇ₐ Q determined the gradient of Q over a, and Pr(J = J′) is used to show the probability when J = J′. 𝔼ₛ{⋯} is the expectation function over s, |⋯| denotes the number of elements in the set, and ∥⋯∥ denotes the Euclidean norm.

II. SYSTEM MODEL

We consider an UAV-assisted network, where each UAV serves a number of users requesting different services. There is a set of UAVs as 𝑈 indexed by 𝑢, where |𝑈| = 𝑈 denotes the number of UAVs. Each UAV is equipped with a single antenna. A time set 𝑇 is indexed by 𝑡, where |𝑇| = 𝑇 denotes the number of time slots. The time slots have an equal duration of 𝑡. The location of UAV 𝑢 at time slot 𝑡 is denoted by 𝑞_𝑢(𝑡) = (𝑥_𝑢(𝑡), 𝑦_𝑢(𝑡), 𝑧_𝑢(𝑡)) where 𝑥_𝑢(𝑡) ∈ ℝ, 𝑦_𝑢(𝑡) ∈ ℝ, and 𝑧_𝑢(𝑡) ∈ ℝ are the horizontal, vertical, and altitude coordinates of UAV 𝑢 at time slot 𝑡, respectively. We assume that the maximum UAV’s velocity is 𝑊_{max}. Therefore, each UAV can travel at a maximum distance as 𝐷_{max} = 𝑊_{max} between two successive time slots. To avoid collisions, the minimum safe distance, 𝐷_{min} must be kept between UAVs. Thus, for UAV 𝑢, the following mobility constraint must be satisfied to restrict its trajectory [27]:

\begin{align}
\text{C1: } & \|\tilde{q}_u(t + 1) - \tilde{q}_u(t)\| \leq D_{max}, \\
\text{C2: } & \|\tilde{q}_u(t) - \tilde{q}_u'(t)\| \geq D_{min}, \forall u \neq u',
\end{align}

Constraint C1 denotes that the maximum movement of the UAVs at each time slot is restricted by their maximum flying speed. Constraint C2 indicates that the minimum distance between two UAVs should not be less than the minimum value, 𝐷_{min}. The user set which is served by UAV 𝑢 is denoted by 𝑈_𝑢 with |𝑈_𝑢| = 𝑈. Each user can be served by only one UAV at a time and is equipped with a single antenna. The total set of users that are served by UAVs is denoted by 𝑈 with |𝑉_𝑢| = 𝑈. We construct a set of total nodes that contains the users and the UAVs as 𝑁 = {𝑈, 𝑈} with |𝑁| = |𝑉_𝑢|. The location of user 𝑘 at time slot 𝑡 is represented by 𝑞_𝑘(𝑡) = (𝑥_𝑘(𝑡), 𝑦_𝑘(𝑡)) ∈ ℝ which are the horizontal and vertical coordinates of user 𝑘, respectively. Since the users change their locations at each time slot, we consider the random walk model for their mobility [28]. The users move uniformly in any direction with a random speed between 0 and 𝑊_{max}. We assume that the network uses three separate bands for links between UAV-UAV, UAV-User, and User-UAV so that there is no overlap between these links. Each bandwidth is divided into several orthogonal subcarriers. We assume that the bandwidth of UAV-UAV communication, 𝐵 is divided into a set of 𝑉 subcarriers each with bandwidth 𝐵_1 denoted by ν, the bandwidth of UAV-User communication denoted by 𝐵 is divided into a set of 𝐿 subcarriers, each with bandwidth 𝐵_1 denoted by l, and the bandwidth of User-UAV communication indicated by 𝐵 that is divided into a set of 𝐸 subcarriers, each with bandwidth 𝐵_1 denoted by e. Fig. 1 shows the considered system model.

A. Communication Model

1) UAV-User Communication (DL Access Network): We consider a PD-NOMA technique for communication between UAVs and users [29], where the maximum transmit power of
Therefore, the channel power gain is obtained as:

\[ \hat{g}_{lk}(t) \] for the NLoS link, i.e.,

\[ \hat{g}_{lk}(t) = 1 \] when user \( k \) is served by UAV \( u \) on subcarrier \( l \) at time slot \( t \) and otherwise \( \hat{g}_{lk}(t) = 0 \), and we ensure that each user can only be served by only one UAV in DL as follows:

\[
C_4 : \left[ \sum_{l \in L} g_{uk}(t) \right] \left[ \sum_{u' \in \mathcal{U}, u' \neq u} \sum_{l \in L} g_{u'k}(t) \right] = 0, \forall u, l.
\] (2)

Moreover, by the following constraint, we enforce that the transmit power of UAV \( u \) over all subcarriers to its serving users do not exceed the power budget of UAV \( u \), \( P_{u}^{\text{max}} \):

\[
C_5 : \sum_{k \in \mathcal{K}_u} \sum_{l \in L} g_{uk}(t) p_{lk}(t) \leq P_{u}^{\text{max}}.
\] (3)

UAV sends the signal to user \( k \) on subcarrier \( l \) through a channel with the channel power gain \( \hat{g}_{uk}(t) \), which consists of line of sight (LoS) and non-line of sight (NLoS) links as follows [30]:

\[
\hat{h}_{lk}(t) = \begin{cases} 
\hat{h}_{l}^{\text{LoS}}_{uk}(t) = \left( \frac{d_{uk}(t) \pi f_{c}}{c} \right)^{-\kappa}, & \text{LoS}, \\
\hat{h}_{l}^{\text{NLoS}}_{uk}(t) = \xi \left( \frac{d_{uk}(t) \pi f_{c}}{c} \right)^{-\kappa}, & \text{NLoS},
\end{cases}
\] (4)

where \( d_{uk}(t) \) is the distance between UAV \( u \) and user \( k \) at time slot \( t \), \( \kappa \) is the path loss component, \( \xi \) is the additional path loss component for the NLoS link, \( f_{c} \) denotes the carrier frequency of subcarrier \( l \), and \( c \) is the light speed. The probability of LoS link, i.e., \( P_{\text{LoS}} \), and the probability of NLoS link, i.e., \( P_{\text{NLoS}} \), are obtained as follows [4]:

\[
\begin{align*}
\Pr_{\text{LoS}} &= \Pr(\hat{h}_{l}^{\text{LoS}}_{uk}(t)) \\
&= \frac{1}{1 + \beta_1 \exp \left(-\beta_2 \left( \frac{180}{\pi} \arctan \left( \frac{\hat{h}_{l}^{\text{LoS}}_{uk}(t)}{d_{uk}(t)} \right) - \beta_1 \right) \right)},
\end{align*}
\] (5)

\[
\begin{align*}
\Pr_{\text{NLoS}} &= \Pr(\hat{h}_{l}^{\text{NLoS}}_{uk}(t)) = 1 - \Pr(\hat{h}_{l}^{\text{LoS}}_{uk}(t)),
\end{align*}
\] (6)

where \( \beta_1 \) and \( \beta_2 \) are related to the operation environment [30]. Therefore, the channel power gain is obtained as:

\[
\hat{h}_{lk}(t) = \Pr_{\text{LoS}} \hat{h}_{l}^{\text{LoS}}_{uk}(t) + \Pr_{\text{NLoS}} \hat{h}_{l}^{\text{NLoS}}_{uk}(t).
\] (7)

The received signal from user \( k \) on subcarrier \( l \) served by UAV \( u \) contains the signal that its UAV and other UAVs send on subcarrier \( l \):

\[
\psi_{uk} = \hat{h}_{uk}(t) \zeta_{uk}(t) + \sum_{u' \in \mathcal{U}, u' \neq u} g_{u'k}(t) \hat{h}_{u'k}(t) \zeta_{u'k}(t) + n_{k}.
\] (8)

From the PD-NOMA concept, to utilize SIC for separating different user’s signals, an ordering among users served by each UAV should be achieved. The channel power gain to interference and noise ratio (CINR) is considered as the criterion for ordering users of an UAV, thus CINR of user \( k \) served by UAV \( u \) can be computed as below [31]:

\[
\text{CINR}_{uk}(t) = \frac{\hat{h}_{uk}(t) \hat{f}_{uk}(t)}{\sum_{u' \in \mathcal{U}, u' \neq u} \hat{h}_{u'k}(t) + B_{1} N_{0}},
\] (9)

where \( B_{1} \) denotes the bandwidth of each subcarrier, and \( N_{0} \) is the power spectral density of the noise on subcarrier \( l \). The users are ordered in decreasing order from the strongest user, i.e., the user with the highest CINR to the weakest user, i.e., the user with the lowest CINR for SIC decoding at the receiver sides. By ordering users, each user can remove the signal of other users who have lower CINR than its signal, and consider the signal of others who have higher CINR as noise. The instantaneous signal to interference and noise ratio (SINR) at user \( k \) on subcarrier \( l \) which is served by UAV \( u \) is computed as follows:

\[
\text{SINR}_{uk}(k, t) = \frac{\hat{g}_{uk}(t) \hat{f}_{uk}(t) p_{lk}(t)}{h_{uk}(t) \sum_{u' \in \mathcal{K}_{u}, u' \neq u} \hat{g}_{u'k}(t) p_{l,u'k}(t) + I_{uk}^{l}(t) + B_{1} N_{0}},
\] (10)

where \( I_{uk}^{l}(t) \) is the inter UAV interference on user \( k \) which is served by UAV \( u \) and is computed as \( I_{uk}^{l}(t) = \sum_{u' \in \mathcal{U}, u' \neq u} g_{u'k}(t) p_{l,u'k} \hat{h}_{u'k}(t) \). The following constraint arises to perform the SIC successfully:

\[
\text{C6: } \text{SINR}_{uk}(i, t) - \text{SINR}_{uk}(k, t) \geq 0, \quad \text{CINR}_{uk}(i, t) > \text{CINR}_{uk}(k, t), \quad \forall u \in \mathcal{U}, k \in \mathcal{K}_{u}, l \in L,
\] (11)

where \( \text{SINR}_{uk}(i, t) \) denotes the SINR of user \( k \) on subcarrier \( l \) at UAV \( u \) at user \( i \) [32]. Based on this constraint, the SINR of the worse user (with lower CINR) at the better users (with higher CINR) should be higher than its own SINR. The data rate for user \( k \) of UAV \( u \) on subcarrier \( l \) at time slot \( t \) is obtained as follows:

\[
r_{uk}^{l}(t) = B_{1} \log_{2} \left( 1 + \text{SINR}_{uk}^{l}(t) \right), \forall k \in \mathcal{K}_{u}, u \in \mathcal{U}.
\] (12)

2) User-UAV Communication (UL Access Network): For the User-UAV communication, we consider the UL PD-NOMA scheme where multiple users transmit their signals to the UAVs. The received signal at UAV \( u \) on subcarrier \( e \) is given by

\[
\theta_{u} = \sum_{k \in \mathcal{K}_{u}} \tilde{g}_{ku}(t) \tilde{h}_{ku}^{e}(t) \tilde{p}_{ku}(t) \tilde{s}_{k} + \sum_{u' \in \mathcal{U}, u' \neq u} \sum_{k \in \mathcal{K}_{u'}} \tilde{g}_{ku}(t) \tilde{h}_{ku}^{e}(t) \tilde{p}_{ku}(t) \tilde{s}_{k} + \tilde{n}_{u}.
\] (13)

\footnote{For SIC decoding, ordering is an important issue for improving the PD-NOMA performance. The optimal ordering can be obtained by solving the optimization problem to find the optimal ordering variables. However, this approach has high computational complexity, so many works use CINR-based ordering which is suitable for a multi-UAV system with inter UAVs interferences [31].}
Similar to DL access network, we define the parameters for UL access network on different subcarrier \( e \), \( \tilde{p}_{ku}(t) \) denotes the transmit power of user \( k \) assigned to UAV \( u \) on subcarrier \( e \), \( \tilde{S}_{ke}^c \) denotes the information of user \( k \) transmitted on subcarrier \( e \), with \( \mathbb{E}[(\tilde{S}_{ke}^c)^2] = 1 \), \( \tilde{N}_{0}^e \) is the Gaussian noise with power spectral density \( N_{0} \), \( \tilde{g}_{e}(t) \) is a binary variable with \( \tilde{g}_{e}(t) = 1 \) when user \( k \) is assigned to UAV \( u \) on subcarrier \( e \) at time slot \( t \) in the UL, and otherwise \( \tilde{g}_{e}(t) = 0 \), and we ensure that each user can only be allocated to one UAV in UL as follows:

\[
C7: \left[ \sum_{e \in \mathcal{E}} \tilde{S}_{ke}^c(t) \right] \left[ \sum_{u' \in \mathcal{U}, u' \neq u} \sum_{e \in \mathcal{E}} \tilde{g}_{e}^c(t) \right] = 0, \forall e, k. \tag{14}
\]

Based on (2) and (14), each user can be allocated to different UAVs separately in DL and UL transmission. In the following constraint, we enforce that the transmit power of user \( k \) over all subcarriers to its assigned UAV does not exceed the power budget of user \( k \), \( \tilde{p}_{k}^{\text{max}} \) :

\[
C8: \sum_{u \in \mathcal{U}} \sum_{e \in \mathcal{E}} \tilde{g}_{e}^c(t) \tilde{p}_{ku}^c(t) \leq \tilde{p}_{k}^{\text{max}}. \tag{15}
\]

User \( k \) sends a signal to UAV \( u \) on subcarrier \( e \) through a channel with the channel power gain \( \tilde{h}_{ke}^c(t) \). Similar equations as (5), (6), and (7) can be used to determine the UL channel gains, \( \tilde{h}_{ke}^c(t) \). By the ordering of users based on their channel gain, \( \tilde{h}_{ke}^c(t) > \tilde{h}_{ke}^c(t') \) \( \forall k^e, k \in \mathcal{K}_u \), the UAVs perform SIC to separate the signals of different users. The UAVs first decode the strongest user’s signal by considering the signals of other users as noise, then after subtracting it from the received signal, decode the next user’s signal from the remaining, and so on. The instantaneous SINR of user \( k \) on subcarrier \( e \) at UAV \( u \) is computed as follows:

\[
\text{SINR}_{ku}^c(t) = \frac{\tilde{h}_{ke}^c(t) \tilde{p}_{ku}^c(t)}{\tilde{T}_{ku}^c(t)} + \tilde{I}_{ku}^c(t) + B_1 \tilde{N}_0^e, \tag{16}
\]

where \( \tilde{T}_{ku}^c(t) \) is the inter-user interference at UAV \( u \) on subcarrier \( e \) which is computed as below:

\[
\tilde{T}_{ku}^c(t) = \sum_{u' \in \mathcal{U}, u' \neq u} \sum_{k \in \mathcal{K}_{u'}} \tilde{g}_{e}^c(t) \tilde{h}_{ke}^c(t) \tilde{p}_{ku'}^c(t). \tag{17}
\]

The received data rate from user \( k \) on UAV \( u \) on subcarrier \( e \) with bandwidth \( B_1 \) at time slot \( t \) is obtained as follows:

\[
\hat{r}_{ku}^c(t) = B_1 \log_2 \left( 1 + \text{SINR}_{ku}^c(t) \right), \quad \forall k \in \mathcal{K}, u \in \mathcal{U}. \tag{18}
\]

3) UAV-UAV Communication: We consider a binary variable to show the assignment of subcarriers as binary \( \sigma_{u'u'}^v \), which is equal to 1 when subcarrier \( v \) is utilized for communication between UAV \( u \) and UAV \( u' \), otherwise, 0. Subcarrier \( v \) is chosen from subcarrier set \( \mathcal{V} \) with size \( V \). The data rate between two UAVs can be calculated as below:

\[
\hat{r}_{u'u'}^v(t) = B_1 \log_2 \left( 1 + \frac{\sigma_{u'u'}^v \tilde{p}_{u'u'}^c(t) \tilde{h}_{u'u'}^c(t) \tilde{h}_{u'u'}^c(t)}{B_1 \tilde{N}_0^e} \right), \tag{19}
\]

where \( \tilde{p}_{u'u'}^c(t) \) is the transmit power over link UAV \( u \) to UAV \( u' \) on subcarrier \( v \), and \( \tilde{h}_{u'u'}^c(t) \) is the channel power gain between UAVs \( u \) and \( u' \) computed as (4) and just contains the LoS link [30]. To satisfy orthogonality among subcarriers, the following constraint must be met:

\[
C9: \sum_{u \in \mathcal{V}} \sum_{u' \in \mathcal{U}, u' \neq u} \sum_{v} \sigma_{u'u'}^v(t) \leq 1. \tag{20}
\]

B. Network Function Virtualization

We assume a set of communication services \( O_{\text{Total}} = \bigcup_{i=1}^{J} O_i \) where \( J \) is the number of services. Service \( O_i \) is described as \( O_i = \{k_i^S, k_i^D, \mathcal{F}_i, b_i, t_i, \tau_i\}. \) Each service flow \( i \), denoted by the pair of source user and destination user \((k_i^S, k_i^D)\), \( \mathcal{F}_i = \{f_{i1}, \ldots, f_{ij}, \ldots, f_{iJ}\} \) is a set of \( J \) VNFs, where \( f_{ij} \) denotes function \( j \) of service \( O_i \) and is assumed to be placed over UAVs. \( b_i \) is the required bit rate of service \( O_i \), \( T_i \) denotes the time duration of service \( O_i \), and \( \tau_i \) denotes the maximum delay tolerated by service \( O_i \). We consider that each UAV can run all functions of services, but based on its available resources and network conditions, the UAV finds the near optimal strategy for running the functions of services. The number of functions and their scheduling should be optimally performed by each UAV. Note that different VNFs of each specific service could be executed on the same or different nodes (UAVs) based on the residual resources of each UAV. However, a VNF of each service only can be executed on a node. In other words, a VNF of each service cannot be split on different nodes. At each time slot, we consider a binary function assignment variable \( x_{fij}^O(t) \) where \( x_{fij}^O(t) = 1 \) means that function \( f_{ij} \) is placed on UAV \( u \) at time slot \( t \) for user \( k \). Note that in some cases, there are no direct links between two UAVs, therefore, some UAVs are utilized as relays that just receive the data and forward it to the next nodes. We define a binary variable \( y_{uk}^O(t) \) to represent if UAV \( u \) acts as a relay for service \( O_i \) route of user \( k \), \( y_{uk}^O(t) = 1 \), and otherwise \( y_{uk}^O(t) = 0 \). In addition, we define two binary variables \( \chi_{unk}^O(t) \) and \( \psi_{unk}^O(t) \), to represent the path chain of each service. \( \chi_{unk}^O(t) \) is equal to 1 when function \( f_{ij} \) of user \( k \) is processed by UAV \( u \) and is sent to node \( n \), \( \psi_{unk}^O(t) \) is equal to 1 when node \( n \) only relays the traffic of service \( O_i \) of user \( k \) toward node \( n \), otherwise, they are equal to 0. To ensure that each function can only placed in one node,\(^2\) we consider the following constraint:

\[
C10: \sum_{u \in \mathcal{U}} x_{fij}^O(t) = 1, \forall i, j, k. \tag{21}
\]

In addition, with the considered flow conservation constraint, we ensure that for a requested service, the traffic leaves its source node \( n \) and eventually arrives at its destination node \( n' \) as follows:

\[
C11: \sum_{n \in \mathcal{U}} \left( \chi_{n'ink}^O + \psi_{n'ink}^O \right) - \sum_{n \in \mathcal{U}} \left( \chi_{nk'\bar{n}}^O + \psi_{nk'\bar{n}}^O \right) = \begin{cases} 1 & n = k^S, \\ -1 & n = k^D, \\ 0 & \forall i, n, k \in \mathcal{K}, \end{cases} \tag{22}
\]

\(^2\)One specific function of each user must not be split to run in different nodes (UAVs).
The new requested function of services during a time slot is considered at the start of the next time slot. An example of the procedure for the management of 3 UAVs and two time slots is shown in Fig. 2. As can be seen in Fig. 2, after processing a function, it is sent to the next node, which causes processing, transmission, and propagation delay. We considered that each UAV runs an instance of each VNF for different users. In other words, if each UAV receives multiple requests of different users to run one specific VNF, then UAV runs single instance of that specific VNF and schedule all requested VNFs for its assigned users. As demonstrated in Fig. 2, at time slot \( t = 1 \), UAV1 runs VNF1 for user1, user2, and user3, then schedules them so that it first runs VNF1 of user3 \( f_{31} \) and finally runs VNF1 of user1 \( f_{11} \). It is worth noting that the VNF placement and scheduling must be performed properly so that the remaining CPU of each UAV and time duration for scheduling not violated. It is also worth noting that each VNF of each user is run on UAVs based on the number of packets. For example, user1 has 2 packets, therefore its corresponding functions \( (f_{11}, f_{12}, \text{and } f_{13}) \) run at two consecutive time slots. The propagation delay depends on the distance between two communication nodes. The propagation delay between nodes \( u \) and \( u' \) for VNF \( j \) of service \( O_i \) of user \( k \) at time slot \( t \) is equal to

\[
\tau_{PDD}^{uj} (t) = \left( x_{uk}^j (t) + y_{uk}^j (t) \right) \left( y_{u'u'}^j (t) + x_{u'u'}^j (t) \right) d_{u'u'} (t) / c.
\]

and (23b) and (23c) ensure that each UAV just plays a role as a function executor or relay. The transmission delay of VNF \( j \) of service \( O_i \) of user \( k \) depends on the assigned capacity of the utilized link capacity:

\[
\tau_{TD}^{uj} (t) = \left( x_{uk}^j (t) + y_{uk}^j (t) \right) \left( y_{u'u'}^j (t) + x_{u'u'}^j (t) \right) b_j (t) / \sigma_{u'u'},
\]

where \( \sigma_{u'u'} \) is the total data rate capacity between two connected UAV \( u \) and UAV \( u' \). The processing delay depends on the function which should be executed in a node and the resource available in the node. We consider a processing delay of VNF \( j \) of service \( O_i \) of user \( k \) at time slot \( t \) at node \( u \) as \( \tau_{PR}^{uj} (t) \) which is computed as follows:

\[
\tau_{PR}^{uj} (t) = x_{uk}^j (t) b_j c_o / C_u,
\]

where \( c_o \) is the required CPU cycle to process one bit. \( C_u \) is the total CPU cycle at UAV \( u \). The utilized CPU cycle by each UAV should not exceed the available CPU cycle at each time slot, which is ensured as:

\[
C12: \sum_{k \in K} \sum_{t \in T} x_{uk}^j (t) c_o b_j \leq C_u, \forall u \in U.
\]

The utilized bandwidth in each link should not exceed the available bandwidth at each time slot, which is ensured as:

\[
C13: \sum_{k \in K} \sum_{t \in T} \left( y_{u'u'}^j (t) + x_{u'u'}^j (t) \right) b_j \leq \sum_{u' \in V} \sigma_{u'u'} (t), \forall u, u' \in U.
\]

C. Dynamic VNF Migration

Since the users and nodes are mobile in our proposed system and we consider time-varying traffic, dynamic VNF migration must be applied to existing VNF instances in order to meet the QoS requirement of the newly arrived requests as well as provide continuous service for the existing users with minimum energy consumption. In addition, the proposed dynamic VNF migration must have maximum impact on the QoS of the existing users. We formulate the dynamic VNF migration problem as a time-slotted model, where the position of UAVs and users are fixed at each time slot and change from the current time slot to the next one. Note that a migration decision is made at the beginning of each time slot. We define the binary decision variable \( m_{uk}^j (t) \), with \( m_{uk}^j (t) = 1 \) if VNF \( j \) of service \( O_i \) of user \( k \) assigned to UAV \( u \) at time slot \( t - 1 \) migrates to UAV \( u \) at time slot \( t \), and \( m_{uk}^j (t) = 0 \) otherwise. Note that there is a relationship between \( x_{uk}^j (t - 1) \), \( x_{uk}^j (t) \), and \( m_{uk}^j (t) \), given by \( m_{uk}^j (t) = x_{uk}^j (t - 1) - x_{uk}^j (t) \). The flow conservation constraint also must be satisfied for each migration:

\[
C14: \sum_{u \in U} m_{uuk}^j (t) x_{uuk}^j (t) - \sum_{u \in U} m_{u'uuk}^j (t) x_{u'uuk}^j (t) = \begin{cases} 1, & u = \hat{u}, \\ -1, & u = \hat{u}, \forall i, k \in K, \\ 0, & \text{otherwise}. \end{cases}
\]
be executed in the new node with different computing capacity. Therefore, any VNF migration occurs only if it does not degrade the quality of the corresponding service. The migration delay from UAV \( u' \) to UAV \( u \) for VNF \( j \) of service \( O_i \) of user \( k \) at time slot \( t \) can be formulated as:

\[
\tau_{\text{MG,j}}^{\text{MG,j}}(t) = m_{\text{uuk}}(t) \frac{\rho_{\text{ij}}}{\omega_{\text{uu}}},
\]

where \( \rho_{\text{ij}} \) is information of VNF \( j \) of service \( O_i \) which must be sent from its source to its destination during the migration process. The whole delay can be written as follows:

\[
D(t) = \sum_{n \in N} \sum_{u \in U} \sum_{k \in K_j \in T_j} \left( \tau_{\text{ruuk}}^{\text{PR,j}}(t) + \tau_{\text{ruuk}}^{\text{PD,j}}(t) + \tau_{\text{ruuk}}^{\text{MG,j}}(t) \right).
\]

For each function of service \( O_i \), we suppose that successful migration can happen when the destination VNF node has communication with the source VNF node and has enough resources to process the migrated function. To ensure QoS, the total delay for service \( O_i \) of user \( k \) must be smaller than the delay threshold \( \tau_i \) as follows:

\[
C15: \sum_{n \in N} \sum_{u \in U} \sum_{k \in K_j \in T_i} \left( \tau_{\text{ruuk}}^{\text{PR,j}}(t) + \tau_{\text{ruuk}}^{\text{PD,j}}(t) + \tau_{\text{ruuk}}^{\text{MG,j}}(t) \right) \leq \tau_i.
\]

Therefore, with/without migration delay \( \tau_{\text{MG,j}}^{\text{MG,j}}(t) \), the sum of processing delay \( \tau_{\text{ruuk}}^{\text{PR,j}}(t) \), the propagation delay \( \tau_{\text{ruuk}}^{\text{PD,j}}(t) \), and the transmission delay \( \tau_{\text{ruuk}}^{\text{MG,j}}(t) \) at time slot \( t \) should be smaller than the delay threshold \( \tau_i \).

III. PROBLEM STATEMENT

The optimization problem is formulated as below, in which we jointly maximize EE and minimize the end-to-end services' delay by finding optimal transmit power, subcarrier assignment, UAVs' location, service function chain path, function placement and scheduling subject to network and resource constraints.

Defining EE as the ratio of the network sum-rate over the total consumed power, we can write the EE based on (32), shown at the bottom of the page, where \( P_f(W) \) is the rotodynamic UAV traveling power consumption which is a function of the UAV's velocity \( W \) and includes the blade profile power consumption, the induced power consumption, and the parasite power consumption which is calculated as follows [34], [35], [36]:

\[
P_f(W) = \frac{G^\frac{2}{3}}{p_\eta} \left( 1 + \frac{3W^2}{t^2_{\text{tip}}} \right) + \left( 1 + \nu \right) \left( \frac{G^\frac{2}{3}}{\sqrt{2\pi}p_\eta} \right)
\]

\[
E(t) = \frac{\sum_{u \in U} \sum_{u' \in U} \sum_{v \in V} \sigma_{\text{uuv}} \omega_{\text{uuv}} (1 + \nu)}{\sum_{u \in U} \sum_{u' \in U} \sum_{v \in V} \sigma_{\text{uuv}} \omega_{\text{uuv}} + P_f(W)} + \sum_{u \in U} \sum_{k \in K_u} g_{\text{uku}}(t) r_{\text{uku}}(t) + \sum_{k \in K_u} \sum_{u \in U} \sum_{e \in E} \tilde{g}_{\text{uku}}(t) p_{\text{uku}}(t)
\]

\[
\left( \frac{1 + \frac{W^2}{4v_0^2}}{2\eta} - \frac{W^2}{2v_0^2} \right) + \frac{1}{2} \frac{d_0d_0}{\nu} \Theta W^3,
\]

where \( W_{\text{tip}} \) and \( v_0 \) are the rotor blade speed and mean rotor induced velocity of UAV, respectively. \( \rho \) and \( d_0 \) denote air density and fuselage drag ratio of the UAV respectively. \( \nu \) is incremental correction factor to induced power. \( \nu \) and \( \nu \) represent the rotor solidity and disc area of UAV, respectively. Finally, \( P_0 \) is on behalf of the blade profile power constant and \( P_s \) represents the induced power constant. Accordingly, we formulate our optimization problem as follows:

\[
\min_{p,\tilde{p},\tilde{p},\tilde{q},x,y,M_s \tilde{g},g} \eta = \frac{1}{2} E(t) - \frac{1}{2} D(t),
\]

Subject to: \( C16: \sum_{w \in U} \sum_{u \in L} \alpha_{k}^{O_i}(t) g_{\text{uku}}(t) \leq \alpha_{k}^{O_i}(t)b_i, \)
\( C17: \sum_{w \in U} \sum_{w' \in U} \alpha_{k}^{O_i}(t) g_{\text{uku}}(t) \leq \alpha_{k}^{O_i}(t)b_i, \)
\( C18: \sum_{w \in U} \sum_{w' \in U} \sum_{u \in L} \alpha_{k}^{O_i}(t) g_{\text{uku}}(t) \leq \alpha_{k}^{O_i}(t)b_i, \)
\( C19: \tau_{\text{ruuk}}^{\text{PR,j}}(t) + \tau_{\text{ruuk}}^{\text{PD,j}}(t) + \tau_{\text{ruuk}}^{\text{MG,j}}(t) \leq \tau_i, \)

where \( p = [p_{\text{uuv}}], \tilde{p} = [\tilde{p}_{\text{uku}}(t)], \tilde{p} = [\tilde{p}_{\text{uku}}(t)], \tilde{q} = [\tilde{q}_{\text{uku}}], \tilde{g} = [\tilde{g}_{\text{uku}}(t)], g = [g_{\text{uku}}(t)], \alpha_{k}^{O_i}(t) \) is one if user \( k \) requests service \( O_i \), otherwise is zero. \( \frac{1}{2} \) and \( \frac{1}{2} \) are two coefficients to balance the EE and the average delay. It is worth noting that we considered these coefficients in our optimization problem in order to balance the EE and average delay values. Therefore, both will have approximately equal impact on our optimization problem. Without these balancing coefficients, the solution may lead to improve only the EE since the average delay normally has very small value compared to the EE. The average delay would have very little impact on the optimization problem if it is not weighted enough. The problem is solved at each time slot in two different scenarios. In the first scenario, we consider that the migration can be performed for each VNF of each user while in the second scenario, there is no migration during the time duration of service for each user. For example, consider user \( k^S_i \) requests service \( O_i \) toward user \( k^D_j \) with service duration \( T_i = 10 \) (time slots). According to the first scenario, the VNF migration can happen within the time duration of the service, i.e., 10 time slots, to guarantee the user QoS and minimize
the cost of the network. On the other hand, in the second scenario, there is no VNF migration until the end of the 10th time slot. As shown in Fig. 3(a), if migration is allowed, then the VNF placement may change based on the user’s requirements, user’s mobility, and network cost, and as shown in Fig. 3(b) if migration is not allowed, then the VNF placement remains the same until the 10th time slot.

IV. PROPOSED ALGORITHM

The optimization problem in (34) is non-convex and NP-hard with lots of optimization variables such as UAV’s trajectories, subcarrier assignment, transmit power allocation, VNF placement, and VNF scheduling. It can be formulated as Markov Decision Process (MDP), where UAVs act as agents and the cellular network consisting of mobile users is considered as the environment. Therefore, at each time slot the UAVs act in the environment and serve the user’s demands which cause the environment to transit to another state. Considering our problem as an MDP process, we can solve it using RL based methods. Hence, we develop a DRL method which can support both discrete and continuous optimization variables (actions) in our problem. Note that the non-learning based methods could not find near optimal solution for such optimization problems with high number of optimization variables in a timely manner [9]. One may decompose such problems into multiple sub-problems so as to solve it via non-learning based methods [12].

A. Background of Reinforcement Learning

The basic common RL method is Q-learning which suffers from limitations in low dimensional states and action spaces [37]. To support high dimensional state and action spaces in Q-learning, DQN leverages a deep neural network that acts as a function approximation for Q-learning which can solve high dimensional state spaces. However, DQN can only handle discrete actions [23]. To overcome the high dimensional and continuous action spaces, DDPG, a model-free off-policy RL method [23] could be used. DDPG reinforcement learning is a robust and appropriate solution for an environment in which the states and actions are continuous. However, in some problems like our proposed optimization problem, agents must perform both discrete and continuous actions. In such problems, it’s more efficient to consider methods that can support combinations of discrete and continuous actions [25].

One of the typical approaches to deal with hybrid continuous and discrete actions is to convert the continuous actions into discrete ones in order to solve the problem with DRL methods which only works with discrete actions [25]. In other words, by quantizing continuous actions in action space, we have fully discrete action space so that methods like DQN could solve the problem [24]. Although, there exist methods which consider both continuous and discrete actions like CA2C [26], they needs lots of modifications to be applied to our optimization problem since they only consider single discrete action, while our proposed problem consists of multiple discrete actions. Also, they may not be a good candidate for the partially observable environment since the agent needs full state information of the environment for action selection which is not applicable for our problem.

B. Proposed Hierarchical Hybrid Continuous and Discrete Actions

In this section, we introduce our proposed HHCDA DRL method which can support both continuous and discrete actions simultaneously. More precisely, as shown in Fig. 4, we utilize DDPG and DQN methods hierarchically for supporting the continuous and discrete actions, respectively. As demonstrated, each UAV as an agent consists of 3 different neural networks to interact with the system model (environment). Accordingly, we first explain the architecture of our HHCDA DRL method in single-agent scenario, then we extend it to multi-agent scenario.

1) Single-Agent: Consider an agent (UAV) with an action \( a \in \mathcal{A} \) which is a combination of arrays of continuous and discrete values as \( a^\xi = (a_1^\xi, \ldots, a_\xi^\xi) \) and \( a^\Gamma = (a_1^\Gamma, \ldots, a_\Gamma^\Gamma) \), respectively, where \( \xi \) and \( \Gamma \) denote the number of continuous and discrete actions. Therefore each action \( a \) includes a set of continuous and discrete values, i.e., \( a = (a^\xi, a^\Gamma) \).

a) Action selection mechanism: As depicted in Fig. 4, for the continuous actions, the DDPG (N2) and for the discrete actions, the DQN (N1) are considered. The action selection mechanism of our proposed HHCDA method is performed in two successive steps. At the first step, the continuous actions \( a^\xi \), i.e., UAVs movements and uplink (UL)/downlink (DL)
power allocations, are generated from the actor network in DDPG (N2) using the state of the environment \( s(t) \) as input, then at the second step, the discrete actions, i.e., VNF path selection and subcarrier allocation, \( a^F_i \) are generated from DQN (N1) using the union of continuous actions and the state of the environment as input.

b) Training process mechanism: Let \( \theta, \omega, \) and \( \phi \) denote the parameters for actor, critic, and DQN networks, respectively. In the first step, the agent selects the appropriate actions based on an approximation of a deterministic policy \( \pi \), i.e., \( \pi(a^S(t)|s(t); \theta) \), then the critic estimates the goodness of actions taken by the actor with Q-value function, \( Q(s(t), a^S(t); \omega) = \mathbb{E}_{a^S(t)}[R(t)|s(t), a^S(t)], \) where the expectation taken over the action \( a^S(t) \). Accordingly, at each training step (time slot) \( t \), the critic calculates the loss using as follows:

\[
L(\omega) = \mathbb{E}_{s,a^S,t,s' \sim \mathcal{B}} \left[ (y(t) - Q(s(t), a^S(t); \omega))^2 \right], \quad (35)
\]

where \( y(t) = r(t) + \gamma Q(s(t+1), a^S(t+1)), \omega' \) is the parameter of the target critic network, and \( \mathcal{B} \) is the replay buffer. The aim of N2 in Fig. 4 is to find a policy \( \pi \) that gives the action \( a^S(t) \) that maximizes the Q-function \( Q(s(t), a^S(t)) \). Therefore, by performing the gradient ascent of the loss function with respect to \( \theta \), we aim to solve the following objective:

\[
\max_{\theta} \mathbb{E}_{s \sim \mathcal{B}} [Q_\theta(s(t), a^S(t))]. \quad (36)
\]

After selecting the appropriate continuous actions at time slot \( t \) as \( a^S(t) \), the DQN module (N1 in Fig. 4) instantly generates \( a^S(t) \) at the same time slot \( t \), which is multiple discrete actions, using \( s_{DQN}(t) \) as the input state, where \( s_{DQN}(t) = a^S(t) \cup s(t) \). For generating each discrete action \( a^S_i(t) \in a^S \), \( i \in \{1,...,F\} \) with state \( s_{DQN}(t) \), there exists a Q-value function as \( Q_i(s_{DQN}(t), a^S_i(t)) \) and is defined as follows:

\[
Q_i(s_{DQN}(t), a^S_i(t)) = \mathbb{E}_{r|s = s_{DQN}(t)}[r|s]. \quad (37)
\]

We use a deep Q-network with parameter \( \phi \) to approximate the above Q-value function, therefore for optimizing the Q-value function we calculate the loss for each discrete action \( i \in \{1,...,F\} \) at each time slot \( t \) as follows:

\[
L^i(\phi) = \mathbb{E}_{s_{DQN}^t, a_i, r, s_{DQN}^t+1} \left[ (y^t_i - Q_i(s_{DQN}^t, a_i))^2 \right], \quad \forall i \in F, \quad (38)
\]

where \( y^t_i = r^t + \gamma Q'_i(a^S_{DQN}^{t+1}, \arg \max_{a^S_i(t+1)} Q_i(s_{DQN}^{t+1}, a^S_i(t+1)). \)

The loss is calculated for each discrete action \( i \), therefore the total loss of selecting all action concurrently is calculated by:

\[
L^i(\phi) = \frac{1}{F} \sum_{i \in \{1,...,F\}} L^i(\phi). \quad (39)
\]

For each HHCDCA agent, we can define a tuple as \( (S, \mathcal{A}, \mathcal{R}, \mathcal{S}', \gamma) \), where \( S \) is the agent’s observation from the environment, \( \mathcal{A} \) consists of all possible agent actions, \( \mathcal{S}' \) is the next state, \( \mathcal{R} \) is the reward function and \( \gamma \) is the discount factor which defines how the future reward is important. More precisely, we define the elements of each part of the tuple as follows:

- The state space \( S \) for single-agent: The state of the agent at time slot \( t \) consists of all UAVs and users’ positions as \( \mathbf{q}(t) \) and \( \mathbf{q}(t) \), respectively, as well as the arrived services of all users as \( \mathcal{O}_{\text{Total}} \) and the residual CPU resource of all UAVs as \( \mathcal{C} = \{C_0, ..., C_U\} \) and the residual bandwidth between UAVs as \( \mathcal{W} = \{w_{u0}, w_{u1}, ..., w_{uu}, ..., w_{uu}, \} \). So the state space for the single-agent is defined as \( S = (\mathbf{q}(t), \mathbf{q}(t), \mathcal{O}_{\text{Total}}, C, \mathcal{W}) \).

- The action space \( \mathcal{A} \) for single-agent: The action space consists of arrays of all UAVs movement \( \Delta \mathbf{q}_U \), UAV-UAV power allocation \( p(t) \), UAV-UAV subcarrier allocation \( \mathcal{S}(t) \), UL power allocation \( p(t) \), DL power allocation \( p(t) \), VNF placement \( \mathcal{V}(t) \), VNF relay \( \mathcal{Y}(t) \), UL subcarrier allocation \( \mathcal{g}(t) \), DL subcarrier allocation \( g(t) \), and VNF migration \( M = \{m_{01}^o(t), m_{01}^f(t), ..., m_{uv}^f(t), \} \). So, the action space can be defined as \( \mathcal{A} = (\Delta \mathbf{q}_U, p(t), \mathcal{S}(t), \mathcal{g}(t), \mathcal{g}(t), \mathcal{g}(t), \mathcal{V}(t), \mathcal{Y}(t), \mathcal{g}(t), g(t), M) \).

- The reward function \( \mathcal{R} \). The reward is a numerical value given to the agent at each time slot \( t \) from the environment based on the achieved EE and latency as well as satisfying constraints C1, ..., C19. Therefore, the reward function for our proposed model can be defined as:

\[
\mathcal{R} = q_1 E - q_2 D, \quad \text{C1-C19 are satisfied}. \quad (40)
\]

2) Multi-Agent: In our proposed multi-agent framework, we assume each UAV as an agent has a DDPG and a DQN for individual action selection as depicted in Fig. 4. In other words, each UAV selects the action consisting of its individual movement, transmit power, and subcarrier allocation for UAV-UAV communication as well as transmit power, subcarrier assignment in UL and DL transmission, VNF placement and VNF scheduling for its specific users. These specific users are assigned to each UAV in DL and UL transmissions with (2) and (14), respectively. In our proposed multi-agent framework we consider that UAVs do not have full observation of the environment, they only have information about their corresponding users, their nearby UAVs and UAV-UAV link status. So our multi-agent framework is considered as partially observable to the UAVs. Therefore, our considered multi-agent framework is Decentralized Partially Observable MDP (Dec-POMDP) [38]. Accordingly, we define the state space, action space, and the reward function of each UAV as follows:

- The state space \( S_u \) for each UAV in multi-agent: At time slot \( t \), each UAV \( u \) has partial observation of the environment that consists of the UAV and its assigned users’ positions as \( \mathbf{q}_u(t) \) and \( \mathbf{q}_u(t) \), respectively, as well as the arrived services of its assigned users as \( \mathcal{O}_{\text{Total, } u} \) and the residual CPU resource of the UAV as \( C_u \) and the residual bandwidth between the UAV-UAV communication as \( \mathcal{W} \). Therefore, the state space for each UAV defined as \( S_u = (\mathbf{q}_u(t), \mathbf{q}_u(t), \mathcal{O}_{\text{Total, } u}, C_u, \mathcal{W}) \).

- The action space \( \mathcal{A} \) for each UAV in multi-agent: The action space for each UAV \( u \) consists of the UAV movement \( \Delta \mathbf{q}_u(t) \), the UAV \( u \) to UAVs power allocation \( p_u(t) \) and subcarrier allocation \( \mathcal{g}_u(t) \), UL power allocation \( p_u(t) \), DL power allocation \( p_u(t) \), VNF
Algorithm 1 The Multi-Agent HHCDA Method

Input: Initialize weights of actor, critic, and DQN networks, θ, ω, and φ, with random values.

Set: Put the value of the main networks into the target networks, \( \theta \rightarrow \hat{\theta} \) and \( \phi \rightarrow \hat{\phi} \).

By considering the memory size of: \( D_{\text{Mem}} \), mini-batch memory size \( D_{\text{Mini-Batch}} \), time slot number \( T_d \), and terminate episode number \( E \), perform following steps

for \( e = 1 : E \) do

for \( i = 1 : T_d \) do

Observe state \( s(t) \) of the environment

for \( i \in (\text{total UAVs}) \) do

Observe state \( i \) from the environment

Obtain the continuous actions \( a_i^*(t) \) from \( s_i(t) \) based on the policy \( \pi \) from the actor network

Obtain the discrete actions \( a_i^t(t) \) from \( s_i(t) + a_i^*(t) \) based on the policy \( \pi_i \) from DQN network

Execute the hybrid action \( a_i^t(t) + a_i^*(t) \) for each UAV \( i \) and obtain the team reward \( r(t) \)

Environment transitions to new state \( s(t + 1) \)

end

Store \((s(t), a_i^t(t), r(t), s(t + 1))\) in experience replay \( D \)

end

Randomly sample from a mini-batch of \( D \)

Calculated following equation to update the critic

\[ L(\omega) = \mathbb{E}_D(y(t) - Q(s(t), a^t(t), \omega))^2 \]

Where \( y(t) = r(t) + \gamma Q(s(t + 1), a^*(t + 1), \omega) \)

Calculated the following equation to update the DQN

\[ L(\phi) = \mathbb{E}_D(y^t(t) - Q(s(t), a^t(t), \phi))^2 \]

Where \( y^t(t) = r(t) + \gamma \max \{Q(s(t), a^t(t + 1), \phi)\} \)

Update the main networks using Adam optimizer

Update the weight of target networks by period \( \tau \)

end

1. Place UAVs: Put the value of the main networks into the target networks, \( \theta \rightarrow \hat{\theta} \) and \( \phi \rightarrow \hat{\phi} \).

2. By considering the memory size of: \( D_{\text{Mem}} \), mini-batch memory size \( D_{\text{Mini-Batch}} \), time slot number \( T_d \), and terminate episode number \( E \), perform following steps

for \( e = 1 : E \) do

for \( i = 1 : T_d \) do

Observe \( s(t) \), \( a^t(t) \), and \( r(t) \) in experience replay \( D \)

end

Randomly sample from a mini-batch of \( D \)

Calculated following equation to update the critic

\[ L(\omega) = \mathbb{E}_D(y(t) - Q(s(t), a^t(t), \omega))^2 \]

Where \( y(t) = r(t) + \gamma Q(s(t + 1), a^t(t + 1), \omega) \)

Calculated the following equation to update the DQN

\[ L(\phi) = \mathbb{E}_D(y^t(t) - Q(s(t), a^t(t), \phi))^2 \]

Where \( y^t(t) = r(t) + \gamma \max \{Q(s(t), a^t(t + 1), \phi)\} \)

Update the main networks using Adam optimizer

Update the weight of target networks by period \( \tau \)

\end

end

V. COMPUTATIONAL COMPLEXITY AND CONVERGENCE ANALYSIS

A. COMPUTATIONAL COMPLEXITY AND CONVERGENCE ANALYSIS

- **User Association and Subcarrier Assignment:** For each UAV, the UAVs aim to maximize their team reward which is calculated as \( R \) based on (40).

- **Decision Making:** The computational complexity of the back-propagation algorithm is \( O(\mathcal{K}^3) \).

- **Complexity of the Training Process:** The UAVs should calculate the Q-function values of all users. In accordance with the previous section, this step has \( O(\mathcal{IJKUM}) \) computational complexity where \( M \) is the size of the training batch. Moreover, fully connected neural network with a fixed number of hidden layers and neurons, the complexity of the back-propagation algorithm is related to the product of the input size and the output size.

B. CONVERGENCE ANALYSIS

For the Q-learning algorithm, the Q-function can converge to the optimal Q-function as \( t \rightarrow \infty \) with probability 1, if \( \sum_{t=0}^{\infty} \alpha^t = \infty \) and \( \sum_{t=0}^{\infty} (\alpha^t)^2 < \infty \) are satisfied and \( |r^\omega(s, a^t)| \) is bounded [39]. Since our proposed algorithm is an extended version of the Q-learning algorithm, it can converge to the optimal Q-function as the mentioned conditions are satisfied. For fast convergence and to train our neural network effectively, we utilize the inverse time decaying learning rate that uses the large learning rate in the first episodes in order to prevent the network from getting stuck in a bad local optimum trap near the initial point and uses the small learning rate in the last training epochs in order to converge to a good local optimum [40]. The convergence of our proposed algorithm is also analyzed through simulations in Section VI.

VI. PERFORMANCE EVALUATION

In this section, the performance of our proposed NFV enabled aerial network is analyzed using different DRL methods. Also, the comparison of our proposed HHCDA method with single-agent and multi-agent approaches with the state-of-the-art ones, i.e., DDPG and Multi-Agent DDPG (MADDPG) are provided. For the evaluations, we consider video streaming service function chaining where the UAVs are deployed to serve end to end live video streaming application [5]. As shown in Fig. 5, the virtual servers such as acquirer, forwarder, and streamer server are abstracted on Multicore i.MX 8 NXP UAVs [41]. We assume that each live stream session request is always mapped to a VNF chain containing a streamer, a compressor, a transcoder, and a cache virtual server. The cache server acts as a proxy that receives video clips from a user,
stores them in memory, and sends them to users or other virtual servers. Storage servers speed up session startup time and prevent source server overload, maintain acceptable overall delay, and improve session startup time, which is the quality of experience standard for live streaming. Instead, compressor servers may help reduce video quality upon request. On the other hand, transcoding functions are essential whenever the requested video codec is different from the original codec. Finally, streamer acts as a multiplier for end users. We assume that every UAV is able to instantiate all types of VNF based on its residual resources. In addition, there are mobile users acting as source and destination which provide or receive High Definition (HD) h.265 video, respectively. We consider HD main profile h.265 video with an average data rate equal to 2.5 Mbps [42] which has good quality and needs less communication resources due to low data rate compared to the previous video coding standards like h.264 and MPEG-2. For the environment, We also consider a square-shaped area with the length of 1000 m where there are multiple users acting as content provider and receiver which transmit and receive live video streams, respectively. The users are randomly and uniformly distributed in the area and are moving with Random Walk model at maximum speed of 5m/s [17]. There are 6 UAVs which can move at 10 m/s as maximum speed toward the x, y, and z directions. Each user requests a live video service which is described with the tuple consisting of destination user, types and number of functions, data rate, and the maximum tolerable delay. As described before, each UAV determines the resource allocation and VNF placement of some specific users. With the cooperation of UAVs, our proposed user assignment based on (2) and (14), prevents the starving of receiving data rate among the mobile users. The simulation is performed using TensorFlow.3 Other simulation parameters are provided in Table I.

It is worth noting that the prefix “-ME” at the end of the name of each method means Migration is Enabled in that method and each method that doesn’t have “-ME” prefix is simulated without VNF migration. The trajectories of the UAVs for one episode after training are depicted in Fig. 6. The starting position of each UAV is marked by a sphere. As depicted in Fig. 6(a), the UAVs movements with HHCDCA-ME are less than with MAHHCDA-ME. This result is due to centralized controlling which has all UAVs information and hence, has better performance, but with the cost of more signaling overhead compared to MAHHCDA-ME. Another result is that considering VNF migration for VNF placement, the UAVs traveled distances are less compared to those not considering VNF migration which significantly decreases the energy consumption. This is expected since with VNF migration the UAVs do not need to move to a better position to satisfy the constraints. In order to show the performance gain of considering NFV-enabled network, we compare our proposed NFV-enabled network to a not NFV-enabled networks. As can be seen from Fig. 6(b) and Fig. 6(d) we can see that using NFV, the UAVs movement decreases significantly. Finally, we compare our proposed MAHHCDA DRL method with MADDPG to show the superiority of our proposed method. As demonstrated, the UAVs traveled distance decreases significantly when applying our proposed MAHHCDA DRL method compared to the MADDPG method. For the convergence analysis, the accumulated rewards (pallid curves) with their average (solid curves) for single-agent and multi-agent

3The source code and implementations are provided in https://ieee-dataport.org/documents/ai-based-and-mobility-aware-energy-efficient-resource-allocation-and-trajectory-design-nfv.
HHCDA with migration (HHCDA-ME and MAHHCDA-ME) compared to the DDPG method are shown in Fig. 6(f). As depicted, our proposed HHCDA-ME (single-agent HHCDA with VNF migration) requires fewer training steps than other methods. On the other hand, MAHHCDA-ME converges to a lower reward value with higher training steps compared with HHCDA-ME, because in our proposed MAHHCDA-ME, the agents have partial observation from the environment. Whereas the HHCDA-ME has whole information about the environment, and accordingly, it outperforms the MAHHCDA-ME. This is true for MADDPG and DDPG as well. Although two single-agent methods perform better than multi-agent ones, they have very high computational complexity and signaling overhead. It is worth mentioning that we can improve multi-agent performance, even more than single-agent methods, by considering full communication between agents where all agents have full observation from the environment [47]. Fig. 7(a) demonstrates the average end-to-end delay versus the mobility of users. As can be seen, by increasing the mobility of users from 0 to 9 km/h, the average delay increases. Our proposed HHCDA-ME has the lowest end-to-end average delay compared to all DRL methods. Considering HHCDA-ME and MAHHCDA-ME, it is concluded that the performance gain of the single-agent method compared with multi-agent ones is 12.6%. On the other hand, considering the HHCDA-ME and HHCDA, it is concluded that the performance gain of VNF migration is 28.1%. Similarly, comparing MAHHCDA-ME with MAHHCDA, the performance

| Parameter | Description | Value |
|-----------|-------------|-------|
| $V_{\text{max}}$ | User mobility (Random Walk) | 0/2/3/6/9 km/h |
| K | Number of users | 1/4/5/10 |
| $R_{\text{min}}$ | Minimum required rate for each service | 2.5 Mbps |
| U | Number of UAVs | 6 |
| $H$ | UAVs altitude | 75-150 m [43] |
| $w_{\text{max}}$ | UAVs speed | 10 m/s |
| | Maximum UAV CPU capacity | 1 GHz (32 bits) |
| | Maximum UAV MEMORY capacity | 2 GB |
| $\beta_1, \beta_2$ | Environment parameters | 0.56, 0.21 |
| $f_c$ | Carrier frequency | 2 GHz |
| $\rho_{\text{uk}}$ | UAV to user maximum transmit power | 30 dBm [44] |
| $\rho_{\text{ku}}$ | User to UAV maximum transmit power | 23 dBm [44] |
| $\rho_{\text{ku}}$ | UAV to UAV maximum transmit power | 30 dBm [44] |
| $\tau$ | Time slot interval | 0.5 ms |
| $B, B$ | Bandwidth for UL and DL transmission | 10 MHz |
| B | Bandwidth for UAV to UAV transmission | 20 MHz |
| | Number of sub-channels at each time slot for UL and DL | 2 |
| | Number and bandwidth of sub-carriers for UAV to UAV communication | 256 and 78 KHz |
| $N_0$ | Noise power spectral | -170 dBm/Hz |
| $\kappa$ | Path loss exponent | 3.5 |
| $\alpha$ | Initial learning rate | 0.001 |
| $\epsilon$ | Epsilon decay | 0.01 |
| $\gamma$ | Discount factor | 0.99 |
| | Activation function for input and layers | ReLU [45] |
| | Activation function for output layer | tanh [45] |
| | Number of episodes | 3000 |
| | Number of iterations | 300000 |
| | Number of hidden layers | 3 |
| | Target network update frequency | 1000 [45] |
| M | Batch size | 128 |
| | Number of neurons in each layer | 512 |

**UAV power consumption parameters [46]**

| Parameter | Description | Value |
|-----------|-------------|-------|
| $\hat{\rho}$ | Air density | 1.225 km/m$^3$ |
| $\Theta$ | Rotor disc area | 0.503 m$^3$ |
| $U_{\text{tip}}$ | Tip speed for the rotor blade | 120 m/s |
| $d$ | Rotor solidity | 0.05 |
| $d_{\text{drag}}$ | Fuselage drag ratio | 0.6 |
| $\nu_0$ | Mean rotor induced velocity | 4.03 |
| $\Omega$ | Blade angular velocity | 300 radians/second |
| $\delta$ | Profile drag coefficient | 0.012 |
| $R$ | Rotor radius | 0.4 m |
| n | Incremental correction factor to induced power | 0.1 |
| G | UAV weight in Newton | 20 |
gain is 25%. We also analyze the performance gain of considering NFV in our system model. For this purpose, we applied our proposed single-agent and multi-agent HHHCDA methods in a system in which each UAV can only run a specific physical function (the methods with “-no-NFV” prefix). As demonstrated, our proposed HHHCDA and MAHHHCDA have 32.2% and 31.5% performance gain compared to HHHCDA-no-NFV and MAHHHCDA-no-NFV, respectively. Finally, we compare our proposed HHHCDA DRL method with DDPG, the well-known and yet powerful continuous action DRL method. Results show that our proposed HHHCDA-ME and MAHHHCDA-ME have 29.1% and 25.5% performance gain compared to DDPG-ME and MADDPG-ME, respectively.

Fig. 7(b) depicts the EE defined with (32) versus the mobility of users. As the mobility of the users increases, the EE decreases significantly. This result is expected since the UAVs should move more to satisfy the users’ requirements resulting in consuming more power. As shown, our proposed HHHCDA-ME obtained the maximum EE with all users’ mobility. Considering the migration, our proposed HHHCDA-ME method has a 15.5% performance gain compared to MAHHHCDA-ME method. Moreover, our proposed HHHCDA-ME and MAHHHCDA-ME increase the EE by 35.5% and 45.1%, respectively. The other result shows that our proposed HHHCDA-ME and MAHHHCDA-ME methods outperform DDPG-ME and MADDPG-ME by 50.5% and 38%, respectively. In addition, considering NFV enabled networks resulted in a 152% and 143% performance gain for single-agent and multi-agent methods, respectively. Fig. 7(c) shows the total energy consumption versus the total required data rate of the users. Note the values of the horizontal axis corresponds to the total required data rate by all users. Since each user requests $2.5\text{Mbps}$ of data rate, therefore, $18 \times 2.5\text{Mbps}$, $18 \times 2.5\text{Mbps}$, $18 \times 2.5\text{Mbps}$, and $36 \times 2.5\text{Mbps}$, respectively. As can be seen, with increasing the required data rate by users, the energy consumption increases. However, our proposed HHHCDA-ME and MAHHHCDA-ME outperform other methods. More precisely, the HHHCDA-ME decreases the energy consumption by 33% compared to DDPG-ME. Also, our proposed MAHHHCDA-ME decreases the energy consumption by 25.6% compared to the MADDPG-ME method. On the other hand, considering migration in our HHHCDA method decreases the energy consumption by 33%. Fig. 8(a) shows the average end-to-end delay versus the number of users. As the number of users increases, the average end-to-end delay increases. As depicted, our proposed HHHCDA-ME achieves the lowest average end-to-end delay. On the other hand, considering NFV when applying our proposed HHHCDA DRL method, we obtain 55% and 59% lower average end-to-end delay in single-agent and multi-agent methods, respectively. In addition, when we consider migration, the average end-to-end delay decreases by 12.5% and 11.4% in single-agent and multi-agent methods, respectively. Finally, we can see that our proposed HHHCDA method decreases
average end-to-end delay compared to the DDG method by 19.2% and 15% in single-agent and multi-agent approaches, respectively. Fig. 8(b) shows the EE versus the number of users. As shown, our proposed HHCD-ME achieves the highest EE. Considering NFV enabled UAVs without VN migration, our proposed HHCD and MAHHCD obtain 60.5% and 77% higher EE compared to HHCD-no-NFV and MAHHCD-no-NFV, respectively. In addition, when we consider migration, our proposed HHCD-ME and MAHHCD-ME increase the EE by 39.5% and 26.2% compared to HHCD and MAHHCD, respectively. Finally, we can see that our proposed HHCD method increases EE compared to the DDG method by 36% and 44.8% in single-agent and multi-agent, respectively. The performance gain related to the EE and average delay of all DRL methods are summarized in Table II. These performance gain values are obtained when considering 24 users with 9 km/h as their mobility. Note that the multi-agent scheme of our proposed HHCD method has less complexity because the size of state space and action space are smaller compared to single-agent ones. In other words, in the multi-agent method the state space and action space are divided into multiple smaller state and action spaces between agents, therefore the computational complexity is decreased.

### VII. Conclusion

In this paper, we investigated a dynamic resource allocation, UAV trajectory design, VNF placement, and scheduling framework for an UAV-assisted network to support heterogeneous services with different QoS requirements. We proposed a hybrid action DRL-based framework, called HHCD, to facilitate energy efficient trajectory design, resource allocation and network management. Simulation results showed that our proposed migration enabled-scheme increase the EE by 36.6% and 28.3% in single-agent and multi-agent methods compared to the state-of-the-art schemes and also provide a good balance between EE and average delay in an environment with high mobility users, especially when the number of users is large in the network. Finally, we analyzed the performance gain of considering NFV enabled network. Results show that when considering NFV, the EE increased by 63% compared to not considering NFV enabled UAVs. However, massive MIMO is a good candidate for coverage extension in cellular-connected if efficiently deployed specially in UAV assisted networks. With this technique we may be able to enhance the capacity of our proposed system. Therefore, in our future work we aim to consider multi-antenna setup for UAV assisted NFV enabled deployment.

### References

[1] W. Saad, M. Bennis, and M. Chen, “A vision of 6G wireless systems: Applications, trends, technologies, and open research problems,” *IEEE Netw.*, vol. 34, no. 3, pp. 134–142, May/Jun. 2020.

[2] A. Laghrissi and T. Taleb, “A survey on the placement of virtual resources and virtual network functions,” *IEEE Commun. Surveys Tuts.*, vol. 21, no. 2, pp. 1409–1434, 2nd Quart., 2019.

[3] J. G. Herrera and J. F. Botero, “Resource allocation in NFV: A comprehensive survey,” *IEEE Trans. Netw. Service Manag.*, vol. 13, no. 3, pp. 518–532, Sep. 2016.

[4] A. Azizi, S. Parsaeeafard, M. R. Javan, N. Mokari, and H. Yankomeroglu, “Profit Maximization in 5G+ networks with heterogeneous aerial and ground base stations,” *IEEE Trans. Mobile Comput.*, vol. 19, no. 10, pp. 2445–2460, Oct. 2020.

[5] J. F. C. Moreno, R. Sattler, R. P. C. Cisterna, L. R. Celsi, A. S. Rodríguez, and M. Mecella, “Online service function chain deployment for live-streaming in virtualized content delivery networks: A deep reinforcement learning approach,” *Future Internet*, vol. 13, no. 11, p. 278, 2021.

[6] F. Hu, Y. Deng, W. Saad, M. Bennis, and A. H. Aghvami, “Cellular-connected wireless virtual reality: Requirements, challenges, and solutions,” *IEEE Commun. Mag.*, vol. 58, no. 5, pp. 105–111, May 2020.

[7] X. Diao, J. Zheng, Y. Wu, Y. Cai, and A. Anpalagan, “Joint trajectory design, task data, and computing resource allocations for NOMA-based and UAV-assisted mobile edge computing,” *IEEE Access*, vol. 7, pp. 117448–117459, 2019.

[8] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep Learning*, Cambridge, MA, USA: MIT Press, 2016.

[9] N. C. Luong et al., “Applications of deep reinforcement learning in communications and networking: A survey,” *IEEE Commun. Surveys Tuts.*, vol. 21, no. 4, pp. 3133–3174, 4th Quart., 2019.

[10] O. Alhussein et al., “A virtual network customization framework for multicast services in NFV-enabled core networks,” *IEEE J. Sel. Areas Commun.*, vol. 38, no. 6, pp. 1025–1039, Jun. 2020.

[11] S. Agarwal, F. Malandrino, C.-F. Chiasserini, and S. De, “Joint VNF placement and CPU allocation in 5G,” in *Proc. IEEE Conf. Comput. Commun.*, Honolulu, HI, USA, Apr. 2018, pp. 1943–1951.

[12] L. Qu, C. Assi, and K. Shaban, “Delay-aware scheduling and resource optimization with network function virtualization,” *IEEE Trans. Commun.*, vol. 64, no. 9, pp. 3746–3758, Sep. 2016.

[13] N. Gholiipoor, H. Saeedi, N. Mokari, and E. A. Jorswieck, “E2E QoS guarantee for the tactile Internet via joint NFV and radio resource allocation,” *IEEE Trans. Netw. Service Manag.*, vol. 17, no. 3, pp. 1788–1804, Sep. 2020.

[14] Y.-T. Chen and W. Liao, “Mobility-aware service function chaining in 5G wireless networks with mobile edge computing,” in *Proc. IEEE Int. Conf. Commun. (ICC)*, Shanghai, China, May 2019, pp. 1–6.

[15] M. Akbari, M. R. Abedi, R. Joda, M. Pourghasemian, N. Mokari, and M. Erol-Kantarci, “Age of information aware VNFN scheduling in industrial IoT using deep reinforcement learning,” *IEEE J. Sel. Areas Commun.*, vol. 39, no. 8, pp. 2487–2500, Aug. 2021.

[16] T. Mahboob, Y. R. Jung, and M. Y. Chung, “Dynamic VNF placement to manage user traffic flow in software-defined wireless networks,” *J. Netw. Syst. Manag.*, vol. 8, pp. 436–456, Jun. 2020.

[17] I. Tang, X. He, P. Zhao, G. Zhao, Y. Zhou, and Q. Chen, “Virtual network function migration based on dynamic resource requirements prediction,” *IEEE Access*, vol. 7, pp. 112348–112362, 2019.
Mohammad Reza Javan (Senior Member, IEEE) received the B.Sc. degree in electrical engineering from Shahid Beheshti University, Tehran, Iran, in 2003, the M.Sc. degree in electrical engineering from the Sharif University of Technology, Tehran, in 2006, and the Ph.D. degree in electrical engineering from Tarbiat Modares University, Tehran, in 2013. He is currently a Faculty Member with the Department of Electrical Engineering, Shahrood University of Technology, Shahrood, Iran. His research interests include design and analysis of wireless communication networks with emphasis on the application of optimization theory, and machine learning methods.

Nader Mokari (Senior Member, IEEE) received the Ph.D. degree in electrical Engineering from Tarbiat Modares University, Tehran, Iran, in 2014. He joined the Department of Electrical and Computer Engineering, Tarbiat Modares University as an Assistant Professor in October 2015. He is currently an Associated Professor with the Department of Electrical and Computer Engineering, Tarbiat Modares University. In recent years, his research has been funded by Iranian Mobile Telecommunication Companies, Iranian National Science Foundation. His research interests cover many aspects of wireless technologies with a special emphasis on wireless networks. His thesis received the IEEE outstanding Ph.D. thesis Award. He received the Best Paper Award at ITU K-2020. He has been elected as an IEEE exemplary reviewer in 2016 by IEEE Communications Society. He is on the editorial board of the IEEE TRANSACTIONS ON COMMUNICATIONS. He was also involved in a number of large scale network design and consulting projects in the telecom industry.

Eduard A. Jorswieck (Fellow, IEEE) is a Managing Director of the Institute of Communications Technology and the Head of the Chair for Communications Systems and a Full Professor with Technische Universitaet Braunschweig, Brunswick, Germany. From 2008 until 2019, he was the Head of the Chair of Communications Theory and a Full Professor with TU Dresden, Germany. He has published some 150 journal papers, 15 book chapters, 3 monographs, and 300 conference papers on these topics. His main research interests are in the broad area of communications. In 2006, he received the IEEE Signal Processing Society Best Paper Award. Since 2017, he has been serving as the Editor-in-Chief of the EURASIP Journal on Wireless Communications and Networking. He has served on the editorial boards for IEEE TRANSACTIONS ON SIGNAL PROCESSING, IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, IEEE SIGNAL PROCESSING LETTERS, and IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY.