User-Centric Satellite Handover for Multiple Traffic Profiles Using Deep Q-Learning

NOUR BADINI, Member, IEEE
MONA JABER, Senior Member, IEEE
MARIO MARCHESE, Senior Member, IEEE
FABIO PATRONE, Member, IEEE
University of Genoa, Genoa, Italy
Queen Mary University of London, London, U.K.

Multiple low Earth orbit (LEO) satellites have recently been launched in constellations to ensure direct Internet access to users anywhere and at any time. Due to the high-speed mobility of LEO satellites, users undergo multiple handovers (HOs) during their service time, which has a negative impact on users' quality of service (QoS) if occurred in high frequency. Moreover, next-generation communication technologies are designed to support a wide spectrum of applications, including artificial intelligence, virtual reality, and Internet of Things. Thus, differentiating user equipments (UEs) with different and varying traffic profiles (TP) has become necessary due to each application's unique performance requirements. However, LEO satellites have limited onboard resources and the launched constellations ensure that each UE will be covered by more than one LEO satellite at any given moment, making it challenging to select the optimal satellite at any given time to assure the optimum QoS. Therefore, a satellite HO strategy has to effectively use the few available satellite resources and prevent network congestion while respecting the various resource requirements per TP. To address all the above requirements, we propose a user-centric multiagent deep Q-network satellite HO strategy, which is the first in the state of the art to address the variety and diversity of UEs' performance requirements and generated traffic statistics. Our method showcases a significant achievement of approximately 60% reduction in HO rate and around 91% reduction in blocking rate compared to conventional single-criterion approaches.

Manuscript received 28 June 2023; revised 6 October 2023 and 24 February 2024; accepted 19 July 2024. Date of publication 29 July 2024; date of current version 6 December 2024.

DOI. No. 10.1109/TAES.2024.3434771

Refereeing of this contribution was handled by V. Weerackody.

Authors' addresses: Nour Badini, Mario Marchese, and Fabio Patrone are with the Department of Electrical, Electronics and Telecommunications Engineering, and Naval Architecture (DITEN), University of Genoa, 16145, Genoa, Italy, E-mail: (nour.badini@edu.unige.it, mario.marchese@unige.it, f.patrone@edu.unige.it); Mona Jaber is with the School of Electronic Engineering and Computer Science, Queen Mary University of London, E1 4NS, London, U.K., E-mail: (m.jaber@qmul.ac.uk). *(Corresponding author: Nour Badini.)*

I. INTRODUCTION

Satellite communications can help next-generation communication technologies extend their reach to where terrestrial networks are incapable of providing Internet access, such as in remote areas or on high-speed vehicles (e.g., airplanes and high-speed trains) [1]. This makes them the most efficient way to reliably link the world's neglected, hard-to-reach, and poorly served places. The integration of satellite communications with terrestrial networks has attracted the interest of many researchers [2], [3], [4], [5], [6], [7], especially low Earth orbit (LEO) satellites, whose altitudes range between 200 and 2000 km, due to their low propagation delay, suppressed signaling attenuation, low power to transmit, and low operational costs for satellite deployment and maintenance compared to other satellites with different altitudes [8].

LEO satellites, on the other hand, orbit Earth quickly, with a consequent limited window of visibility between each satellite and a ground user equipment (UE). This results in frequent handover (HO) to guarantee stable communications while satellites rapidly change their coverage areas [9]. This aspect led to the design and deployment of ultra-dense constellations, such as the Starlink project, to cover broad areas of the planet concurrently [10]. They generally guarantee that each ground UE will be covered by more than one Low Earth Orbit (LEO) satellite at each instant, raising the challenge of selecting the best satellite at each instance that guarantees the best quality of service (QoS) per UE.

To address the above challenge, we propose a multiobjective satellite HO strategy that takes into consideration different factors, such as the elevation angle, the remaining visibility time (RVT), and the number of available channels per satellite, and helps each UE to select the best candidate from its covering satellite list to guarantee a good QoS throughout the whole communication duration.

Furthermore, in the absence of a centralized controller, UEs may only get a limited amount of information about the satellite system while competing for the limited resources per satellite. This necessitates the implementation of a distributed (user-centric) satellite HO strategy on the ground users where each user can take actions independently, according to its own view of the system.

On the other hand, next-generation communication technologies are intended to support the unprecedented diversity of various emerging applications, such as artificial intelligence, virtual reality, 3-D media, and the Internet of Things (IoT), which have led to distinguishing UEs with different and varying traffic profiles (TPs), i.e., different performance requirements and generated traffic statistics, from the network resource viewpoint. This requires the implementation of a satellite HO strategy that respects the

varied resource requirements per TP to efficiently use the limited available resources and avoid network congestion.

The primary objectives of our proposed HO criteria are to reduce the number of HOs and minimize the blocking rate by balancing the load among satellites taking into account different and varying resource requirements per UE. These objectives are particularly critical in scenarios involving LEO satellite networks, where frequent HOs can lead to a significant signaling overhead, resource consumption, and UE dissatisfaction due to interruptions [11], [12], [13]. The main contributions in this article can be summarized as follows.

1) A novel multiagent deep Q-learning (MADQL)-based HO optimization approach that takes into account the variation and diversity of performance requirements of different UE classes. It allows more efficiently managing the limited available satellite resources and achieves low blocking rate per user.
2) The implementation of the proposed method to access HO decisions in a satellite–terrestrial integrated network (STIN) scenario created within the Network Simulator 3 (NS-3)-based STIN simulator to test it in a closer to the real-world testbed.
3) A sensitivity analysis in which we validate the proposed method under different traffic scenarios and discuss the QoS gain in comparison with state-of-the-art methods.

The rest of this article is organized as follows. Section II provides an overview of the related works addressing satellite HOs. The problem formulation is presented in Section III. Section IV describes the methodology followed for the multiagent reinforcement learning (MARL) and MADQL-based HO optimization strategies. Simulation results are presented and discussed in Section V. Finally, Section VI concludes this article.

II. RELATED WORKS

Several satellite HO strategies have been proposed in the literature following different criteria based on different parameters including the RVT, shortest distance, received signal strength, number of available channels, and load distribution among satellites. For instance, in [14] and [15], the number of available channels per satellite was regarded as the fundamental HO criterion. To achieve minimal drop blocking and enforced termination chances, Karapantazis and Pavlidou [14] separated the multimedia traffic into two groups and handled the satellite HO requests according to the queue condition of each traffic type. On the other hand, in order to prevent resource reservations, Papapetrou and Pavlidou [15] suggested a dynamic Doppler-based HO prioritizing approach that utilizes Doppler shift monitoring to estimate the number of HO demands along with the actual occurrence time. The HO criterion adopted in the aforementioned papers can establish a balanced load in the system, but cannot ensure high communication quality since it may lead to a severe number of HO events with consequent unstable and often interrupted communications. The highest RVT is considered as the main criterion for satellite selection in [16]. The RVT reflects the availability of a satellite for communication at a given time, in addition to how long the satellite will stay in the line of sight of the UE. This criterion significantly reduces the number of HO occurrences, and thus reduces interruptions, at the cost of a high blocking rate. Juan et al. [17] introduced an antenna gain-based HO strategy that takes advantage of the predictability of satellite movement and the antenna gain of satellite beams to reduce service failures and unwanted HO events. An intersatellite HO approach based on potential game theory is presented in [18] for the purpose of lowering the average number of HO occurrences and balancing the constellation network load. An HO control strategy based on the received signal strength is suggested in [19]. However, multiple UEs could connect to the satellite that has the best-received signal strength, which can cause access congestion on that satellite and result in extreme load imbalance among the satellites. Juan et al. [20] performed a mobility performance study of the Release-16 conditional HO, which reduces the radio link and HO failures but numerously increases unnecessary HOs rate.

The mentioned literature papers only analyze a single HO criterion for a given optimization aim, making it difficult to propose a complete and satisfactory solution. Therefore, many studies have emphasized the use of reinforcement learning (RL) multicriteria decision-making processes to reach an overall satellite selection solution. For example, He et al. [21] presented a load-aware MARL HO approach that intends to limit the number of HOs while taking into consideration the load of the satellite. They considered two HO criteria, which are the minimum elevation angle and the currently available satellite channels, and have been able to achieve a lower blocking rate compared to load-unaware systems. Xu et al. [22] adopted an RL strategy that takes into account the service time, communication channel resources, and the relay overhead for the HO events execution in order to maximize the UE's quality of experience. Wang et al. [23] proposed an RL satellite HO scheme that aims to reduce the number of satellite HOs while minimizing the HO-failure rate by taking into consideration the carrier-to-noise ratio and interference-to-noise ratio criteria.

Most recent studies either evaluate one HO criterion for a given optimization objective or propose a method that considers numerous criteria from the perspective of a single UE only. Nonetheless, in the absence of a central controller, UEs may only obtain limited information about the satellite system in relation to themselves. In addition, due to the limited satellite channel budget, competition for available channels between UEs served by the same satellite may potentially lead to a severely unbalanced satellite load. This mandates the adoption of a decentralized (user-centric) satellite HO method that considers the UE's real-time resource competition. For example, He et al. [21] used multiagent RL and the authors in [24] and [25] used multiagent deep-RL to tackle the decentralization challenge by treating each user as an agent with a partial perspective of the system

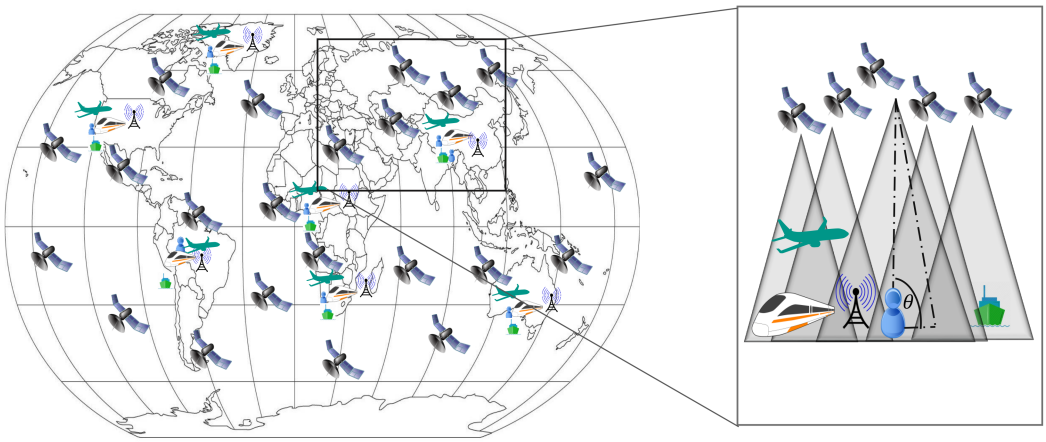

Fig. 1. LEO satellite communication scenario: (Left-hand side) A constellation of LEO satellites are distributed around the Earth's surface to cover the whole Earth. (Right-hand side) A subarea of the Earth where a number of users are located and covered by a number of LEO satellites.

and the ability to take actions independently. However, these approaches do not encourage load balancing among satellites, as they do not provide preference for connecting to the satellite with more available channels, increasing the likelihood of future UE blockage. In contrast, the distributed MARL approach proposed in [26] is a load-balancing HO technique that was successful in lowering the blockage rate and minimizing the number of HOs.

Moreover, to the best of the authors' knowledge, there is no study that takes into consideration the diversity of UEs applications where each user has different and varying resource requirements. This makes the available studies limited to one type of application while next-generation communication technologies are intended to support the unprecedented diversity of various emerging applications, which is the main innovation aspect introduced in this article.

III. PROBLEM FORMULATION

The implemented scenario consists of two main components: a group of K (UEs and a set of N LEO satellites acting as base stations. This architectural configuration is commonly referred to as "regenerative satellite-based NG-RAN" in the 3rd Generation Partnership Project (3GPP) Technical Report 38.821 [27]. We denote the sets of UEs and LEO satellite base stations as $\mathcal{K} = 1, 2, \ldots, K$ and $\mathcal{N} = 1, 2, \ldots, N$, respectively. The UEs represent ground nodes able to get direct access to the network through LEO satellites. At each instant, each UE is generally covered by more than one LEO satellite, as illustrated in Fig. 1. This work focuses mainly on hard HO scenarios, where a HO decision may be triggered due to the satellite movement or to secure more resources to a UE. However, given the hard nature of the HO, frequent interruptions would negatively impact the effective throughput perceived by the UE.

Thus, the optimization problem this study seeks to address is which satellite, at each instant, is the optimal candidate for each UE to allow offering continuous connectivity and ensure the required QoS, by minimizing the number of HOs and the average blocking rate.

The channel model for nonterrestrial networks has been standardized in the 3GPP Technical Report TR 38.811 [28], where the path loss for a user–satellite link is given by the following:

$$\mathrm{PL} = \mathrm{FSPL}(d, f_c) + \mathrm{SF} + \mathrm{CL}(\theta, f_c) + \mathrm{AL}. \quad (1)$$

The free space path loss (FSPL) depends on the distance (d) between the communication endpoints and the carrier frequency (f_c).

Shadowing (SF) represents signal attenuation due to physical obstacles in the propagation environment and is modeled as a log-normal random variable with zero mean and variance σ^2_{SF}. This variance is influenced by the elevation angle, scenario type, and frequency, which are all used to locate corresponding values in a table provided in [28]. The elevation angle is similarly used to determine the clutter (CL) and atmospheric (AL) losses that account for signal attenuation caused by reflection or scattering from objects and attenuation due to atmospheric absorption, respectively.

Thus, according to the 3GPP specifications, the elevation angle has a significant impact on determining the overall link budget, where higher elevation angles typically result in improved link quality, attributed to reduced atmospheric attenuation and fewer obstructions, thereby ensuring a more reliable communication link. Hence, enforcing a minimum elevation angle constraint can ensure a certain level of communication quality, vital for uninterrupted connectivity, which serves as a practical threshold for ensuring minimum acceptable link quality without delving extensively into detailed link budget analysis.

We consider a user k (UE_k) within the coverage area of a satellite n (SAT_n) only if the elevation angle $\theta_{k,n}$ between them illustrated in Fig. 2 is greater than or equal to a minimum threshold θ_0 to guarantee minimum acceptable link quality and avoid link disruptions due to possible physical obstacles. $\theta_{k,n}$ can be determined by using the user's and

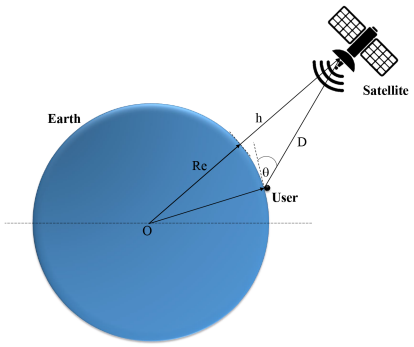

Fig. 2. Geometric representation of the elevation angle θ between a satellite and a ground user.

satellite's position information as follows:

$$\theta_{k,n} = \arcsin\left(\frac{h_n^2 + 2 \times R_e \times h_n - D_{k,n}^2}{2 \times R_e \times D_{k,n}}\right) \tag{2}$$

where h_n is the altitude of SAT_n above the Earths' surface, R_e is the radius of the Earth, and $D_{k,n}$ is the distance between UE_k and SAT_n.

At time t, for every UE_k, all the satellites that satisfy the above elevation angle constraint are considered as covering satellites for UE_k and can be distinguished by the covering indicator $C_{k,n}^t$ as follows:

$$C_{k,n}^t = \begin{cases} 1 & \text{if } \text{UE}_k \text{ is covered by } \text{SAT}_n \text{ at time } t \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

We define $\mathcal{N}_k^t = \{1, 2, \ldots, N_k\} \forall k \in \mathcal{K}$, as the subsets of $\mathcal{N}$ containing the satellites in visibility for user k, i.e., that can be considered as valid candidates for possible user k's HO, at time t. We also define $X_{k,n}^t$ to indicate if UE_k is connected to SAT_n at time t as follows:

$$X_{k,n}^t = \begin{cases} 1 & \text{if } \text{UE}_k \text{ is connected to } \text{SAT}_n \text{ at time } t \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

assuming that each UE is always connected to only one satellite at a time, i.e., $\sum_{n \in \mathcal{N}} X_{k,n}^t = 1 \forall k \in \mathcal{K}$.

Moreover, in next-generation communication technologies, the concept of the bandwidth parts (BWPs) is introduced to facilitate the management and allocation of different portions of the available spectrum for specific purposes. The available spectrum can be subdivided into smaller segments known as BWPs of equal or variable bandwidths, depending on the network configuration and deployment scenario, allowing for flexible resource allocation based on factors, such as UE TP and QoS needs, as shown in Fig 3. Users may be allocated multiple BWPs to accommodate high traffic demands or specific QoS requirements, with the allocation being dynamically adjusted based on changing network conditions. This dynamic and adaptive allocation of BWPs optimizes network resource utilization and enhances user experience by customizing resource allocation to individual TPs and service needs. Overall, BWPs facilitate efficient spectrum utilization and enable the network to meet the diverse needs of users and applications in a dynamic and adaptive manner [29] [30].

In order to account for users with differing TPs, we assume that each UE changes randomly its corresponding resource requirements after each communication duration T. For the sake of simplicity, we distinguish between four TPs (four applications) defined in the 3GPP Technical specification "Service requirements for the 5G system" of Release 17 [31] that differ in their data rate requirements and delay tolerance as follows.

1) TP1 Vehicular connectivity: $R_k = 2W$, s.t. $\hat{R}_k \leq R_k$.
2) TP2 Airplane connectivity (inflight Internet): $R_k = 2W$, s.t. $\hat{R}_k = R_k$.
3) TP3 Narrow-band IoT: $R_k = W$, s.t. $\hat{R}_k \leq R_k$.
4) TP4 Public safety/emergency response: $R_k = W$, s.t. $\hat{R}_k = R_k$.

R_k and $\hat{R}_k$ refer to the target (required) resources and the minimum acceptable resources for UE_k, respectively. We assume that delay-tolerant users (TP1 and TP3) are more lenient in receiving fewer resources than their resource requirements ($R_k \geq \hat{R}_k$), whereas delay-sensitive users (TP2 and TP4) are more rigid and follow the concept of either all or none ($R_k = \hat{R}_k$). As a consequence, delay-sensitive users will be considered blocked if they do not have all of their required resources at time t, while delay-tolerant users will not.

We assume that the complete bandwidth of each satellite is divided into L_n BWP channels of bandwidth W and each user can transmit or receive on one or more of those channels depending on their current TP. The channel budget restriction is therefore provided by

$$l_n^t \leq L_n \qquad \forall n \in \mathcal{N} \tag{5}$$

where

$$l_n^t = \sum_{k \in \mathcal{K}} l_{k,n}^t \cdot X_{k,n}^t. \tag{6}$$

$l_{k,n}^t$ represents the resources of SAT_n allocated to UE_k at time t, such that $\hat{R}_k^t \leq l_{k,n}^t \leq R_k^t$, and $X_{k,n}^t = [0, 1]$ allows to consider only users connected to SAT_n at time t.

UE_k is considered blocked at time t if it tries to connect to a satellite with not enough resources. It is represented by BN_k^t as follows:

$$BN_k^t = \begin{cases} 1 & \text{if } \text{UE}_k \text{ chooses to connect to } \text{SAT}_n \\ & \text{and } l_n^t + \hat{R}_k^t > L_n \\ 0 & \text{otherwise.} \end{cases} \tag{7}$$

Finally, the considered optimization problem can be stated as follows:

$$\min_{X_{k,n}^t} \text{HO}_{\text{avg}} = \frac{\sum_{k \in \mathcal{K}} \text{HO}_k}{K} \qquad \forall k \in \mathcal{K} \tag{8a}$$

$$\text{s.t. } \theta_{k,n} \geq \theta_0 \qquad \forall n \in \mathcal{N}_k \quad \forall k \in \mathcal{K} \tag{8b}$$

$$\text{s.t. } l_n^t \leq L_n \qquad \forall n \in \mathcal{N}. \tag{8c}$$

Equation (8a) represents the goal to minimize the average number of HOs (HO_{avg}) experienced by all UEs in the network. HO_k represents the total number of HOs experienced by each UE k since the beginning of their

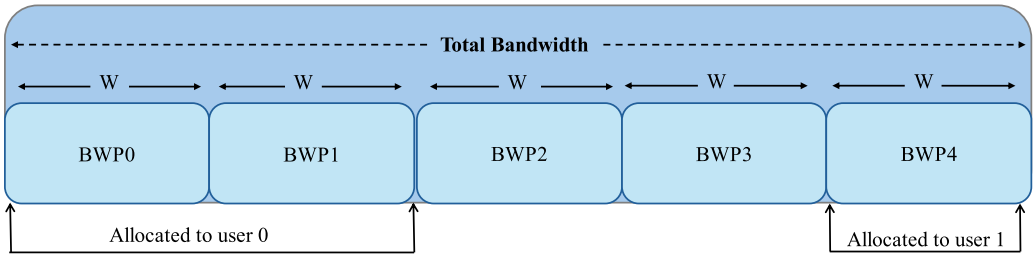

Fig. 3. BWP partitioning example. The total bandwidth is divided into five BWPs each of bandwidth W. BWP0 and BWP1 are allocated to user 0 (whose resource requirement is 2 W), while BWP4 is allocated to user 1 (whose resource requirement is 1 W).

connection. The objective is to find at each instant t of the connection period a configuration ($X_{k,n}^t$) that minimizes the average number of HOs for all UEs. Equation (8b) represents the constraint to guarantee the least acceptable link quality by setting a minimum threshold for the elevation angle. Equation (8c) represents the goal of minimizing the average blocking rate by ensuring that the total load of SAT_n is less than or equal to its total number of available channels L_n.

The problem formulated in (8) is a combinatorial integer optimization problem, which is NP-hard in general. To solve this problem, it will be transformed into a MARL-based optimization problem based on stochastic game in the following section.

IV. METHODOLOGY

In this section, we first introduce the RL-based satellite HO strategy implemented in [26] to investigate its performance and challenges when addressing multiple TPs. Next, we propose a deep-RL-based approach that overcomes the computational challenges of the RL-based approach and offers a solution to the multi-TP problem.

A. Satellite HO Based on RL

Our key issue is to determine which is the best satellite each user should connect to during a HO event to retain an extended connection and thus minimize the number of following unnecessary HO events while balancing the load among the satellites. We assume that each UE decides whether to do a HO or not and to which satellite once per second.

RL is a computational approach that may be used to analyze and automate goal-directed learning. It distinguishes itself from other computational techniques by relying on an agent learning via multiple and direct interactions with its environment, without necessitating perfect supervision or full environment models [32]. Any RL problem can be defined by the five-tuple $\{g, \mathcal{S}, \mathcal{A}, r, \pi\}$. $g \in \mathcal{G}$ represents the RL agent, which is the component that interacts with the environment at each time step t ($\mathcal{G}$ is the set of all agents). The agent observes the current state of the environment $s^t \in \mathcal{S}$ ($\mathcal{S}$ is the set of all possible states of g), and then takes an action $a^t \in \mathcal{A}$ ($\mathcal{A}$ is the set of all possible actions of g). Based on the taken action a^t, the agent obtains the instantaneous reward r^{t+1} according to the predefined reward function r. The specified action in state s^t may cause the transition to a new state $s^{t+1} \in \mathcal{S}$. Finally, π is the policy that guides the agent to choose the suitable action according to its current state. In the case of MARL, where G agents are considered, G tuples $\{g, \mathcal{S}_g, \mathcal{A}_g, r_g, \pi_g\}$ will be defined. They represent the five-tuple values of agent $g \in \mathcal{G}$, where $\mathcal{S}_g$ and $\mathcal{A}_g$ are the subsets of $\mathcal{S}$ and $\mathcal{A}$ identifying the states and actions related to agent g, respectively.

In RL, the agent's goal is to discover the best policy π_* that maximizes the expected cumulative reward (denoted by Ω) gained by iterating through a number of consecutive episodes NoE which differ from a different initial configuration. An episode is the sequence of states visited by an agent during a period T from the initial state s^0 to the terminal state s^T. The progression of time within an episode refers to the progression of steps or interactions the agent makes within that specific episode. Moreover, multiple episodes involve the repetition of the learning process across different instances or runs of the same scenario, allowing the agent to learn from a variety of experiences and improve its performance over time.

The expected cumulative reward of user k Ω_k for an episode of duration T is given as follows [21]:

$$\Omega_k(s, \pi) = \sum_{t=0}^{T} \gamma E\{r_k^t | s^0 = s, \pi\} \tag{9}$$

where $\gamma \in [0, 1)$ is the discount factor used to weigh the rewards.

Our goal lies under model-free learning, in which agents learn about their environment by using trial and error. Q-learning is one of the most popular RL algorithms in the model-free technique [33]. The term "Q" refers to a function that has a straightforward updating mechanism. The agent begins with arbitrary initial values for $Q(s, a)$ for all $s \in \mathcal{S}$ (usually set to zero) and then updates the Q-values with the learning rate $\alpha^t \in [0, 1)$ as follows:

$$Q^{t+1}(s^t, a^t) = (1 - \alpha^t) Q^t(s^t, a^t) + \alpha^t [r^t + \gamma \max_a Q^t(s^{t+1}, a^t)]. \tag{10}$$

In our case, a decentralized satellite HO strategy is needed since UEs can only access partial information about the satellite system while competing for the restricted available

communication resources of the satellites. Thus, we implemented a decentralized multiagent Q-learning strategy where all its components can be defined as follows.

1) *Agent:* Each UE is considered to be an agent that independently takes actions. The set of agents $\mathcal{G}$ is equal to the set of UEs $\mathcal{K}$.
2) *State:* The state s_k^t at time t represents the current observations that agent g_k receives from the environment defined as the three-tuple $s_k^t =< \overline{C_k^t}, \overline{l^t}, \overline{V_k^t} >$, in which $\overline{C_k^t}$, $\overline{l^t}$, and $\overline{V_k^t}$ are vectors with size N. $\overline{C_k^t} = [C_{k,0}^t, C_{k,1}^t, \ldots, C_{k,n}^t, \ldots, C_{k,N}^t]$ includes the coverage indicator between g_k and $\text{SAT}_n \forall n \in N$ at time t as defined in (3). Similarly, $\overline{l^t} = [l_0^t, l_1^t, \ldots, l_n^t, \ldots, l_N^t]$ denotes for every SAT_n the number of loaded channels at time t, and $\overline{V_k^t} = [V_{k,0}^t, V_{k,1}^t, \ldots, V_{k,n}^t, \ldots, V_{k,N}^t]$ contains information about the RVT between all the satellites and agent k.
3) *Action:* It represents a choice by an agent to attach to one of the satellites $n \in \mathcal{N}$. An action of an agent k at time t is defined in this article as a_k^t, where it is equal to one of the satellites $n \in \mathcal{N}$ such that $C_{k,n}^t = 1$.
4) *Reward:* The adopted Q-learning reward function is defined as follows:

$$r_k^t(s_k^t, a_k^t) = \begin{cases} -p_1 & \text{if } C_{k,n}^t = 1, X_{k,n}^t = 0 \\ -p_2 & \text{if } C_{k,n}^t = 1, X_{k,n}^t = 1 \\ & l_n^t > L_n \\ f(t,k,n) & \text{if } C_{k,n}^t = 1, X_{k,n}^t = 1 \\ & l_n^t \leq L_n. \end{cases} \tag{11}$$

According to (11), when an action leads to a HO or blocking, the instantaneous reward function is associated with penalties $p_1 = 300$ and $p_2 = 100$, respectively. However, a positive reward $f(t,k,n) = V_{k,n}^t + w_n^t$ is offered when the action avoids HO and blocking. The reward value is bigger when the agent picks a satellite with higher RVT delaying the need for consecutive HOs, and hence decreasing the average number of HOs, and lower load, where $w_n^t = L_n - l_n^t$ indicates SAT_n's resources accessible at time t, encouraging the load balancing among satellites, hence decreasing the average blocking rate.

When establishing an action selection policy, it is crucial to balance exploitation with exploration: exploitation occurs when an agent picks the optimal action based on the current Q-values (also known as a greedy policy); exploration includes the agents attempting more previously unexplored actions in order to explore a broader action space. We combine the ϵ-greedy policy ($0 \leq \epsilon \leq 100$), where the agent chooses the best action (i.e., the one with the highest Q-value) for $\epsilon\%$ of the time, with the Boltzman exploration, where the agent chooses better random actions for $(100 - \epsilon)\%$ of the time as follows:

$$a_*^t = \begin{cases} \arg\max_{a^t} \pi^t(a^t) & \text{if } \varepsilon < \epsilon^t \\ \arg\max_{a^t} Q_k^t(s_k^t, a^t) & \text{otherwise} \end{cases} \tag{12}$$

Algorithm 1: Multiagent Q learning.

Initialise:
$t = 0$
$l_n = l^0 \ \forall n \in \mathcal{N}$
$\mathcal{S}_k = \{s_k^0\} =< \overline{C_k^0}, \overline{l^0}, \overline{V_k^0} > \forall k \in \mathcal{K}$
$Q_k^0(s_k^0, a_k^0) = 0 \ \forall k \in \mathcal{K}$
While $ep < NoE$
While $t < T$
for $k \in \mathcal{K}$ **do**
  Choose random agent k;
  Observe $s_k^t =< \overline{C_k^t}, \overline{l^t}, \overline{V_k^t} >$;
  Choose action a_k^t based on Eq. (12);
  Move to a new state
  $s_k^{t+1} =< \overline{C_k^{t+1}}, \overline{l^{t+1}}, \overline{V_k^{t+1}} >$;
  Get the reward r_k^{t+1};
  Update Satellites-Load to l^{t+1};
  if $s_k^{t+1} \in \mathcal{S}_k$ **then**
    Update the Q-value $Q_k^t(s_k^t, a_k^t)$ by Eq. (10);
  else
    Add the new state s_k^{t+1} to $\mathcal{S}_k$;
    Initialise the Q-value of the new state to zero;
  end if
end for
End While
Reset $t = 0$, $l_n = l^0 \ \forall n \in \mathcal{N}$, $\mathcal{S}_k = \{s_k^0\} \quad \forall k \in \mathcal{K}$;
end while

where ϵ^t increases linearly with time to motivate the agents to begin their learning process by exploring more and then begin to exploit with probability increasing with t.

The action's selection probability $\pi^t(a^t)$ is weighted by its associated Q-value as follows:

$$\pi^t(a^t) = \frac{\exp \frac{Q_k^t(s_k^t, a^t)}{\tau}}{\sum_{a^t} \exp \frac{Q_k^t(s_k^t, a^t)}{\tau}} \tag{13}$$

where τ is the temperature factor. It regulates the likelihood of performing actions other than the one with the greatest Q-value, also known as degree of exploration. When τ is high, all options are equally considered; when it is low, high-rewarding options are more likely to be selected.

For the implementation of the distributed MARL, each UE observes its current state at each time step, chooses an action based on a policy π, and updates its Q-table according to (10). Due to the fact that satellites are constantly moving, the UE's covering set of satellites and the corresponding RVT change at every time instance, causing a transaction to a new state independently of the taken action. Simultaneously, the agent's action may change each satellite load, causing a change in the state too. As a result, there is a vast array of potential states that are challenging to forecast and characterize at the beginning of learning.

The method in Algorithm 1 is proposed in this article to resolve this issue.

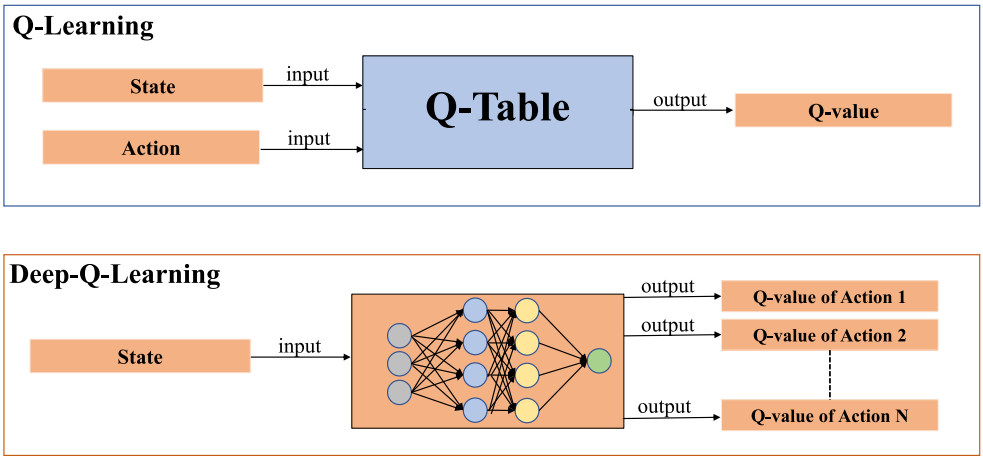

Fig. 4. Q-learning versus DQL: difference in the agent's brain. In Q-learning, the Q-table serves as the agent's brain, where the state and the action of the agent are the inputs of the Q-table, and the Q-value of those inputs is the output of the Q-table, while in DQL, a DNN acts as the brain, where the state is the input of the DNN, and the approximation of all the Q-values of each corresponding possible action for this specific state is the output of the neural network.

At the start of the learning ($t = 0$), the set of states for each agent k contains only one state $\mathcal{S}_k = \{s_k^0\}$. The load distribution among satellites, RVT, and $\mathcal{N}_k$ then change correspondingly after each time step, leading the agent to shift to a new state s_k'. If s_k' is already in the set of states, its Q-value is updated; otherwise, s_k' is added to $\mathcal{S}_k$ and its Q-value is initialized to zero. Then, at each instant t, each agent observes its current state s_k^t and chooses an action a_k^t following a policy π. After this, it will receive the corresponding instantaneous reward r_k^{t+1} and update its own Q-table according to (10). The agents execute actions successively one after another, following a different random sequence at each instant. When the agent k takes an action, the load on the satellites changes correspondingly. Information about an agent's action can be acquired by the other agents before the next one makes its own action, even if the agents are unaware of each other's reward functions and Q-tables.

The MARL satellite HO optimization method achieves remarkable results while considering only one TP for all UEs [26]. However, randomly changing the resource requirements for each UE after each episode in order to consider different and changing applications considerably increases the number of states. This results in an enormous Q-table, which makes it almost impossible for an agent to effectively learn in a reasonable amount of time, so limiting the Q-learning use to only one type of traffic application, as will be shown in Section V. To solve this problem, we consider deep Q-learning (DQL), which uses deep neural networks (DNNs) instead of Q-table to predict the Q values for each action.

B. Satellite HO Based on Deep-RL

There are two issues with having an extremely large number of states and actions. The first one is that as the number of states grows, more memory is needed to keep and update the state–action table. Second, as the number of states grows, it takes much more time to thoroughly examine each state and appropriately populate the Q-table [32]. Those two issues limit the usability of Q-learning only to small-scale models, while the learning process becomes out of control when dealing with large-scale models.

To overcome the above limitations, we considered DQL. It differs from the standard Q-learning by the agent's brain. In Q-learning, the Q-table serves as the agent's brain, while in DQL, a DNN provides a good approximation of the Q-value of an action, i.e., $Q(s, a) \approx Q(s, a, \Lambda)$, where Λ is the set of weights of the DNN, which is updated after each learning phase, thus mapping from partially observed states to actions rather than fully observing every state and keeping a list of the corresponding Q-values in an enormous lookup table [34], as shown in Fig. 4. The selection of DQL as the primary RL algorithm is underpinned by several compelling factors. First and foremost, our satellite communication system poses a challenge due to its inherent complexity. This complexity arises from dynamic LEO satellite positions, fluctuating UE resource requirements, and the need to balance multiple performance objectives. DQL emerges as a well-suited choice for addressing such intricate problems, especially when dealing with discrete action spaces. Data efficiency is another paramount consideration. In real-world applications, data collection can be resource-intensive and costly. DQL's ability to learn efficiently from a limited dataset proves advantageous, making it suitable for practical deployment. Furthermore, the simplicity and interpretability of the chosen algorithm also weigh in the decision as it offers a straightforward architecture, enhancing the comprehensibility of learned policies, in addition to the training stability, incorporating techniques, such as experience replay and target networks. These mechanisms contribute to faster convergence and improve the assurance of reliable results.

In DQL, the state is provided as the input to the modeled DNN and the output is the estimated Q-value for every possible action that might be taken in the given observed state.

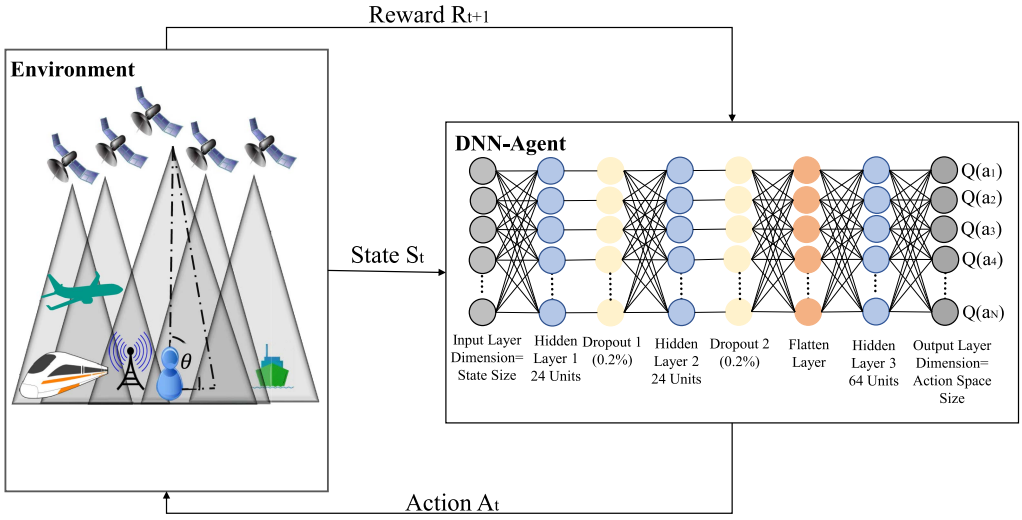

Fig. 5. Compositions of the proposed DQL strategy: environment and deep neural network (DNN) structure. The environment is composed of a number of LEO satellites and users. The DNN (agent's brain) is composed of an input layer of dimensions equal to the state size, three fully connected hidden dense layers to learn the features of the state information, two drop-out layers, a flatten layer, and an output layer of dimension equal to the action space size.

We consider the same state and action spaces defined in Section IV-A for the Q-learning methodology. The structure of the proposed DNN is shown in Fig. 5.

The proposed DNN is composed of an input layer of dimensions equal to the state size, three fully connected dense layers to learn the features of the state information, a flatten layer used to expand the output of the previous dense layer into a vector before inputting it into the following fully connected dense layer to get the Q-values for each action $a \in \mathcal{A}$, and an output layer of dimension equal to the action space size. We also include in the model two dropout layers with a probability of 0.2% for their well-known ability to avoid overfitting problems [35]. The input of the proposed DNN is the state s_k^t of UE_k, which is transformed into a matrix $\Phi(s_k^t)$ before being input into the DNN. We consider each UE as an agent also in this case for the same reason of avoiding network congestion.

According to the multiobjective optimization problem of reducing the total number of HO events while minimizing the blocking rate, we define the reward function in four different cases as follows:

$$r_k^t(s_k^t, a_k^t) = \begin{cases} -z_1 & \text{if } X_{k,n}^t = 0, l_n^t > L_n \\ -z_2 & \text{if } X_{k,n}^t = 0, l_n^t \le L_n \\ f_1(t,k,n) & \text{if } X_{k,n}^t = 1, l_n^t > L_n \\ f_2(t,k,n) & \text{if } X_{k,n}^t = 1, l_n^t \le L_n. \end{cases} \tag{14}$$

The first two cases represent when an action leads to a HO and the agent chooses to move to a satellite with enough resources or a loaded satellite. The agent will receive a high negative penalty $z_1 = 500$ or a moderate negative penalty $z_2 = 300$, respectively, as the first case will result in blocking. On the other hand, the other two cases represent when an action does not result in any HO and the UE is connected to a loaded satellite or not. The agent will receive a negative penalty $f_1(t,k,n) = 100 * w_n^t$ (since $w_n^t < 0$ for loaded satellites), in order to motivate it to always move to a satellite with the lower load if necessary to avoid blocking, or a big positive reward $f_2(t,k,n) = V_{k,n}^t$, in order to motivate it to move to a satellite with higher RVT when necessary to avoid consecutive future HOs, respectively.

Moreover, in order to make the learning more stable and convergent, DQL uses experience replay and mini-batch learning, which randomizes collected data samples and smooth data distribution changes. After each time step, the resultant transformations $(\Phi(s_k^t), a_k^t, \Phi(s_k^{t+1}), r_k^{t+1})$ are progressively stored in a replay memory $\mathbb{M}$. The deep Q-network is updated by using experience replay in a supervised learning-based method that takes into account both recent and past experiences. Therefore, to update the network, experience replay stores observed transitions in the replay memory $\mathbb{M}$ and samples evenly a mini-batch from this memory bank. As a result, the Q-network can reduce memory correlation, enhancing learning effectiveness [36].

The DQL calculates the cost together with the memory randomly sampled from $\mathbb{M}$ at each time step and trains the network that adjusts the Q-value parameter based on the loss function provided by

$$L(\Lambda) = \mathbb{E}[y^j - Q(\Phi(s^j), a^j; \Lambda)^2] \tag{15}$$

where

$$y^j = \begin{cases} r^j & \text{if } \Phi(s^{j+1}) = \Phi(s^T) \\ & \text{i.e.,} s^{j+1} \text{ is a terminal state} \\ r^j + \gamma \max_{a^{j+1} \in \mathcal{A}} Q(\Phi(s^{j+1}), a^{j+1}; \Lambda) & \\ & \text{otherwise.} \end{cases} \tag{16}$$

The adopted action selection policy in the proposed MADQL is an ϵ-greedy policy as follows:

$$a_*^t = \begin{cases} \text{Random action } a' \in \mathcal{A} & \text{if } \varepsilon < \epsilon^t \\ \arg\max_{a^t} Q_k(s_k^t, a^t, \Lambda) & \text{otherwise} \end{cases} \tag{17}$$

where ϵ^t increases linearly with time to motivate the agents to begin their learning process by exploring more and then begin to exploit with a probability increasing with t.

Algorithm 2 summarizes the DQL agent learning process. Every UE is an independent DQL agent, therefore the well-trained DNN is implemented on each UE. At the beginning, the state of each UE_k $\forall k \in \mathcal{K}$ is initialized to $s_k^0 \in \mathcal{S}_k$, the current TP per UE is randomly chosen to ensure the variation and diversity of the applications used by each UE during each episode, and the load distribution among satellites is initialized accordingly. At each time step t, each agent observes its current state s_k^t, chooses an action a_k^t according to (17) causing a transition to a new state s^{t+1}, and receives a reward r_k^{t+1} computed following (14). After each transition to a new state, both vectors s_k^t and s_k^{t+1} are transformed into the matrices $\Phi(s_k^t)$ and $\Phi(s_k^{t+1})$, respectively, in order to input the whole experience $(\Phi(s_k^t), a_k^t, \Phi(s_k^{t+1}), r_k^{t+1})$ in the replay memory $\mathbb{M}$. Then, a mini-batch of size N_b is randomly sampled from the whole replay memory. Note that we set the maximum size of the replay memory to N_m so that old experiences are deleted, starting from the oldest ones, when space is needed to add new ones.

Algorithm 2: Multiagent deep-Q learning.

Initialise:
$\mathbb{M}$ - Empty experience replay buffer
N_b - Mini Batch size
N_m - Replay Memory Size
Λ - DNN Weights initialisation
$t = 0$
$\mathcal{S}_k = \{s_k^0\} =< \overline{C_k^0}, \overline{l^0}, \overline{V_k^0} > \forall k \in \mathcal{K}$
While $ep < NoE$
  Allocate the appropriate RR for the randomly selected TP per agent;
  $l_n = l^0 \ \forall n \in \mathcal{N}$
While $t < T$
for $k \in \mathcal{K}$ **do**
    Choose random agent k;
    Observe $s_k^t =< \overline{C_k^t}, \overline{l^t}, \overline{V_k^t} >$;
    Choose action a_k^t based on Eq. (17);
    Move to a new state $s_k^{t+1} =< \overline{C_k^{t+1}}, \overline{l^{t+1}}, \overline{V_k^{t+1}} >$;
    Transform s_k^t and s_k^{t+1} into matrices $\Phi(s_k^t)$ and $\Phi(s_k^{t+1})$, respectively;
    Get the reward r_k^{t+1} computed following Eq. (14);
    Update Satellites-Load to l^{t+1};
    Store $(\Phi(s_k^t), a_k^t, \Phi(s_k^{t+1}), r_k^{t+1})$ in $\mathbb{M}$;
    Sample a random mini-batch of size N_b from $\mathbb{M}$ for training;
    Train the DNN of agent k;
    Set y_t following Eq. (16);
    Update the DNN parameters Λ by performing a gradient descent step on Eq. (15);
  end for
End While
  Reset $t = 0$, $\mathcal{S}_k = \{s_k^0\} \ \forall k \in \mathcal{K}$;
end while

V. PERFORMANCE EVALUATION

A. Scenario Setup

To evaluate the performance achievable by using our proposed solution, we used a STIN simulator that we implemented and described in [30]. It is a network-level simulator based on the discrete-time discrete-event network simulator NS-3 that simulates packet data networks by using custom traffic models [37]. Since its official release does not allow simulating satellite communication networks, the software has been modified including a developed satellite mobility module that enables the integration of nodes moving and acting as LEO satellites in the NS-3 simulation environment. The SGP4 mathematical model, which is frequently employed to estimate the speed and location of LEO satellites, serves as the foundation for the module [38].

To prove the efficiency of the proposed strategy, we decided to compare it with the following four other approaches from the literature that we implemented within NS-3.

1) *Minimum distance:* Single-criterion strategy, the HO choice is made only on the basis of the minimum separation distance between the UE and the associated satellite, i.e., each UE decides to connect to the nearest satellite at any given time.
2) *Minimum load:* Single-criterion strategy, the HO choice is made only on the basis of the minimal load per satellite, i.e., each UE decides to connect to the satellite with the minimum load among the ones in visibility from the UE at any given time.
3) *Multicriteria load aware (MCLA) [21]:* MCLA strategy based on RL. The HO event is made considering the minimum elevation angle and the available satellite channels.
4) *MARL [26]:* Multicriteria load balancing HO strategy based on RL. The HO event is made considering the RVT, the satellite load, and the elevation angle.

The simulation parameters are all summarized in Table I.We assume that the UEs are uniformly spread over the Earth's surface and the satellites are all at the same altitude, uniformly spread among the multiple orbital planes, and equally spaced within each plane. The learning rate (α) is set to 0.1, which is commonly chosen as a default value since it allows balancing between exploration and exploitation and enables the agent to update gradually, incorporating valuable feedback while exploring different actions. Furthermore, the exploration rate ϵ is chosen to increase linearly with time to motivate agents to explore more at the beginning, gathering information about the environment. Then, as time progresses, ϵ decreases, promoting a shift toward exploitation. This balance between exploitation and exploration is crucial to allow agents to learn while still leveraging their acquired knowledge. The temperature

TABLE I
Simulated Scenarios Parameters

Parameter	Single TP	Multiple TPs
Number of satellites N	48	50
Number of UE K	6	6
Satellite altitude H	600 km	600 km
Number of orbital planes	4	5
Number of satellites per orbital plane	12	10
Orbital planes eccentricity	0 (circular)	0 (circular)
Orbital planes inclination i	88°	88°
Orbital planes argument of perigee	90°	90°
Minimum elevation angle between UEs and satellites for transmissions θ_0	20°	20°
Number of available satellite BWPs	5	7
Bandwidth of one BWP W	50 MHz	50 MHz
Total Spectrum Bandwidth	250 MHz	350 MHz
Number of possible TPs	1	4
α	0.1	0.1
γ	0.95	0.99
ϵ	0.1–0.82	0.1–0.9
τ	10	-
NoE	2500	1200
Duration of an episode T	600 s	120 s

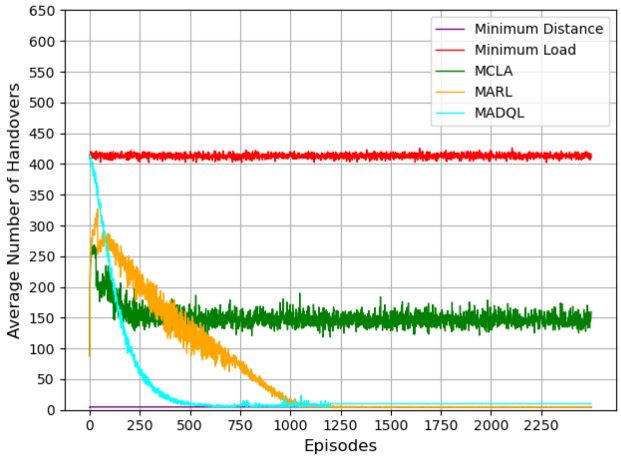

Fig. 6. Single TP: Average number of HOs as a function of the episodes for the five considered approaches: minimum distance, minimum load, MCLA(load aware), MARL(load balance), and MADQL.

factor (τ) is set to 10, to increase the randomness in the action selection process during the exploration phase, which encourages the agent to explore a wider range of actions and avoid getting stuck in suboptimal choices. T in Table I refers to the number of steps of an episode, which reflects the actual movement and position of the satellites within a time duration of T seconds that can be predicted in advance by the STIN simulator [30].

B. MADQL Satellite HO for a Single TP

In this section, we set all the simulation parameters to the same values adopted in [26] that are summarized in Table I, "Single TP" column. Fig. 6 depicts the average number of HOs as a function of the episodes for each of the five HO approaches.

The minimum load method results in the highest average number of HOs since it takes into account only one criterion

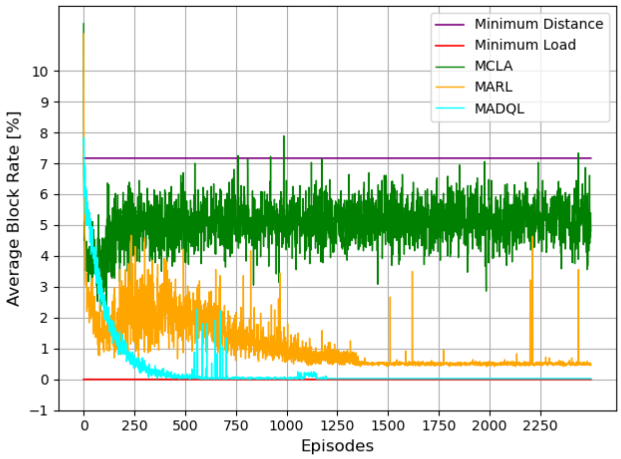

Fig. 7. Single TP: Average blocking rate as a function of the NoE for the five considered approaches: minimum distance, minimum load, MCLA(load aware), MARL(load balance), and MADQL.

that is affected by each HO event that, in some cases, leads to multiple HO event chains. However, the MARL HO strategy outperforms the MCLA approach, with 95% lower number of HOs per user. Furthermore, the MARL approach converges to the same average HO value (around 3.7 HOs) as the minimum distance method, which is known in the literature to achieve the minimum possible number of HOs. Similarly, the MADQL approach also converges to around only 4.2 HOs very close to the minimum distance approach. Fig. 6 also shows that MADQL starts to converge faster than the MARL due to the ability of DNNs to predict the Q-values of newly visited states rather than requiring to go through all the states more than once to calculate the Q-values explicitly.

Fig. 7 shows the obtained blocking rate as a function of the episodes. Although the minimum distance strategy achieves the minimum number of HOs, it results in the highest blocking rate, because it ignores the load restriction of each satellite. The minimum load approach achieves zero blocking rate at the cost of a huge number of consecutive HOs and a consequent relevant decrease of the achievable QoS due to the HO process times. Moreover, Fig. 7 shows that the MARL strategy achieves a very low blockage rate of 0.42%, outperforming the MCLA approach by almost 84%. While, on the other hand, the MADQL approach further outperforms the MARL strategy by converging to the minimal blocking rate of 0.033%.

These findings imply that the proposed MADQL method avoids distributing UEs to overloaded satellites, reducing the likelihood of blockage while achieving a minimum number of delay overheads resulting from the few HOs. This proves that the proposed MADQL method achieves remarkable results in solving the HO optimization problem summarized (8) while considering one TP.

C. MADQL Satellite HO for Different and Varying TPs

In this section, the considered simulation parameters are summarized in Table I, "Multiple TPs" column. As a first

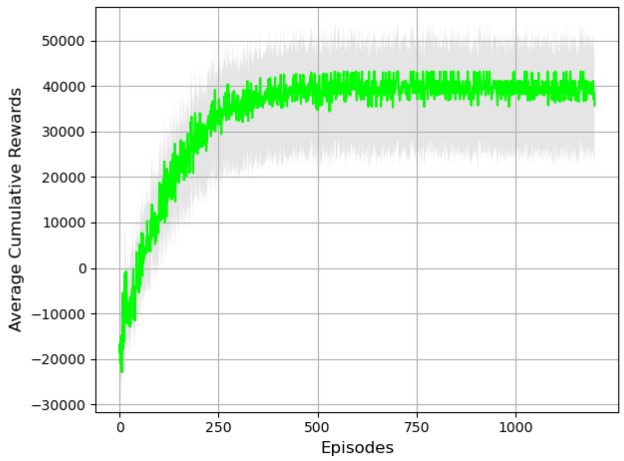

Fig. 8. Multiple TPs—MADQL: Average cumulative rewards as a function of the NoE.

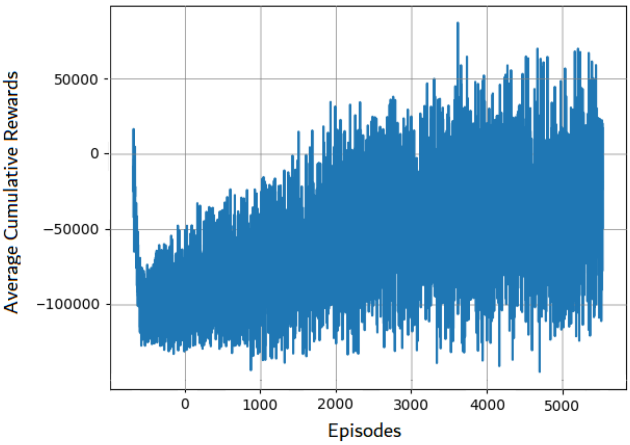

Fig. 9. Multiple TPs—MARL: Average cumulative rewards as a function of the NoE.

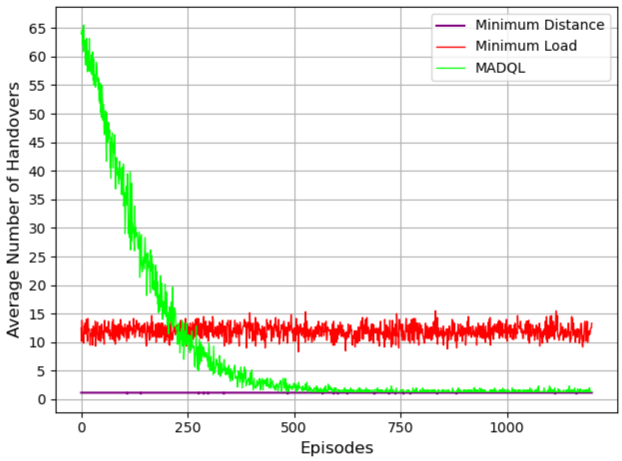

Fig. 10. Multiple TPs—MADQL versus minimum load and minimum distance: Average number of HOs as a function of the NoE.

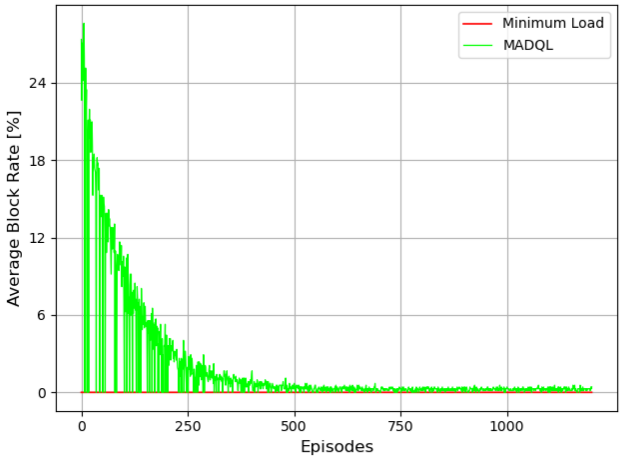

Fig. 11. Multiple TPs—MADQL versus minimum load: Average blocking rate as a function of the NoE.

step to see the difference between the two proposed solutions, Figs. 8 and 9 depict the average cumulative rewards as a function of the episodes.

For the MADQL solution, as the computed episodes increase from 0 to 1200, the average cumulative rewards increase from −3000 to around 40 000 starting to converge after only 250 episodes. The figure shows the average cumulative reward for all the independent agents, where the gray area is the range within which they all converge to their own different optimal values. Instead, as illustrated in Fig. 9, the MARL strategy was unable to learn even after increasing the number of episodes (NoE) to 5000, resulting in a divergent curve. The main reason is the massive increase in the number of states that results in an enormous Q-table due to the random change of the resource requirements for each UE after each episode made to consider different and changing applications.

These observations imply that Q-Learning-based methods, i.e., MARL and MCLA, are limited to the usage of only one type of traffic application and are not applicable when addressing multiple types of traffic applications per user. Thus, MARL and MCLA approaches will not be considered as valid comparison methods in the following sections. To prove the efficiency of the proposed MADQL method for multiple TPs, we will compare it to the minimum distance and minimum load approaches.

Fig. 10 illustrates the gain of the proposed MADQL method in terms of the average number of HOs compared to the minimum distance and minimum load methods.

The minimum load method achieves the highest average number of HOs, which is between 10 and 14 HOs in the 120-s episode duration. Fig. 10 also shows that the proposed MADQL method converges to only about 1.2 HOs, which is the same value achieved by the minimum distance method, which is known to achieve the least possible average number of HOs. This proves the efficiency of our proposed MADQL HO optimization method when considering different and varying applications per UE in terms of minimizing the average number of HOs.

On the other hand, in Fig. 11, we compare the average blocking rate for the proposed MADQL with the minimum load method only, as it has been previously proved that the minimum distance approach performs poorly with respect to blocking rate. This is due to the fact that the minimum distance method does not take into consideration either the resource requirements of each UE at each instance or the load constraint of each satellite while making a HO decision.

TABLE II
Methods Comparison: Comparison of the Minimum Distance, Minimum Load, MCLA, MARL, and MADQL Methods for a Single Traffic Profile (Left-Hand Side) and for Multiple Traffic Profiles (Right-Hand Side)

Method	Single TP			Multiple TP		
	Applicable	Average handover	Average block rate	Applicable	Average handover	Average block rate
Minimum distance	YES	3.7	7.2%	YES	1.2	0%–60%
Minimum load	YES	400	0%	YES	10–14	0%
MCLA [21]	YES	150	4%–7%	NO	-	-
MARL [26]	YES	3.7	0.42%	NO	-	-
MADQL	YES	4.2	0.033%	YES	1.2	0.03%

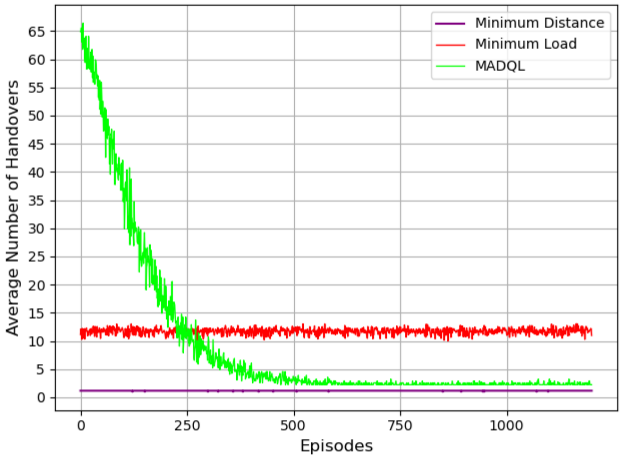

Fig. 12. Multiple TPs—MADQL versus minimum load and minimum distance: Average number of HOs as a function of the NoE with 80% of the users requiring a high number of resources (TP1 or TP2) and 20% a low number of resources (TP3 or TP4).

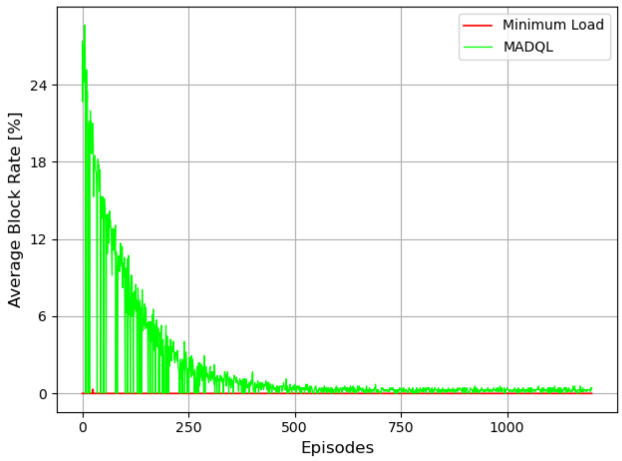

Fig. 13. Multiple TPs—MADQL versus minimum load: Average blocking rate as a function of the NoE with 80% of the users requiring a high number of resources (TP1 or TP2) and 20% a low number of resources (TP3 or TP4).

On the contrary, the minimum load method achieves a consistent 0% blocking rate, as shown in Fig. 11, while the proposed MADQL method converges to about only 0.03% of blocking. This proves the efficiency of our proposed MADQL HO optimization method when considering different and varying applications per UE in terms of minimizing the average blocking rate.

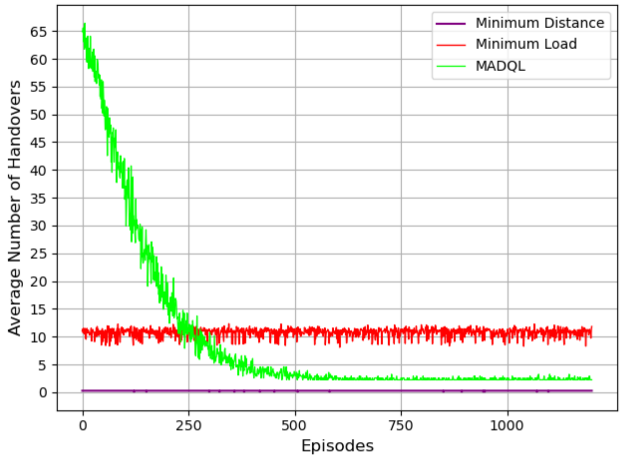

Fig. 14. Multiple TPs—MADQL versus minimum load and minimum distance: Average number of HOs as a function of the NoE with 20% of the users requiring a high number of resources (TP1 or TP2) and 80% a low number of resources (TP3 or TP4).

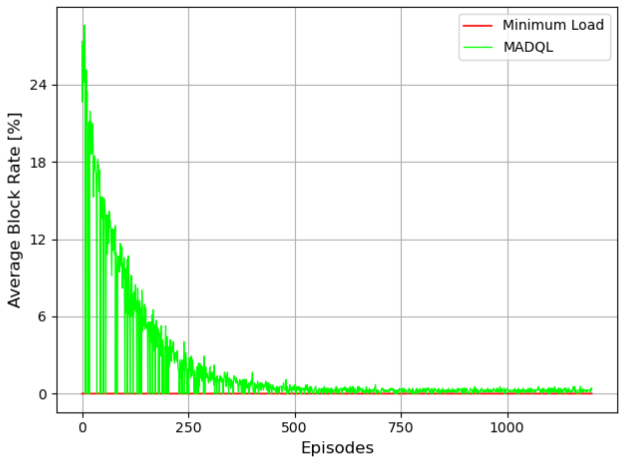

Fig. 15. Multiple TPs—MADQL versus minimum load: Average blocking rate as a function of the NoE with 20% of the users requiring a high number of resources (TP1 or TP2) and 80% a low number of resources (TP3 or TP4).

To sum up, the comparison of all the mentioned methods for single and multiple TPs is summarized in Table II.

With new applications and services continuously emerging and evolving within the communication landscape, it is indeed impossible to predict every possible service or application that may be introduced to the system. However, our approach is intentionally designed to address this

uncertainty by harnessing the learning capabilities of RL algorithms. These algorithms can adapt and learn from new scenarios, albeit with some adaptation time. In order to test the resilience of the proposed MADQL HO optimization method, we run two more tests changing the percentage of users requiring W or 2 W resources from the satellite. In detail, in the first test, 80% of the users require 2 W resources (TP_k = TP1 or TP2) and 20% require W resources (TP_k = TP3 or TP4), while, in the second test, 20% of the users require a 2 W resources and 80% of them require W resources.

Figs. 12 and 13 illustrate the results of the first test in terms of the average number of HOs and the average blocking rate as a function of the NoE, respectively. Even when most of the users have high requirements, the proposed MADQL approach achieves great results in terms of the average number of HOs by converging to the same value achieved by the minimum distance approach. In the same way, the proposed MADQL approach achieves great results with an average blocking rate of approximately 0.05%.

Similarly, Figs. 14 and 15 illustrate the results of the second test, whose results are in line with what has been already shown about the first test.

VI. CONCLUSION

In this article, we proposed a novel MADQL satellite HO optimization strategy that addresses UEs with different and varying TPs, which, to the best of the authors' knowledge, is the first time in the recent literature. The proposed method was implemented and tested in a realistic STIN simulator, which proves its efficiency in managing STIN HOs for users with varying TPs by achieving a very low number of HOs and blocking rate while balancing the load among the satellites and enhancing user's experience. Specifically, the proposed method achieves approximately 60% reduction in HO rate and around 91% reduction in blocking rate compared to single-criterion approaches. Moreover, we conducted a sensitivity analysis for different possible TPs distribution scenarios, which demonstrated the robustness of the proposed MADQL HO optimization method. Future work will consist of implementing the proposed HO strategy as a built-in tool within the NS3-based STIN simulator implemented in [30] and make it accessible to the research community. In addition, extend the proposed scenario to an end-to-end STIN with and without the presence of inter-satellite links and incorporate a channel model to allow us to accurately compute additional parameters, such as the end-to-end delay and the signal-to-interference-plus-noise ratio, which could be used both to evaluate the network under varying channel conditions and as additional input information of the HO strategy.

REFERENCES

[1] G. Giambene, E. O. Addo, and S. Kota, "5G aerial component for IoT support in remote rural areas," in *2019 IEEE 2nd 5G World Forum (5GWF)*, 2019, pp. 572–577.

[2] M. Marchese, F. Patrone, and A. Guidotti, "The role of satellite in 5G and beyond," in *A Roadmap to Future Space Connectivity: Satellite and Interplanetary Networks*. Berlin, Germany: Springer, 2023, pp. 41–66.

[3] M. Bacco et al., "Networking challenges for non-terrestrial networks exploitation in 5G," in *2019 IEEE 2nd 5G World Forum (5GWF)*, 2019, pp. 623–628.

[4] L. Boero, R. Bruschi, F. Davoli, M. Marchese, and F. Patrone, "Satellite networking integration in the 5G ecosystem: Research trends and open challenges," *IEEE Netw.*, vol. 32, no. 5, pp. 9–15, Sep./Oct. 2018.

[5] F. Rinaldi et al., "Non-terrestrial networks in 5G & beyond: A survey," *IEEE Access*, vol. 8, pp. 165178–165200, 2020.

[6] M. Giordani and M. Zorzi, "Non-terrestrial networks in the 6G era: Challenges and opportunities," *IEEE Netw.*, vol. 35, no. 2, pp. 244–251, Mar./Apr. 2021.

[7] A. Guidotti et al., "Architectures and key technical challenges for 5G systems incorporating satellites," *IEEE Trans. Veh. Technol.*, vol. 68, no. 3, pp. 2624–2639, Mar. 2019.

[8] R. Deng, H. Qin, H. Li, D. Wang, and H. Lyu, "Non-cooperative LEO satellite orbit determination based on single pass Doppler measurements," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 59, no. 2, pp. 1096–1106, Apr. 2023.

[9] N. U. Hassan, C. Huang, C. Yuen, A. Ahmad, and Y. Zhang, "Dense small satellite networks for modern terrestrial communication systems: Benefits, infrastructure, and technologies," *IEEE Wireless Commun.*, vol. 27, no. 5, pp. 96–103, Oct. 2020.

[10] G. Inalhan, M. Tillerson, and J. P. How, "Relative dynamics and control of spacecraft formations in eccentric orbits," *J. Guid., Control, Dyn.*, vol. 25, no. 1, pp. 48–59, 2002.

[11] Y. Su, Y. Liu, Y. Zhou, J. Yuan, H. Cao, and J. Shi, "Broadband LEO satellite communications: Architectures and key technologies," *IEEE Wireless Commun.*, vol. 26, no. 2, pp. 55–61, Apr. 2019.

[12] S. Park and J. Kim, "Trends in LEO satellite handover algorithms," in *2021 12th Int. Conf. Ubiquitous Future Netw. (ICUFN)*. IEEE, 2021, pp. 422–425.

[13] S. Jung, M.-S. Lee, J. Kim, M.-Y. Yun, J. Kim, and J.-H. Kim, "Trustworthy handover in LEO satellite mobile networks," *ICT Exp.*, vol. 8, no. 3, pp. 432–437, 2022.

[14] S. Karapantazis and F.-N. Pavlidou, "QoS handover management for multimedia LEO satellite networks," *Telecommun. Syst.*, vol. 32, no. 4, pp. 225–245, 2006.

[15] E. Papapetrou and F.-N. Pavlidou, "Analytic study of Doppler-based handover management in LEO satellite systems," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 41, no. 3, pp. 830–839, Jul. 2005.

[16] T. Rehman, F. Khan, S. Khan, and A. Ali, "Optimizing satellite handover rate using particle swarm optimization (PSO) algorithm," *J. Appl. Emerg. Sci.*, vol. 7, no. 1, pp. 53–63, 2017.

[17] E. Juan, M. Lauridsen, J. Wigard, and P. Mogensen, "Handover solutions for 5G low-Earth orbit satellite networks," *IEEE Access*, vol. 10, pp. 93309–93325, 2022.

[18] Y. Wu, G. Hu, F. Jin, and J. Zu, "A satellite handover strategy based on the potential game in LEO satellite networks," *IEEE Access*, vol. 7, pp. 133641–133652, 2019.

[19] Z. Wang, L. Li, Y. Xu, H. Tian, and S. Cui, "Handover control in wireless systems via asynchronous multiuser deep reinforcement learning," *IEEE Internet Things J.*, vol. 5, no. 6, pp. 4296–4307, Dec. 2018.

[20] E. Juan, M. Lauridsen, J. Wigard, and P. Mogensen, "Performance evaluation of the 5G NR conditional handover in LEO-based non-terrestrial networks," in *2022 IEEE Wireless Commun. Netw. Conf.*. IEEE, 2022, pp. 2488–2493.

[21] S. He, T. Wang, and S. Wang, "Load-aware satellite handover strategy based on multi-agent reinforcement learning," in *2020 IEEE Glob. Commun. Conf*, 2020, pp. 1–6.

[22] H. Xu, D. Li, M. Liu, G. Han, W. Huang, and C. Xu, "QoE-driven intelligent handover for user-centric mobile satellite networks," *IEEE Trans. Veh. Technol.*, vol. 69, no. 9, pp. 10127–10139, Sep. 2020.

[23] J. Wang, W. Mu, Y. Liu, L. Guo, S. Zhang, and G. Gui, "Deep reinforcement learning-based satellite handover scheme for satellite communications," in *2021 13th Int. Conf. Wireless Commun. Signal Process. (WCSP)*, 2021, pp. 1–6.
[24] H. Liu, Y. Wang, and Y. Wang, "A successive deep Q-learning based distributed handover scheme for large-scale LEO satellite networks," in *2022 IEEE 95th Veh. Technol. Conf.: (VTC2022-Spring)*, 2022, pp. 1–6.
[25] Y. Cao, S.-Y. Lien, and Y.-C. Liang, "Deep reinforcement learning for multi-user access control in non-terrestrial networks," *IEEE Trans. Commun.*, vol. 69, no. 3, pp. 1605–1619, Mar. 2021.
[26] N. Badini, M. Marchese, M. Jaber, and F. Patrone, "Reinforcement learning-based load balancing satellite handover using NS-3," in *2023 Int. Conf. Commun. (ICC)*, 2023, pp. 1–6.
[27] T. Darwish, G. K. Kurt, H. Yanikomeroglu, M. Bellemare, and G. Lamontagne, "LEO satellites in 5G and beyond networks: A review from a standardization perspective," *IEEE Access*, vol. 10, pp. 35040–35060, 2022.
[28] 3GPP, "Study on new radio (NR) to support non-terrestrial networks," Tech. Rep. TR38.811(2020), 2020.
[29] J. Mao, L. Zhang, P. Xiao, and K. Nikitopoulos, "Interference analysis and power allocation in the presence of mixed numerologies," *IEEE Trans. Wireless Commun.*, vol. 19, no. 8, pp. 5188–5203, Aug. 2020.
[30] N. Badini, M. Marchese, and F. Patrone, "NS-3-based 5G satellite-terrestrial integrated network simulator," in *2022 21st Mediterranean Electrotechnical Conf. (MELECON)*, 2022, pp. 1–6.
[31] 3GPP, "Service requirements for the 5G system," *Eur. Telecommun. Standards Inst.*, Sophia Antipolis Cedex, France, Tech. Rep. ETSI TS 122 261 V15.9.0, Sep. 2022.
[32] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 2018.
[33] G. A. Rummery and M. Niranjan, "On-line Q-learning using connectionist systems," Dept. Eng., Univ. Cambridge, Cambridge, U.K., Tech. Rep. CUED/F-INFENG/TR 166, 1994, vol. 37.
[34] A. Warrier, S. Al-Rubaye, D. Panagiotakopoulos, G. Inalhan, and A. Tsourdos, "Interference mitigation for 5G-Connected UAV using deep Q-learning framework," in *2022 41st Digit. Avionics Syst. Conf. (DASC)*, 2022, pp. 1–8.
[35] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.
[36] N. C. Luong et al., "Applications of deep reinforcement learning in communications and networking: A survey," *IEEE Commun. Surveys Tut.*, vol. 21, no. 4, pp. 3133–3174, Fourthquarter 2019.
[37] [Online]. Available: https://www.nsnam.org/
[38] D. Vallado, P. Crawford, R. Hujsak, and T. Kelso, "Revisiting spacetrack report#3," in *Proc. AIAA/AAS Astrodynamics Specialist Conf. Exhibit*, 2006, Art. no. 6753.

Nour Badini (Member, IEEE) received the master's degree in computer and communication engineering from Lebanese University, Beirut, Lebanon, in 2020, and the Ph.D. degree in science and technology for electronic and telecommunication engineering from the University of Genoa, Genoa, Italy, in 2023.

She is currently Postdoctoral Researcher with the Satellite Communications and Heterogeneous Networking Laboratory, University of Genoa. Her main research interests include the integration of satellites into 5G and beyond communication networks, routing and handover management in nonterrestrial networks, quality of service over satellite communications, and exploiting machine learning-based techniques for performance enhancement in heterogeneous networks.

Mona Jaber (Senior Member, IEEE) received the Master's degree in electrical, electronics, and communications engineering from the American University of Beirut, in 2014 and the Ph.D. degree in electronic engineering from the University of Surrey, in 2017.

She is a Senior Lecturer in IoT with the School of Electronic Engineering and Computer Science, Queen Mary University of London (QMUL), London, U.K. She is the Director of the "Digital Twins for Sustainable Development Goals" Research Lab, QMUL, where she attracted the first multidisciplinary core team to further studies in this area. She has authored or coauthored in the areas of sustainable energy, smart mobility, and privacy-preserving e-health. As part of her industry research collaboration efforts, she has established a ground-breaking project that uses optical fiber systems for the detection and classification of active travel—a robust, scalable, and privacy-preserving method that informs smart city and transportation planning. Her research interests include zero-touch networks, the intersection of ML and IoT in the context of sustainable development goals, and IoT-driven digital twins.

Ms. Jaber was the recipient of the N2Women Rising Star in computer networking and communications in 2022. She is also a Steering Committee Member of IEEE Women in Engineering.

Mario Marchese (Senior Member, IEEE), was born in Genoa, Italy, in 1967. He received the Laurea (cum laude) degree in ingegneria elettronica, and the Ph.D. (Italian "Dottorato di Ricerca") degree in telecommunications from the University of Genoa, Genoa, Italy, in 1992 and 1997, respectively.

From 1999 to 2005, he was with the Italian Consortium of Telecommunications (CNIT), University of Genoa Research Unit, where he was the Head of Research. From 2005 to 2016, he was an Associate Professor with the University of Genoa, where since 2016, he has been a Full Professor. He is the author of the book "*Quality of Service over Heterogeneous Networks*" (John Wiley & Sons, Chichester, 2007), and author/coauthor of more than 300 scientific works, including international magazines, international conferences, and book chapters. His main research interests include networking, quality of service over heterogeneous networks, software defined networking, satellite DTN and nanosatellite networks, and networking security.

Dr. Marchese was the Chair of IEEE Satellite and Space Communications Technical Committee from 2006 to 2008. He was the Winner of the IEEE ComSoc Award "2008 Satellite Communications Distinguished Service Award" in "recognition of significant professional standing and contribution' in the field of satellite communications technology."

Fabio Patrone (Member, IEEE) received the Bachelor's and Master's degrees in telecommunications engineering, from the University of Genoa, Italy, in 2010 and 2013, respectively, and the Ph.D. degree in science and technology for electronic and telecommunication engineering from the University of Genoa, Italy, in 2016.

He is currently an Assistant Professor with the University of Genoa, Genoa, Italy, where he is doing his research activity with the Satellite Communications and Heterogeneous Networking Laboratory (SCNL). His research interests include routing, handover, scheduling, and congestion control algorithms in nonterrestrial networks, the study and development of machine learning (ML)-based techniques for cybersecurity solutions, such as ML-based intrusion detection systems, and the employment of networking technologies, such as network function virtualization (NFV) and software defined networking (SDN), for the integration of nonterrestrial networks with the terrestrial infrastructure within B5G/6G.

Open Access provided by 'Universitàdegli Studi di Genova' within the CRUI CARE Agreement