Multi-Agent Fingerprints-Enhanced Distributed Intelligent Handover Algorithm in LEO Satellite Networks

Feng Yang, *Graduate Student Member, IEEE*, Wenjun Wu, *Member, IEEE*, Yang Gao, *Graduate Student Member, IEEE*, Yang Sun, *Member, IEEE*, Teng Sun, and Pengbo Si, *Senior Member, IEEE*

***Abstract*—The next-generation wireless network is expected to use low-earth orbit (LEO) satellite networks to deliver seamless and high-capacity global communications services. Due to the high-speed mobility of LEO satellites, massive and frequent handovers inevitably occur. Moreover, handover becomes more complicated with the ever-growing constellation scale, number of mobile terminals (MTs), and demands for emerging delay-sensitive applications. In this paper, a decentralized Markov decision process (DEC-MDP) is adopted to formulate the handover problem in the LEO satellite network with finite bursty traffic. The target is maximizing the total reward associated with the service revenue and the cost of handover and packet loss. To deal with the high computational complexity caused by the large state space and action space, the solution is designed using a multi-agent double deep Q-network (MADDQN) with fully decentralized framework, which also allows each MT to train and use an individual local DDQN to avoid load imbalance between satellites. Further, to alleviate the non-stationary issue of the environment in parallel learning, multi-agent fingerprints are applied in MADDQN, and the proposed algorithm is called multi-agent fingerprints-enhanced double deep Q-network-based distributed intelligent handover (MAF-DDQN-DIH) mechanism. The implementation of MAF-DDQN-DIH in practical communication systems are discussed, and the corresponding communication overhead and computational complexity are analyzed. Simulation results demonstrate that the designed multi-agent fingerprints are effective and the proposed MAF-DDQN-DIH algorithm outperforms the comparison handover algorithms in terms of the total reward.**

***Index Terms*—LEO satellite networks, handover, decentralized Markov decision process, multi-agent double deep Q-network, multi-agent fingerprints.**

I. Introduction

With the rapid increase of mobile terminals (MTs) and applications, terrestrial wireless networks need to satisfy the ever-growing demands for ubiquitous and seamless services in remote, rural, and urban areas [1]. Massively deploying terrestrial base stations (BSs) may not be an effective remedy because their deployment is difficult [2]. Taking advantage of high altitudes, broad operating spectrum, and ultra-dense topology, the low-earth orbit (LEO) satellite network, in which LEO satellites with on-board BSs serve as the expansion of terrestrial wireless networks to provide seamless and high-capacity global communications services, has been widely acknowledged as a potential solution [3], [4].

However, each LEO satellite can only provide services to MTs for a short time due to its continuous and fast movement [5]. Thus, MTs may need to switch to different visible LEO satellites more than once during a call to maintain uninterrupted connections. Moreover, with the rapidly increasing numbers of deployed satellites and MTs, and the ever-growing demands for various applications [6], e.g., Internet access, extended reality, and telemedicine services, the handover problem becomes more challenging compared with traditional LEO satellite networks.

Inspired by terrestrial networks, some researchers have proposed event-triggered handover schemes based on measurement events or some new handover triggering events and additional triggering criteria. A potential handover procedure in LEO satellite networks is the conditional handover, which was originally developed for terrestrial wireless networks to improve mobility robustness by executing handover decisions in advance [7]. It was proposed to use the distance between the MT and the neighboring satellites as a location-based handover triggering event [8]. A handover strategy based on antenna gain that takes advantage of the predictability of the satellite movements as well as the antenna gain of the satellite beams was developed in [9]. The performance of the traditional signal measurement-based handover triggering mechanism was compared with that of distance, elevation angle, and timer-based mechanisms [10]. Besides, a handover threshold determination method based on a reconfigurable factor graph, which includes the received signal strength indication, location, elevation angle, ephemeris, and other factors, was also proposed [11]. However, the above-mentioned studies only consider seamless call service with the objective to reduce the handover failures, unnecessary handovers, ping-pong rates and radio link failures. Therefore, these studies are unable to evaluate the performance indicators

Manuscript received 2 October 2023; revised 27 March 2024; accepted 3 June 2024. Date of publication 11 June 2024; date of current version 17 October 2024. This work was supported in part by the Natural Science Foundation of Beijing Municipality under Grant L212003 and in part by the National Natural Science Foundation of China under Grant U2233217 and Grant 62371029. The review of this article was coordinated by Prof. Giovanni GG Giambene. *(Corresponding author: Wenjun Wu.)*

Feng Yang, Wenjun Wu, Yang Gao, Yang Sun, and Pengbo Si are with the Faculty of Information Technology, Beijing University of Technology, Beijing 100124, China (e-mail: yangfeng@emails.bjut.edu.cn; wenjunwu@bjut.edu.cn; cathyshou@emails.bjut.edu.cn; sunyang@bjut.edu.cn; sipengbo@bjut.edu.cn).

Teng Sun is with the 54th Research Institute of China Electronics Technology Group Corporation, Shijiazhuang 050081, China (e-mail: 375663046@qq.com).

Digital Object Identifier 10.1109/TVT.2024.3412287

such as packet delay and packet loss rate for numerous emerging delay-sensitive applications such as autonomous vehicular systems, augmented reality, and virtual reality [6]. Further, as the status of application data queues and the delay requirements of applications have not been considered in these handover schemes, their performance is also uncertain.

Considering a LEO satellite network which provides Internet access service for MTs, a MT-centric handover scheme was proposed, in which MT's downlink data is buffered in multiple satellites and the MT always accesses the satellite with the best link quality [12]. But this scheme may not scale to the scenario with massive MTs due to a significant number of MTs in the same area may connect to the same satellite with the best link quality which leads to serious access congestion on a certain satellite.

Graph theory has been utilized to develop the LEO satellite handover methods due to a handover procedure that can be described as choosing an optimal path from the available handover paths. A network-flows graph, which considers the requests of MTs and the quality of service (QoS) provided by satellites, was formulated for the issue of satellite handover, and the appropriate satellites can be chosen by MTs based on the flow matrix [5]. Further taking into account the coverage time, elevation angle, and free channel state, a dynamic graph was given to represent all the possible handover paths, and then, an optimal handover path can be carefully chosen by Floyd algorithm [13]. Moreover, an access graph model with satisfaction weight and load weight was established, and a greedy method was adopted to solve the shortest handover path [14]. However, these handover strategies are centralized, and handover decisions must be made according to global state information. As a result of the numerous satellites and MTs in the LEO network, the decision-making complexity and communication overhead are considerable. Besides, those simple QoS criteria used in [13] and [14] may still not meet the performance requirements of emerging applications.

As most of the studied LEO satellite handover problems are NP-hard and may have many local extrema [15], evolutionary algorithms have been exploited to design the handover algorithms for LEO satellite networks. The handover was modeled as a multi-objective optimization problem with the purpose of balancing load and maximizing throughput under the constraint of handover delay, and then an improved discrete binary particle swarm optimization-based intelligent handover scheme was developed to choose the optimal network [15]. A utility function was designed, which consists of the elevation angle, remaining visible time, handover response time, and handover request time; then, a handover scheme based on the potential game was proposed to maximize the MTs' utilities [16]. Owing to the dynamic characteristics of LEO satellite networks, the computational complexity of the evolutionary algorithm could be too high.

Reinforcement learning [17] is a promising method to solve handover problems under complex network environments [18], as it can significantly reduce the computational complexity during handover by making decisions using a well-trained agent. A distributed handover mechanism using multi-agent Q-learning was developed in [19] to reduce the average handover cost while meeting the load constraint of each satellite. However, because of the limited number of state-action pairs that can be stored in the Q-table, the proposed algorithm is difficult to scale to scenarios with massive satellites. Fortunately, deep reinforcement learning is an excellent alternative to address the decision-making issue with the huge state and action spaces [20]. Considering the independence between satellites and the limited number of satellites that can serve each MT, a low-complexity handover mechanism based on the deep Q-network was proposed in [21]. However, the optimization objective does not consider the number of handovers, which is a key performance indicator in the LEO satellite handover problem. Moreover, due to the full-buffer traffic is used in [21], the delay which is an import performance indicator for most of the future applications in LEO satellite networks also does not considered.

In this paper, the issues of the delay requirements of emerging applications, as well as the communication overhead and decision-making complexity of the handover algorithms in the LEO satellite network with massive satellites and MTs are the main concerns. The handover issue is investigated in the LEO satellite network with finite bursty traffic [22]. The distributed optimization algorithm based on the multi-agent double deep Q-network (MADDQN) with multi-agent fingerprints is developed to obtain handover decisions with low communication overhead and low decision-making complexity. The major contributions of this paper are as follows:

1) *The finite bursty traffic is used to model the data of emerging applications and to realize the evaluation of delay and packet loss.* Most of the existing research on handover algorithms in LEO satellite networks uses the full buffer traffic model to evaluate mobility management performance like handover times and system throughput under extreme conditions. With the growing demand for delay-sensitive applications, handover also needs to consider performance metrics such as delay and packet loss. Therefore, the finite bursty traffic model which can be used to evaluate these performance metrics is more necessary when studying the handover issue in LEO satellite networks.
2) *The fully decentralized handover decision-making framework is designed to reduce the overhead and complexity and to avoid load imbalance between satellites.* With the increasing numbers of satellites and MTs in LEO satellite networks, the centralized handover schemes which rely on the global system information will bring severe issues of communication overhead and decision-making complexity. Moreover, due to the high altitude and wide coverage of LEO satellites, multiple MTs in the same area may have similar visible LEO satellites and channel quality. If these MTs share a common double deep Q-network (DDQN) in the decision process, they tend to connect to the same satellite as the states input to their DDQN-based agents are similar, which results in a highly imbalanced satellite load. To encourage diverse individualized behaviors and reduce overhead and complexity, the decentralized framework, which enables each MT to use and train an individual local DDQN, is a more economical and effective alternative.

3) *The multi-agent fingerprints are used to alleviate the non-stationary problem of the environment caused by parallel learning.* When the fully decentralized handover decision-making framework is adopted, all the MTs are learning in parallel, and their decision policies are changing in the training process. Thus, from the perspective of each MT, the environment becomes non-stationary due to the changing policies of other MTs. To address this problem, a feasible approach is to extend the local state of an MT using multi-agent fingerprints that can reflect the policies of other MTs [23]. Inspired by this idea, the handover actions of other MTs, which are directly related to the policies of these MTs, are designed as multi-agent fingerprints to enhance the performance.
4) *The feasibility of applying the proposed multi-agent fingerprints-enhanced DDQN-based distributed intelligent handover (MAF-DDQN-DIH) algorithm to a practical LEO satellite network is discussed.* Due to the limited research on fully decentralized framework in which MTs decide the inter-satellite handovers based on local state, how to implement the MAF-DDQN-DIH algorithm proposed in this paper in practical LEO satellite networks is an important issue. Therefore, the handover procedures supporting MAF-DDQN-DIH algorithm are designed, and the corresponding communication overhead is analyzed. Furthermore, considering the scalability of the algorithm, the computational complexity of MAF-DDQN-DIH is analyzed too.

The following is how the rest of this article is organized. Section II introduces the system model under consideration. Section III formulates the handover problem as a decentralized Markov decision process (DEC-MDP) model. Section IV proposes the MAF-DDQN-DIH mechanism. Section V and Section VI give the implementations of the MAF-DDQN-DIH algorithm and the simulation results, respectively. Finally, Section VII concludes this paper.

II. System Model

The considered LEO satellite network model, traffic model, and handover model are provided in this section.

A. LEO Satellite Network Model

In this paper, the constellation of OneWeb which consists of 648 LEO satellites is considered. These LEO satellites are distributed in 18 orbital planes with an altitude of 1200 km and an inclination of 87.9°. Each orbital plane deploys approximately 40 satellites, with a separation of 9° between adjacent orbital planes. Generally, a terrestrial area is covered by multiple LEO satellites. The mobile terminals (MTs) are randomly distributed in a terrestrial area, and each MT can communicate with only one visible satellite at any given time. Assume the classic round-robin scheduling is adopted by LEO satellites, and thus, all the MTs connecting to the same LEO satellite share the limited communication resources in a fair way.

Because of the extremely dynamic nature of LEO satellite networks, the satellite handover problem in a snapshot lasting for a specific time period is considered [19]. Fig. 1 provides a simplified diagram of a snapshot of the LEO satellite network. It is comprised of ground stations (GSs), N LEO satellites, and K randomly located MTs. We denote the sets of LEO satellites and MTs by $\mathcal{B} = \{b_i\}_{i=1,\ldots,N}$ and $\mathcal{U} = \{u_j\}_{j=1,\ldots,K}$ respectively. Furthermore, assuming that the LEO satellite network runs in synchronous, equal-length time steps [24], the whole considered period is slotted into T time steps, which are indexed by $t = 1, \ldots, T$, and the length of a time step is Δt.

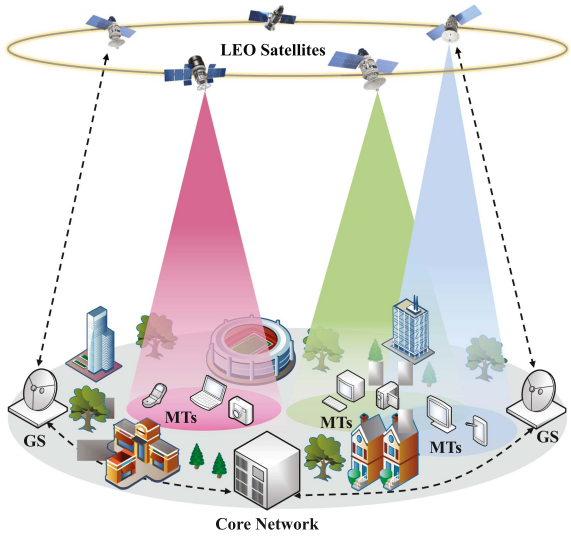


Fig. 1. Architecture of the LEO satellite network.

It is assumed that MTs are stationary since LEO satellites move at a significantly faster speed than the MTs [21]. It also should be noted that each of the N LEO satellites is only visible for a portion of the time period under consideration because of the continuous fast movement of LEO satellites. The elevation angle constraint that b_i can provide services to u_j are given by [19]

$$\theta_{i,j} \geq \theta_0, \forall b_i \in \mathcal{B}, \forall u_j \in \mathcal{U}, \tag{1}$$

where $\theta_{i,j}$ and θ_0 denote the elevation angle between b_i and u_j and the minimum elevation angle, respectively. Besides, the visible LEO satellites of a MT at different time steps may vary because of the ongoing movement of LEO satellites. $\mathcal{B}_{j,t}$ denotes the set of LEO satellites that were visible to u_j at time step t.

Assuming that different LEO satellites covering the same region operate on different frequency bands, co-channel interference is ignored. Therefore, the link quality between an LEO satellite and a MT is indicated by the carrier-to-noise ratio (CNR). The locations of MTs and LEO satellites, as well as the CNRs between MTs and LEO satellites, are constant during each time step [19], [24].

For u_j, the CNR when associated with b_i can expressed as

$$\gamma_{i,j}(t) = \frac{P_{\mathrm{TX}} G_{\mathrm{TX}} G_{\mathrm{RX}} L_{\mathrm{TX}} L_{i,j}(t) h_{i,j}(t)}{\Theta F_{\mathrm{RX}} W_{\mathrm{RF}}}, \tag{2}$$

where P_{TX}, G_{TX}, and L_{TX} denote the transmit power, the antenna gain of the transmitter, and the feeder loss of the transmitter, respectively, G_{RX} and F_{RX} denote the antenna gain

of the receiver and the receiver equivalent noise temperature, respectively, $L_{i,j}(t)$ denotes the path loss between b_i and u_j during time step t, Θ denotes the Boltzmann constant, W_{RF} denotes the reference bandwidth, and $h_{i,j}(t)$ denotes the small-scale channel power gain between b_i and u_j during time step t. We assume $h_{i,j}(t)$ follows the Rician fading model with a Rician factor Φ [12], where Φ is the ratio of power between the line-of-sight (LOS) and non-line-of-sight (NLOS) components. Moreover, similar to that in [21], $h_{i,j}(t)$ remains unchanged during each time step but varies from one time step to another as the space-terrestrial transmission environment between satellites and MTs changes slowly during each time step. Considering that the dynamic scheduling process, which occurs once per time slot, is a decision-making process on a smaller time scale compared to the handover process, and the objective of this paper is to design a distributed intelligent handover algorithm to enable effective handover decisions for MTs, the classic and basic round-robin (RR) scheduling algorithm is employed. It is worth noting that while a more advanced dynamic scheduling algorithm may better improve system performance, it does not affect the effectiveness of the proposed handover algorithm. As a result of adopting the RR scheduling mechanism, all LEO satellites distribute bandwidth resources equally to their service MTs [25]. Therefore, the downlink transmission rate of u_j from b_i during time step t can be achieved as

$$c_{i,j}(t) = \frac{W}{M_i(t)} \log_2(1 + \gamma_{i,j}(t)), \tag{3}$$

where W denotes the bandwith of b_i, and $M_i(t)$ denotes the number of active MTs connected to b_i during time step t.

B. Traffic Model

Since the delay-sensitive applications will account for a significant proportion in LEO satellite networks in the future, the finite bursty traffic model [22] is adopted to facilitate the evaluation of latency related service performance. Specifically, a Poisson process is used to model the packet arrivals of u_j, and the arrival rate is set as λ [22]. Two parameters are used to describe each packet of u_j. These parameters are [22]:

- q, which is the packet size.
- T_{drop}, which is the maximum tolerable delay. If a packet is not delivered in entirely within T_{drop} time, it will be dropped.

For the traffic flow of each MT, the First-In-First-Out (FIFO) service strategy is adopted. Thus, the delay of each packet is composed of the transmitting delay and queueing delay. Specifically, when u_j associates with b_i at time step t, the transmission process of u_j's traffic flow is demonstrated in Fig. 2. When the delay of a packet in the transmission queue or a packet being transmitted is greater than T_{drop}, the packet will be dropped.

C. Handover Model

In light of the constantly changing network topology of LEO satellite networks, the centralized handover schemes are usually dominated by GSs [5]. However, when large-scale constellations of LEO satellite networks are used and massive MTs are served,

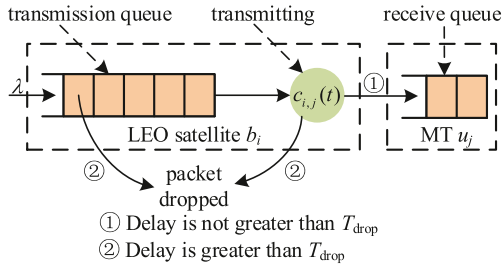


Fig. 2. Transmission process of u_j's traffic flow.

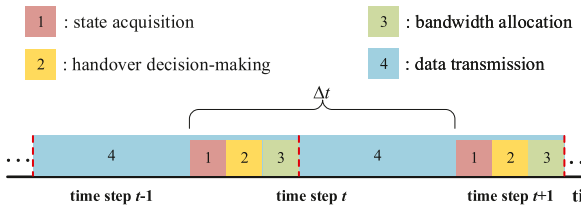


Fig. 3. Timeframe of the distributed handover decision-making process.

the communication overhead for the GSs obtaining the global network information will be significant, and the complex centralized handover decisions put heavy burdens on the GSs. Therefore, a distributed handover-making decision process with low communication overhead and low decision-making complexity is designed. Further, due to the high altitude and wide coverage of LEO satellites, multiple MTs in the same area may have similar visible LEO satellites and channel quality. If these MTs make handover decisions based on the similar system state and the same handoff strategy, they tend to connect to the same satellite, which results in a highly imbalanced satellite load. Thus, the personalized local handover strategies are necessary.

As shown in Fig. 3, the timeframe of our proposed distributed handover decision-making process is mainly composed of four phases, as follows:

1) State Acquisition Phase: Each active MT acquires the local state through measurement and broadcast information.

2) Handover Decision-Making Phase: Each active MT decides on a handover independently according to the local state and its personalized local handover strategy.

3) Bandwidth Allocation Phase: Each LEO satellite allocates bandwidth resources to the active MTs it serves, and the downlink transmission rate between each LEO satellite and each active MT it serves is estimated by (3).

4) Data Transmission Phase: Each LEO satellite sends packets to its serving active MTs. To maintain the continuity of services, the data transmission phase will last until the next data transmission phase starts.

III. DEC-MDP Formulation for the Handover Problem

Similar to our previous work [18], a DEC-MDP model is used to formulate the handover problem. Considering the delay requirement of the emerging applications in future LEO satellite networks, the total reward includes the cost of packet loss in addition to the service revenue and the cost of handover. The DEC-MDP model is expressed mathematically using a tuple

$(\mathcal{U}, \mathcal{S}, \mathcal{A}, P, R)$ [26], where $\mathcal{U}$, $\mathcal{S}$, $\mathcal{A}$, P, and R denote the agent set, the environment state space, the joint action space, the transition function, and the joint reward function, respectively. Following that are detailed descriptions of the state space, action space, reward function, and state transition.

A. State Space

The environment state space $\mathcal{S}$ can be denoted by $\mathcal{S} = \mathcal{S}_1 \times \ldots \times \mathcal{S}_j \times \ldots \times \mathcal{S}_K$, where $\mathcal{S}_j$ denotes the local state space of u_j. During time step t, the environment state is indicated as $\boldsymbol{s}_t = (\boldsymbol{s}_{1,t}, \ldots, \boldsymbol{s}_{j,t}, \ldots, \boldsymbol{s}_{K,t}) \in \mathcal{S}$, and the local state $\boldsymbol{s}_{j,t} \in \mathcal{S}_j$ of u_j is indicated as

$$\begin{aligned} s_{j,t} = [&\tilde{c}_{1,j,t}, \ldots, \tilde{c}_{i,j,t}, \ldots, \tilde{c}_{N,j,t}, \\ &\tilde{M}_1(t), \ldots, \tilde{M}_i(t), \ldots, \tilde{M}_N(t), \\ &v_{1,j,t}, \ldots, v_{i,j,t}, \ldots, v_{N,j,t}, \\ &I_{j,t}, \alpha_{j,t}, \beta_{j,t}, d_{j,t}]^{\mathrm{T}}, \end{aligned} \tag{4}$$

where $\tilde{c}_{i,j,t}$ denotes the achievable transmission rate between u_j and b_i on one resource block (RB), $\tilde{M}_i(t)$ denotes the number of active MTs currently being served by b_i, $v_{i,j,t}$ denotes the remaining visible time of b_i to u_j, $I_{j,t}$ denotes the index of the LEO satellite currently associated with u_j, $\alpha_{j,t}$ denotes the number of packets in u_j's transmission queue, $\beta_{j,t}$ and $d_{j,t}$ respectively denote the remaining data volume to be transmitted and the remaining time before it is discarded for the first packet in u_j's transmission queue. When $b_i \notin \mathcal{B}_{j,t}$, its corresponding elements $\tilde{c}_{i,j,t}$ and $v_{i,j,t}$ in $\boldsymbol{s}_{j,t}$ are all set to 0. Therefore, the sizes of the local state spaces at different time steps are the same, which facilitates the design of the solution.

B. Action Space

The joint action space $\mathcal{A}$ is defined as $\mathcal{A} = \mathcal{A}_1 \times \cdots \times \mathcal{A}_j \times \cdots \times \mathcal{A}_K$, where $\mathcal{A}_j$ denotes the action space of u_j. $\boldsymbol{a}_t = (a_{1,t}, \ldots, a_{j,t}, \ldots, a_{K,t}) \in \mathcal{A}$ is the joint action during time step t, where $a_{j,t} \in \mathcal{A}_j$ is the action executed by u_j. According to the handover model in Section II-C, during time step t, each active MT needs to choose an LEO satellite at the handover decision phase. If u_j selects the LEO satellite b_i, $a_{j,t} = b_i$. To ensure that u_j has the same size of action space at different time steps, we set the action space $\mathcal{A}_j$ as the LEO satellite set $\mathcal{B}$. However, u_j selects an LEO satellite from the visible LEO satellite set $\mathcal{B}_{j,t} \subseteq \mathcal{A}_j$ when making a handover decision in time step t, which can reduce some unnecessary exploration.

C. Reward Design

The joint reward function R can be expressed as $R = R_1 \times \ldots \times R_j \times \ldots \times R_K$, where R_j denotes the reward function of u_j. $\boldsymbol{r}_t = R(\boldsymbol{s}_t, \boldsymbol{a}_t) = (r_{1,t}, \ldots, r_{j,t}, \ldots, r_{K,t})$ is the joint reward obtained after conducting the joint action $\boldsymbol{a}_t$ in the environment state $\boldsymbol{s}_t$, where $r_{j,t}$ is the reward obtained by u_j. For the LEO satellite handover problem, the number of handovers is the main performance metric [16]. For the finite bursty traffic modeling emerging application, the volume of data received successfully within required delay limitation and the number of packets dropped are two main performance metrics [22]. Therefore, three kinds of utilities are taken into account to reflect the above three performance metrics, namely the service revenue, the cost of handover, and the cost of packet loss. Locally, we define the reward function of u_j after taking joint action $\boldsymbol{a}_t$ at global state $\boldsymbol{s}_t$ as

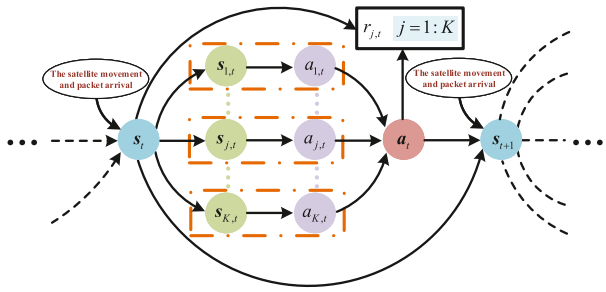


Fig. 4. State transition diagram of the DEC-MDP.

$$r_{j,t} = R_j(\boldsymbol{s}_t, \boldsymbol{a}_t) = \mu_1 \frac{n_{j,t}}{q} f_1 + \mu_2 \delta_{j,t} f_2 + \mu_3 \eta_{j,t} f_3, \tag{5}$$

where μ_1, μ_2 and μ_3 denote the weights of the service revenue, the cost of handover and the cost of packet loss respectively, $n_{j,t}$ denotes the volume of data that u_j successfully received at time step t, $\delta_{j,t}$ is a dummy variable to indicate whether u_j has switched at time step t or not: $\delta_{j,t} = 1$ yes and 0 otherwise, $\eta_{j,t}$ is the number of dropped packets for u_j during time step t, and f_1, f_2, and f_3 denote the scores obtained by u_j after successfully receiving a packet, after completing an inter-satellite handover, and after discarding a packet, respectively.

D. State Transition

The state transition diagram of the DEC-MDP is demonstrated in Fig. 4. Based on the handover model in Section II-C, during the state acquisition phase, u_j obtains its packet queue information and current connection, as well as the estimated transmission rate and remaining visible time with each satellite, and then forms the local state $\boldsymbol{s}_{j,t}$. After that, during the handover decision-making phase, the local action $a_{j,t}$ is generated based on the local state $\boldsymbol{s}_{j,t}$. When the bandwidth allocation phase and the data transmission phase are completed, the joint reward $\boldsymbol{r}_t$ can be calculated. In addition, the environment state changes to $\boldsymbol{s}_{t+1}$ which depends on the previous environment state $\boldsymbol{s}_t$, the joint action $\boldsymbol{a}_t$, the movements of all LEO satellites, and the packet arrivals of all MTs. Note that for the inactive MT $u_{j'}$ in the time step t, its corresponding $\boldsymbol{s}_{j',t}$, $a_{j',t}$, and $r_{j',t}$ are all None.

According to the aforementioned definitions, the optimization objective of the DEC-MDP is to achieve the maximum total reward, which can be stated as

$$C = \sum_{t=1}^{T} \sum_{u_j \in \mathcal{U}} r_{j,t}. \tag{6}$$

IV. Multi-Agent Fingerprints-Enhanced Distributed Intelligent Handover Mechanism

Due to the complex state transition probability and huge joint state-action space in the DEC-MDP, it is a nondeterministic exponential time (NEXP)-complete problem to mathematically achieve the decision-making strategy with optimality guarantees [27]. Therefore, multi-agent deep reinforcement learning is the most capable method to approximately solve such types of issues [28], [29], [30].

A. Framework of the Distributed Intelligent Handover Mechanism

The proposed distributed intelligent handover mechanism is named MAF-DDQN-DIH mechanism, which uses DDQN for local handover decision-making.

As defined by the DEC-MDP model, the local state space, action space, and reward function are the same for all agents. Thus, it is possible to train all agents more efficiently using the parameter sharing method [31], and all agents can share a common DDQN model. However, in LEO satellite networks, due to the high altitude and wide coverage of LEO satellites, multiple agents in the same area may have similar visible LEO satellites and channel quality, and thus, they obtain the similar local states. When parameters are shared, multiple agents will be associated with the same satellite because they typically adopt similar actions under similar local states, which results in a highly imbalanced satellite load. Moreover, due to the need for frequent parameter synchronization and data exchange, the communication overhead of this method is considerable.

To encourage diverse individualized behaviors, MAF-DDQN-DIH algorithm adopts a fully decentralized framework as illustrated Fig. 5. In this framework, each agent makes handover decisions based on a individual DDQN, and then trains its own DDQN using its own transitions.

B. Multi-Agent Fingerprints

In the fully decentralized handover decision-making framework, the environment becomes non-stationary for each agent due to the changing policies of other agents in the training process. This can be observed from the Bellman equation of agent u_j given the policies $\boldsymbol{\pi}_{-j,t} = (\pi_{1,t}, \ldots, \pi_{j-1,t}, \pi_{j+1,t}, \ldots, \pi_{K,t})$ of all other agents [23]:

$$\begin{aligned} &Q_j^*\left(\boldsymbol{s}_{j,t}, a_{j,t} \,|\boldsymbol{\pi}_{-j,t}\right) \\ &= \sum_{\boldsymbol{a}_{-j,t}} \prod_{j' \in -j} \pi_{j',t}\left(a_{j',t} \,|\boldsymbol{s}_{j',t}\right)[R_j\left(\boldsymbol{s}_{j,t}, a_{j,t}, \boldsymbol{a}_{-j,t}\right) \\ &+ \gamma \sum_{\boldsymbol{s}_{j,t+1}} P\left(\boldsymbol{s}_{j,t+1} \,|\boldsymbol{s}_{j,t}, a_{j,t}, \boldsymbol{a}_{-j,t}\right) \max_{a_{j,t+1}} Q_j^*\left(\boldsymbol{s}_{j,t+1}, a_{j,t+1}\right)]. \end{aligned} \tag{7}$$

where $\gamma \in (0, 1]$ is a constant. The nonstationary component is $\prod_{j' \in -j} \pi_{j',t}(a_{j',t}|s_{j',t})$, which changes as the other agents' policies change over time in the training process. Therefore, incorporating the policies of other agents into the state is a naive method for addressing the non-stationarity issue of the environment [32]. However, the policies of other agents are composed of high-dimensional neural networks. If the neural network parameters of other agents are directly augmented to the local state of each agent, the local state space will be very large and infeasible to learn.

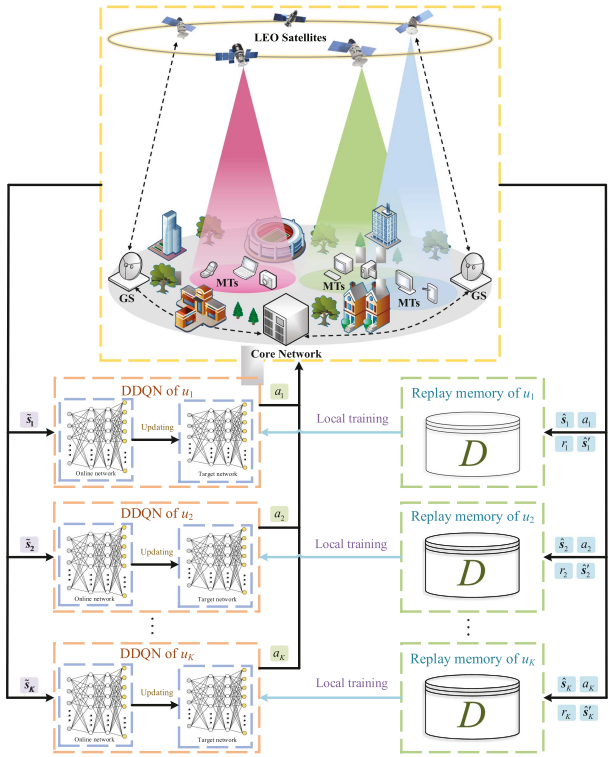


Fig. 5. Framework of the MAF-DDQN-DIH scheme.

Instead, incorporating fingerprints related to the policies of other agents into the state is a feasible method. Considering that the actions taken by other agents are directly related to their policies, we use the actions $\boldsymbol{a}_{-j,t} = (a_{1,t}, \ldots, a_{j-1,t}, a_{j+1,t}, \ldots, a_{K,t})$ taken by other agents as fingerprints. Since inactive MT $u_{j'}$ only selects to connect to the currently serving satellite and is not allocated any bandwidth resources, the action $a_{j',t}$ of $u_{j'}$ will not affect u_j. Therefore, the multi-agent fingerprints actually used by u_j is $\hat{\boldsymbol{a}}_{-j,t} = (\hat{a}_{1,t}, \ldots, \hat{a}_{j-1,t}, \hat{a}_{j+1,t}, \ldots, \hat{a}_{K,t})$, where $\hat{a}_{j',t} = -1$ when $u_{j'}$ is an inactive MT, indicating the inactive state, otherwise $\hat{a}_{j',t} = a_{j',t}$.

In the training process, the local training samples can be collected to the local replay memory with a delay of two time steps. Specifically, at the beginning of time step t, each LEO satellite b_i broadcasts a message to all the MTs in its coverage area, which includes the index set of all the MTs it currently serves (i.e. the set of all MTs that choose b_i as the action in time step $t-1$) and a vector indicating whether these MTs are active. Based on the broadcast messages at time step $t-1$, u_j can obtain the set of other active MTs at time step $t-1$. Then, based on the broadcast messages at time step t, u_j can obtain the actions taken by these other active MTs at time step $t-1$, that is, the multi-agent fingerprints of u_j at time step $t-1$. Thus, u_j can obtain its multi-agent fingerprints $\hat{\boldsymbol{a}}_{-j,t-2}$ and $\hat{\boldsymbol{a}}_{-j,t-1}$, respectively, at

time step $t-1$ and t. Then, at time step t, u_j store its local training sample $(\hat{\boldsymbol{s}}_{j,t-2}, a_{j,t-2}, r_{j,t-2}, \hat{\boldsymbol{s}}_{j,t-1})$ from time step $t-2$ in its local replay memory, where $\hat{\boldsymbol{s}}_{j,t-2} = \{\boldsymbol{s}_{j,t-2}, \hat{\boldsymbol{a}}_{-j,t-2}\}$ and $\hat{\boldsymbol{s}}_{j,t-1} = \{\boldsymbol{s}_{j,t-1}, \hat{\boldsymbol{a}}_{-j,t-1}\}$. This delayed training sample storage method does not affect the random mini-batch selection from the local replay memory for local model training.

In the local handover decision-making phase, the state input to the local agent must be obtained in real time. Because all agents make decisions simultaneously, u_j can only obtain the previous actions $\boldsymbol{a}_{-j,t-1}$ of other agents while it is making a decision in practice. Thus, the actions $\boldsymbol{a}_{-j,t-1}$ taken by other agents in the previous time step are used as the approximate fingerprints in the state $\tilde{\boldsymbol{s}}_{j,t} = \{\boldsymbol{s}_{j,t}, \tilde{\boldsymbol{a}}_{-j,t}\}$ which is used by u_j to make a real-time handover decision. In $\tilde{\boldsymbol{s}}_{j,t}$, $\tilde{\boldsymbol{a}}_{-j,t} = (\tilde{a}_{1,t}, \ldots, \tilde{a}_{j-1,t}, \tilde{a}_{j+1,t}, \ldots, \tilde{a}_{K,t})$, where $\tilde{a}_{j',t} = -1$ when $u_{j'}$ is in the inactive state during time step t, otherwise $\tilde{a}_{j',t} = a_{j',t-1}$. Considering that reducing the unnecessary handovers is one of the primary objectives of the proposed handover algorithms, the probability of agents choosing to connect to the current serving satellite is high. Thus, we use the actions taken by the agents in the previous time step as an approximation of their actions in the current time step is reasonable. Moreover, during the testing process, since u_j does not need to store training samples, it only uses approximate fingerprints for handover decisions, which is feasible in practice. Therefore, each LEO satellite only needs to broadcast a message containing the index set of all active MTs it currently serves, and the MTs make handover decisions in a distributed way.

C. Training Process

The training process of MAF-DDQN-DIH algorithm at each time step consists of local transition collecting stage and local model training stage. The following provides a detailed explanation of how these two stages operate.

1) Local Transition Collecting Stage: At the state acquisition phase of time step t, u_j observes $\boldsymbol{s}_{j,t}$ and $\boldsymbol{a}_{-j,t-1}$, and uses them to form local state $\tilde{\boldsymbol{s}}_{j,t}$. In addition, if u_j was also active in the previous time step, u_j uses $\boldsymbol{s}_{j,t-1}$ and $\boldsymbol{a}_{-j,t-1}$ to form local state $\hat{\boldsymbol{s}}_{j,t-1}$. Then, based on the local state $\tilde{\boldsymbol{s}}_{j,t}$, u_j chooses a LEO satellite $a_{j,t}$ by using the $\varepsilon-$greedy strategy. Finally, u_j obtains a local reward $r_{j,t}$. Moreover, if u_j was also an active MT previous two time steps, u_j can store a local transition $(\hat{\boldsymbol{s}}_{j,t-2}, a_{j,t-2}, r_{j,t-2}, \hat{\boldsymbol{s}}_{j,t-1})$ in the local replay memory $\mathcal{D}_j$.

2) Local Model Training Stage: First, u_j randomly samples a mini-batch $\mathcal{D}_{j,W} = \{\boldsymbol{d}_{j,w}\} \subset \mathcal{D}_j$, which contains W transitions and $\boldsymbol{d}_{j,w} = (\hat{\boldsymbol{s}}_{j,w}, a_{j,w}, r_{j,w}, \hat{\boldsymbol{s}}'_{j,w})$ denotes the w-th transition in $\mathcal{D}_j$. Then, similar to that in [33], the target value of $\boldsymbol{d}_{j,w}$ is computed as

$$y_{j,w} = r_{j,w} + \gamma Q\left(\hat{\boldsymbol{s}}'_{j,w}, \underset{a_j}{\operatorname{argmax}}\, Q\left(\hat{\boldsymbol{s}}'_{j,w}, a_j; \boldsymbol{\theta}_{j,t}\right); \tilde{\boldsymbol{\theta}}_{j,t}\right), \tag{8}$$

where $\boldsymbol{\theta}_{j,t}$ and $\tilde{\boldsymbol{\theta}}_{j,t}$ denote the parameters of the online network and the target network for u_j. By using the W target values from the mini-batch $\mathcal{D}_{j,W}$, the value of the loss function can be calculated as

$$\mathcal{L}\left(\boldsymbol{\theta}_{j,t}\right) = \frac{1}{W}\sum_{w=1}^{W}\left[y_{j,w} - Q\left(\hat{\boldsymbol{s}}_{j,w}, a_{j,w}; \boldsymbol{\theta}_{j,t}\right)\right]^2. \tag{9}$$

Finally, u_j updates the parameter $\boldsymbol{\theta}_{j,t}$ using the Adam optimization approach according to the value of the loss function. Furthermore, every $\tilde{N}$ time steps, the parameters of the target network are synchronized with those of the online network.

The details of the whole training process at one training episode are given in Algorithm 1, which systematically summarizes the operations of the above two stages.

It is worth noting that it is feasible for each MT to obtain its handover policy differently from other MTs' by independently training the local DDQN model under the same global environmental parameters due to the randomness in local DDQN model initialization, randomness in local $\varepsilon-$greedy exploration strategy, and randomness in local training sample selection. Therefore, the fully decentralized framework can enable MTs to take personalized local actions to avoid the satellite load imbalance problem.

D. Testing Process

In the testing process, each active MT uses the well-trained local model to decide on a handover according to the greedy policy, which always selects the visible LEO satellite with the maximum Q value. Due to the generalization of the neural network and the fact that any area-specific information is not considered in the designed local state, the trained local model will not become useless as the MT moves into a new area. Even if the performance of the local model decreases after the MT enters a new area, the MT can directly fine-tune the existing local model without retraining a new local model, which significantly decreases the training time by using the intelligence of the existing local model.

V. Implementations of the MAF-DDQN-DIH Mechanism

In this section, the implementations of the MAF-DDQN-DIH algorithm in a practical LEO satellite network are illustrate first, and then the communication overhead and the computational complexity are analyzed.

A. Implementations

The implementation of MAF-DDQN-DIH algorithm in the LEO satellite network requires additional explanation because of the distinct architectures of the LEO satellite network and the terrestrial network. With the coordination of MTs and LEO satellites, the handover procedure using the MAF-DDQN-DIH algorithm is described in Fig. 6. In the handover procedure, communication between LEO satellites can be realized through inter-satellite links (ISLs).

In detail, at the beginning of each time step, each LEO satellite broadcasts a message to all the MTs in its coverage, which includes the index set of active MTs served by it. Then, the MT uses its local DDQN to select a LEO satellite according to its local state. If a handover is triggered, the MT submits its

Algorithm 1: Procedure for the MAF-DDQN-DIH Mechanism Training.

Input: $\mathcal{D}_j$ - replay memory of u_j; $\boldsymbol{\theta}_{j,1}$ - initial online network parameters for u_j; $\tilde{\boldsymbol{\theta}}_{j,1}$ - replicate of $\boldsymbol{\theta}_{j,1}$; $\mathcal{U}$ - set of all MTs; $\tilde{N}$ - parameter update interval of the target network.

for $t \in [1, T]$ **do**
  for $u_j \in \mathcal{U}$ **do**
    if u_j is active **then**
      u_j observes $\boldsymbol{a}_{-j,t-1}$ and $\boldsymbol{s}_{j,t}$.
      u_j uses $\boldsymbol{a}_{-j,t-1}$ and $\boldsymbol{s}_{j,t}$ to form state $\tilde{\boldsymbol{s}}_{j,t}$.
      if u_j was active in the previous time step **then**
        u_j uses $\boldsymbol{a}_{-j,t-1}$ and $\boldsymbol{s}_{j,t-1}$ to form state $\hat{\boldsymbol{s}}_{j,t-1}$.
      end if
      u_j adopts the $\varepsilon-$greedy strategy to select a LEO satellite $a_{j,t}$.
    end if
    $u_j \leftarrow u_{j+1}$.
  end for
  for $u_j \in \mathcal{U}$ **do**
    if u_j is active **then**
      u_j connects to satellite $a_{j,t}$ and receives reward $r_{j,t}$.
      if u_j was active in the previous two time steps **then**
        u_j stores the tuple $(\hat{\boldsymbol{s}}_{j,t-2}, a_{j,t-2}, r_{j,t-2}, \hat{\boldsymbol{s}}_{j,t-1})$ in replay memory $\mathcal{D}_j$.
      end if
    end if
    $u_j \leftarrow u_{j+1}$.
  end for
  for $u_j \in \mathcal{U}$ **do**
    u_j samples a random minibatch of tuples $\mathcal{D}_{j,W}$ from $\mathcal{D}_j$.
    for $\boldsymbol{d}_{j,w} = (\hat{\boldsymbol{s}}_{j,w}, a_{j,w}, r_{j,w}, \hat{\boldsymbol{s}'}_{j,w}) \in \mathcal{D}_{j,W}$ **do**
      if $\hat{\boldsymbol{s}'}_{j,w}$ is the terminal state **then**
        $y_{j,w} = r_{j,w}$.
      else
        Calculate $y_{j,w}$ based on (8).
      end if
    end for
    u_j updates the parameter $\boldsymbol{\theta}_{j,t}$ using the Adam optimization approach according to (9).
    $u_j \leftarrow u_{j+1}$.
  end for
  for $u_j \in \mathcal{U}$ **do**
    if $\tilde{N}|t$ **then**
      u_j resets the target network parameter $\tilde{\boldsymbol{\theta}}_{j,t} \leftarrow \boldsymbol{\theta}_{j,t}$.
    end if
    $u_j \leftarrow u_{j+1}$.
  end for
  $t \leftarrow t + 1$.
end for

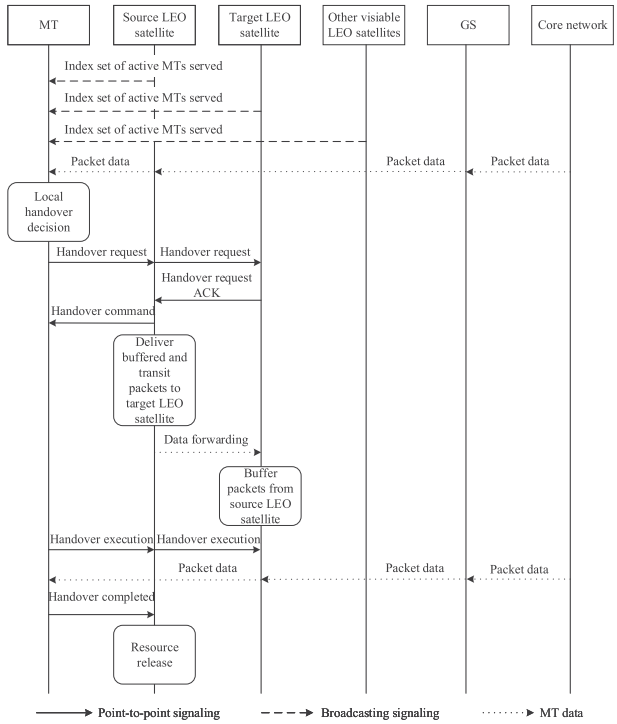


Fig. 6. LEO satellite network handover procedure based on MAF-DDQN-DIH.

handover request to the target LEO satellite. This handover is carried out by the source and target LEO satellites following confirmation of the handover command. The resource of the source LEO satellite is released once the handover is completed.

B. Communication Overhead

According to Fig. 6, compared with the traditional handover scheme, the additional operations of the MAF-DDQN-DIH handover are that each LEO satellite broadcasts a message including the index set of active MTs it serves, and MTs in need of a handover send handover requests to their source LEO satellites. Therefore, the communication overhead induced by these two procedures is focused on.

Firstly, for the broadcasting messages, the quantity of signaling exchanges could be computed as $\max_{u_j \in \mathcal{U}} |\mathcal{B}_{j,t}|$, which is equal to the maximum number of visible satellites for all MTs in the area at time step t. Secondly, for the sending handover requests, the quantity of signaling exchanges is $\tilde{K}_t$, where $\tilde{K}_t$ is the number of MTs triggering handover at time step t. In summary, the total quantity of extra signaling changes of the MAF-DDQN-DIH algorithm is $(\max_{u_j \in \mathcal{U}} |\mathcal{B}_{j,t}| + \tilde{K}_t)$ for the handover process at time step t, and each signaling exchange uses only several bits.

C. Computational Complexity

The computational complexity of the MAF-DDQN-DIH algorithm is similar to that in [18] except for that the agent is trained by each MT in the fully distributed training procedure.

Generally, the computational complexity associated with the training procedure of a DDQN could be computed as $O(W\sum_{z=1}^{Z} H_{z-1}H_z)$[18], where Z denotes the number of dense layers, H_z denotes the output dimensionality of the z-th dense layer, and $H_0 = |\tilde{\boldsymbol{s}}_{j,t}|$ denotes the length of the local state vector. Thus the computational complexity associated with the fully distributed training procedure of the MAF-DDQN-DIH algorithm is

$$\Omega_{\mathrm{tr}} = KO\left(W\sum_{z=1}^{Z} H_{z-1}H_z\right) = O\left(KW\sum_{z=1}^{Z} H_{z-1}H_z\right), \tag{10}$$

which has a linear complexity with respect to the number of MTs.

The computational complexity associated with the decision-making procedure using a DDQN is $O(\sum_{z=1}^{Z} H_{z-1}H_z)$. Since only the active MTs participate in the decision-making procedure of the MAF-DDQN-DIH algorithm, the computational complexity is

$$\Omega_{\mathrm{de}} = \hat{K}_t O\left(\sum_{z=1}^{Z} H_{z-1}H_z\right) = O\left(\hat{K}_t\sum_{z=1}^{Z} H_{z-1}H_z\right), \tag{11}$$

which has a linear relationship with the number of active MTs.

With the increasing interest in artificial intelligence (AI) for mobile applications, MT vendors such as Qualcomm, Huawei, Samsung, and MediaTek have launched smartphones featuring powerful dedicated AI chips, with computational performance of up to 3.2 trillion floating-point operations per second [34], [35]. Therefore, the power consumption of training and running local DDQN models on MTs is acceptable [35]. Moreover, with the evolution of mobile system-on-chip technologies, the computational performance of MTs will further increase, and the power consumption of training and running local DDQN models on MTs will further decrease.

VI. Simulation and Numerical Results

The MAF-DDQN-DIH algorithm is compared with six handover algorithms, including "MA-DDQN-DIH", "Max-CNR", "Max-VIS", "Min-Load", "Random" and "Greedy", to assess its mobility performance. A detailed description of how these algorithms work is given in the following.

- **MAF-DDQN-DIH**: As described in Section IV.
- **MA-DDQN-DIH**: Apart from the absence of mult-agent fingerprints in the local state, all other settings remain consistent with MAF-DDQN-DIH.
- **Max-CNR**: Each active MT chooses the visible LEO satellite with the highest CNR.
- **Max-VIS**: Each active MT chooses the visible LEO satellite with the longest remaining visible time.

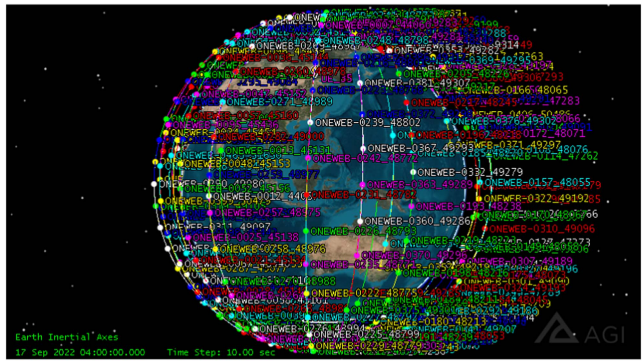


Fig. 7. OneWeb constellation in STK.

TABLE I
Simulation Parameters

Parameters	Value
number of LEO satellites	$N = 28$
number of MTs	$K \in \{10, 20, 30, 40, 50\}$
minimum elevation angle [19]	$\theta_0 = 10^\circ$
T [19], [21]	600
Δt [19], [21]	1 s
frequency of transmitter	14.5 GHz
frequency of receiver	14.5 GHz
$P_{\mathrm{TX}}G_{\mathrm{TX}}L_{\mathrm{TX}}^{-1}$	30 dBW
Θ	1.3806×10^{-23} W/KHz
$L_{i,j}(t)$	free space loss and RainModel
$G_{\mathrm{RX}}/F_{\mathrm{RX}}$	20 dB/K
W_{RF}	32 MHz
W	20 MHz
λ	$\{0.2, 0.4, 0.6, 0.8, 1\}$ packet/s
q	30 Mb
T_{drop} [22]	8 s
(f_1, f_2, f_3)	$(1, -1, -1)$
(μ_1, μ_2, μ_3)	$\{(3/8, 1/4, 3/8), (1/3, 1/3, 1/3), (1/4, 1/2, 1/4)\}$
Φ [12]	10

- **Min-Load**: Each active MT chooses the visible LEO satellite with the least load.
- **Random**: Each active MT randomly chooses a visible LEO satellite.
- **Greedy**: Each active MT u_j chooses the visible LEO satellite that can obtain the maximum reward as defined in (5). Because u_j's real reward depends on the joint action of all active MTs, whereas the reward estimated by the Greedy algorithm only considers u_j's action, the real reward may differ from the estimated reward.

A. Simulation Settings

As shown in Fig. 7, the OneWeb constellation is constructed in STK using the Two-Line Orbital Element (TLE) data provided in CelesTrak website [36]. We take into account a square area with 2 km side length [19], centered on $(39.8853^\circ\mathrm{N}, 116.477^\circ\mathrm{E})$, where MTs are uniformly scattered within the area. Table I provides a summary of the simulation parameters. Since only the handover of the area during $T = 600$ time steps is considered, $N = 28$ LEO satellites in OneWeb constellation participate in the simulation, and $|\mathcal{B}_{j,t}| \leq N$.

The online network or target network used in the simulation consists of two hidden layers and one output layer. Specifically, the output dimensionality of each hidden layer is 128, and the output dimensionality of the output layer is 28. Moreover, the replay memory is configured with a size of $|\mathcal{D}_j| = 10000$,

the mini-batch is configured with a size of $W = 32$, the Adam optimizer is configured with a learning rate of 0.0001, the discount factor is specified as $\gamma = 0.9$, and the update frequency of the target network parameter is set to $\tilde{N} = 600$ time steps. We train the agent for 150 epochs, with each epoch consisting of 20 episodes, and the rate of the exploration is 0.1.

To obtain the training experiences, 20 network snapshots are generated and simulated during the training procedure, and each snapshot, which lasts $T = 600$ time steps, is treated as a complete episode. During the testing procedure, 50 network snapshots independent of the training snapshot are created.

B. Results and Analysis

1) Performance With Various Reward Weights: First, the selection of the values of the reward weights is explained, and then the performance of seven handover methods is compared under various reward weights while $K = 50$ and $\lambda = 1$.

Firstly, we fairly consider the three designed reward components – the service revenue, the cost of handover and the cost of packet loss. Due to the different dimensions of these three performance indicators, equivalent data volume related to each indicator is used as intermediate quantity for comparison. According to the definition of reward given in (5), the service revenue is calculated by $(n_{j,t}/q)f_1$, which means that a successfully received data packet (30 Mbits) can add f_1 ($f_1 = 1$) scores to the service revenue. As for the cost of packet loss given by $\eta_{j,t}f_3$, it means that the failed transmission of a data packet (30 Mbits) will get f_3 ($f_3 = -1$) scores. Since the handover process takes some time to implement, the amount of data that may be transmitted during this period can be calculated. Considering the lack of accurate empirical data for handover delay between satellites using OFDM technology, the handover delay of terrestrial network which is typically set at 100 ms [24] is used for reference. Obviously, the inter-satellite handover delay is higher than the terrestrial reference value and can be assumed to be several hundred milliseconds to 1 s. Thus, the data loss caused by an inter-satellite handover is approximately on the same order of magnitude of a data packet (30 Mbits) when the data package arrival rate is set between 0.2 to 1 and the system is working without congestion. Therefore, one handover causes the change of f_2 ($f_2 = -1$) scores in the cost of handover $\delta_{j,t}f_2$. Considering traditional handover algorithms such as Max-CNR, Max-VIS, Min-Load, and Random don't consider the influence of reward weights, (μ_1, μ_2, μ_3) are set to $(1/3, 1/3, 1/3)$ for fair comparison. Specifically, this ensures that the value of each weighted reward components is consistent with the actual amount of received or lost data, thus objectively reflecting the overall performance of the algorithms through the total reward.

Additionally, to investigate the impact of different reward weights on the agent's preferences and the results, we also consider two cases: $\mu_1 = \mu_3 > \mu_2$ and $\mu_1 = \mu_3 < \mu_2$. Specifically, we set (μ_1, μ_2, μ_3) to $(3/8, 1/4, 3/8)$ and $(1/4, 1/2, 1/4)$ respectively.

The training results in Figs. 8(a), 9(a), and 10(a) indicate that MAF-DDQN-DIH and MA-DDQN-DIH all converge after 150 training epochs under various reward weights. Since MAF-DDQN-DIH achieves fewer handovers while maintaining its service revenue and cost of packet loss almost the same as those of MA-DDQN-DIH, MAF-DDQN-DIH attains a higher total reward than MA-DDQN-DIH. Additionally, the MAF-DDQN-DIH converges faster than MA-DDQN-DIH. For example, Fig. 8(a) shows that MAF-DDQN-DIH's total reward is more than 6600 after about 40 training epochs, while MA-DDQN-DIH takes 150 epochs to reach that level. This implies that MAF-DDQN-DIH can achieve good performance after a short training period, which is crucial for practical applications that require rapid deployment.

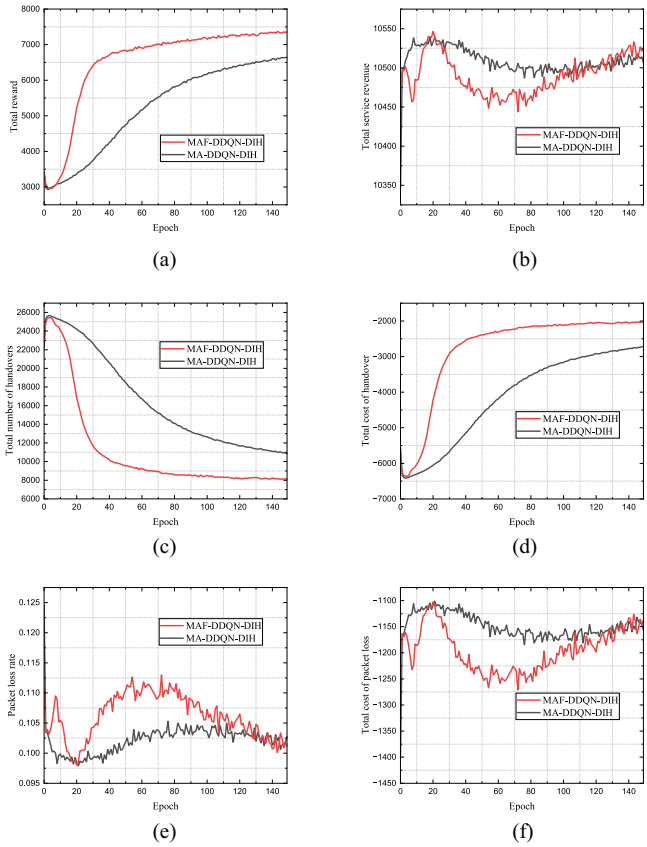


Fig. 8. Learning results with $(\mu_1, \mu_2, \mu_3) = (\frac{3}{8}, \frac{1}{4}, \frac{3}{8})$. (a) Total reward. (b) Service revenue. (c) Number of handovers. (d) Cost of handover. (e) Packet loss rate. (f) Cost of packet loss.

It can be seen that the curves in Figs. 8, 9, and 10 exhibit fluctuations. Due to the similar trends of fluctuations, Fig. 8 is taken as an example for explanation. At the beginning of the training, MA-DDQN-DIH attempt to increase total service revenues and reduce packet loss rates by increasing the number of handovers to enhance the total reward. MAF-DDQN-DIH also follows this trend of MA-DDQN-DIH after experiencing a short-term oscillation. However, because the gains in service revenues and packet loss cost are not as significant as the losses in the handover cost, their overall rewards decrease slightly. Subsequently, these two algorithms explore handover strategies that reduced the number of handovers to improve their total rewards. At this stage, they sacrifice the performance of service revenues and packet

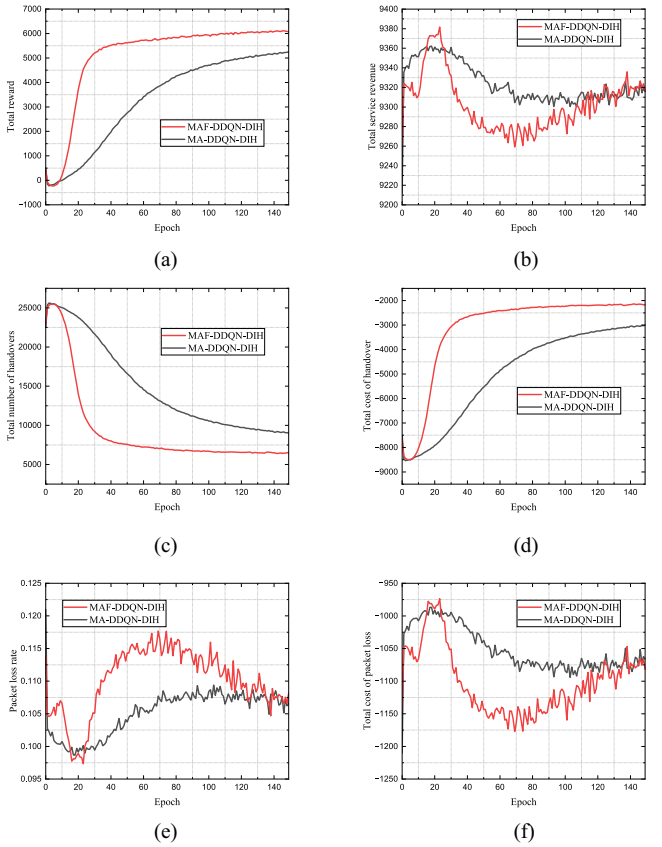


Fig. 9. Learning results with $(\mu_1, \mu_2, \mu_3) = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$. (a) Total reward. (b) Service revenue. (c) Number of handovers. (d) Cost of handover. (e) Packet loss rate. (f) Cost of packet loss.

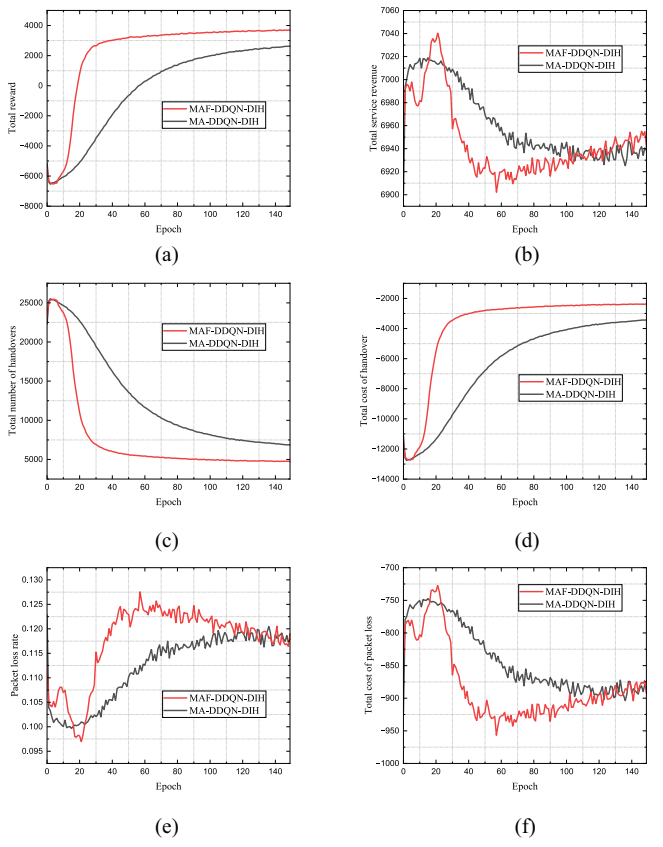


Fig. 10. Learning results with $(\mu_1, \mu_2, \mu_3) = (\frac{1}{4}, \frac{1}{2}, \frac{1}{4})$. (a) Total reward. (b) Service revenue. (c) Number of handovers. (d) Cost of handover. (e) Packet loss rate. (f) Cost of packet loss.

loss rates. MAF-DDQN-DIH explores much more aggressively than MA-DDQN-DIH, and it reduces the number of handovers much more quickly. When the number of handovers decreases to a relatively low level, these two algorithms continuously sought better strategies to improve performance of service revenues and packet loss rates. Meanwhile, the number of handovers doesn't increase but slightly decreases, which show the intelligence of the learning-based algorithm, and the total reward is increased. Due to more aggressive exploration process, MAF-DDQN-DIH ultimately finds a handover strategy that significantly reduces the number of handovers compared to MA-DDQN-DIH while maintaining the performance of service revenue and packet loss rate similar to that of MA-DDQN-DIH, thereby achieving a higher total reward than MA-DDQN-DIH.

In a word, these training results demonstrate that the multi-agent fingerprints designed in this paper effectively overcome the non-stationary problem of the environment, thereby improving the performance of the handover algorithm.

The test results in Fig. 11 show that the performance of MAF-DDQN-DIH is superior to those of other comparison algorithms under various reward weights. Moreover, Fig. 11(c) shows that as the weight of handover cost increases, the number of handovers using MAF-DDQN-DIH, MA-DDQN-DIH and Greedy algorithms decreases. This phenomenon indicates that changes in the reward weights can lead to changes in the preferences of intelligent agents for the corresponding system performance metric, thereby adapting to different system design objectives. Although Greedy algorithm can also adapt to the changes in the reward weights, its overall performance is not good. For the MAF-DDQN-DIH algorithm, when the reward weights are set to $(3/8, 1/4, 3/8)$, $(1/3, 1/3, 1/3)$, and $(1/4, 1/2, 1/4)$, the throughputs are 1.375 Gbps, 1.367 Gbps, and 1.338 Gbps, the total numbers of handovers are 3405.520, 1822.700, and 506.300, and the packet loss rates are 13.682%, 14.712%, and 18.525%, respectively. This suggests that we can increase the weight assigned to handover cost when there's a greater demand for reducing handovers than for improving throughput and reducing packet loss rates in a given QoS requirement, and conversely, reduce the weight assigned to handover cost. Therefore, for any given QoS requirements, the weight of each reward component can be set based on the preference level for system performance indicator corresponding to this reward component.

It is noticeable that the total reward in the test results is higher compared to the training results. This is because we set the exploration rate to 0.1 during training and 0 during testing. The zero exploration rate in testing significantly reduces the handover cost caused by random exploration at the cost of slightly increasing in packet loss cost, and thus, obtain the performance gain in the total reward. Furthermore, Fig. 11(c) shows that the decrease in the number of handovers using MA-DDQN-DIH

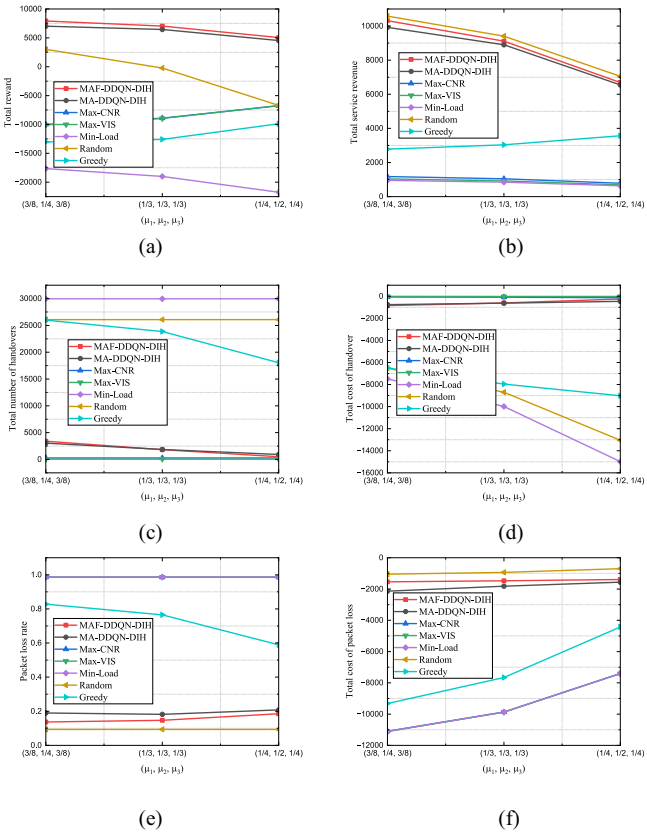


Fig. 11. Performance comparison with various reward weights. (a) Total reward. (b) Service revenue. (c) Number of handovers. (d) Cost of handover. (e) Packet loss rate. (f) Cost of packet loss.

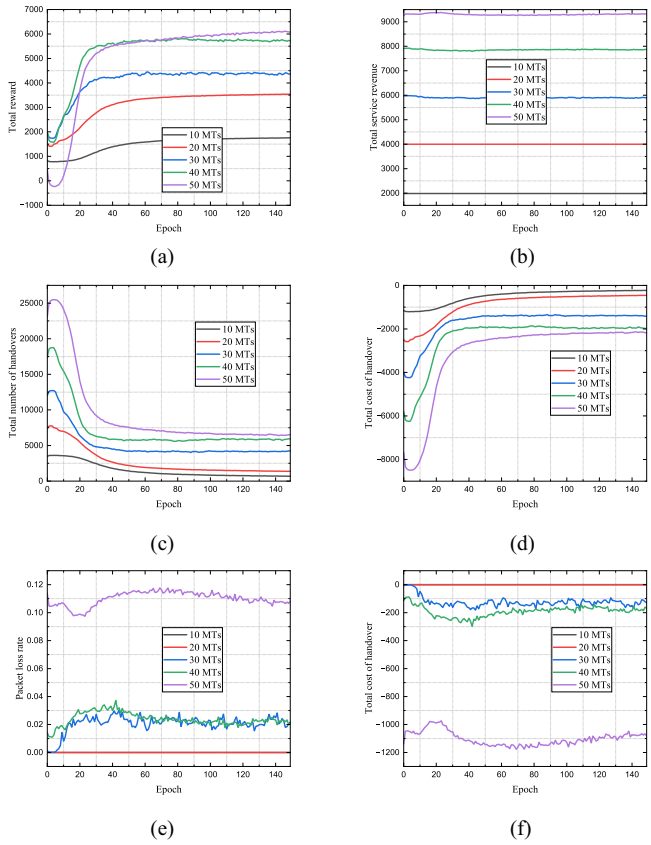


Fig. 12. Learning results of MAF-DDQN-DIH with various numbers of MTs. (a) Total reward. (b) Service revenue. (c) Number of handovers. (d) Cost of handover. (e) Packet loss rate. (f) Cost of packet loss.

is particularly significant compared to the training results, but the performance loss of service revenue and packet loss using this algorithm is also significant. Therefore, the total reward of MA-DDQN-DIH is the worse than MAF-DDQN-DIH, which validates the effectiveness of fingerprints in addressing environmental instability issues. Obviously, to maintain the similar performance of handovers and packet loss rate in test results as in training results, setting the exploration rate to 0.1 is an option. However, setting the exploration rate to 0 can obtain better performance gain in total rewards. Therefore, in practical deployment, the decision to set the exploration rate to 0.1 or 0 can be made based on different system design preferences.

2) Performance With Various Numbers of MTs: The performance of different mechanisms with various numbers of MTs is compared. The number of MTs is set as $K \in \{10, 20, 30, 40, 50\}$ while $\lambda = 1$ and $(\mu_1, \mu_2, \mu_3) = (1/3, 1/3, 1/3)$.

According to Fig. 12(a), after 150 training epochs, the total reward converges. It can be seen from Fig. 12(b), (c), and (e) that as the number of training epochs grows, the total service revenue is almost unchanged, the total number of handovers is gradually reduced, and the packet loss rate is slightly increased. This is because the parameters of DDQN are randomly initialized, which means that the MAF-DDQN-DIH algorithm in the previous several epochs is equivalent to the Random algorithm. Since the channel quality between the MT and each visible LEO satellite is very close, using the Random algorithm will balance the load of all visible LEO satellites through frequent handovers, thus achieving excellent service revenue and packet loss rate performance. However, the Random algorithm also causes a huge total cost of handover. Therefore, at the beginning of training, MAF-DDQN-DIH achieves excellent service revenue and packet loss rate performance. As training progresses, In order to achieve better comprehensive performance, the MAF-DDQN-DIH algorithm reduces the total cost of handover greatly while maintaining the total service revenue and slightly increasing the packet loss rate.

Fig. 13 demonstrates that the performance of MAF-DDQN-DIH is superior to other handover algorithms. Compared to MA-DDQN-DIH, the advantage of MAF-DDQN-DIH lies in its ability to achieve higher service revenues and lower packet loss rates through similar or fewer handover times. This indicates that MAF-DDQN-DIH can make more effective handover than MA-DDQN-DIH. It could be concluded from Fig. 13(a) that when the number of MTs is small ($K \in \{10, 20\}$), the total reward of the MA-DDQN-DIH scheme is very close to the MAF-DDQN-DIH scheme. This is because when there are few MTs, the impact of non-stationary environmental problems caused by parallel learning of MTs is relatively small. Therefore,

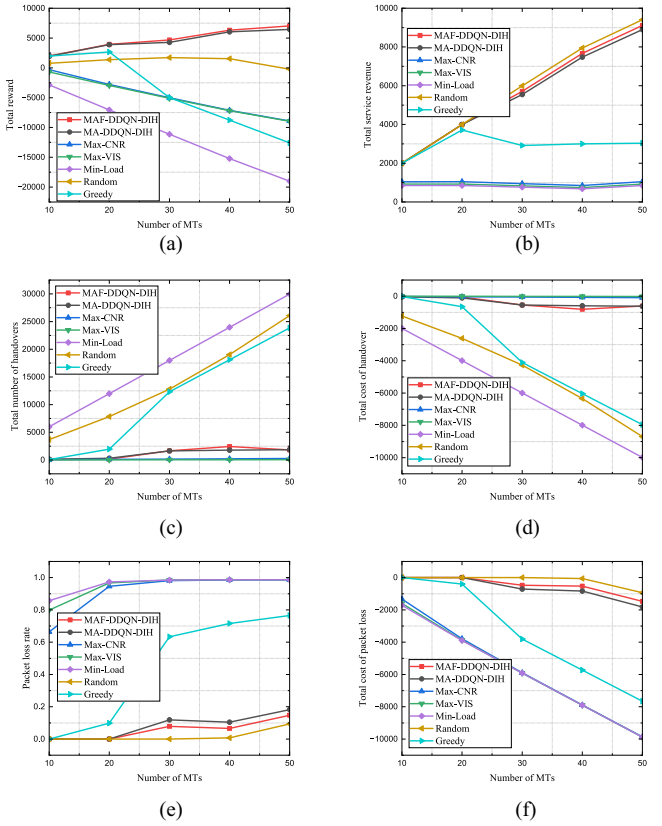


Fig. 13. Performance comparison with various numbers of MTs. (a) Total reward. (b) Service revenue. (c) Number of handovers. (d) Cost of handover. (e) Packet loss rate. (f) Cost of packet loss.

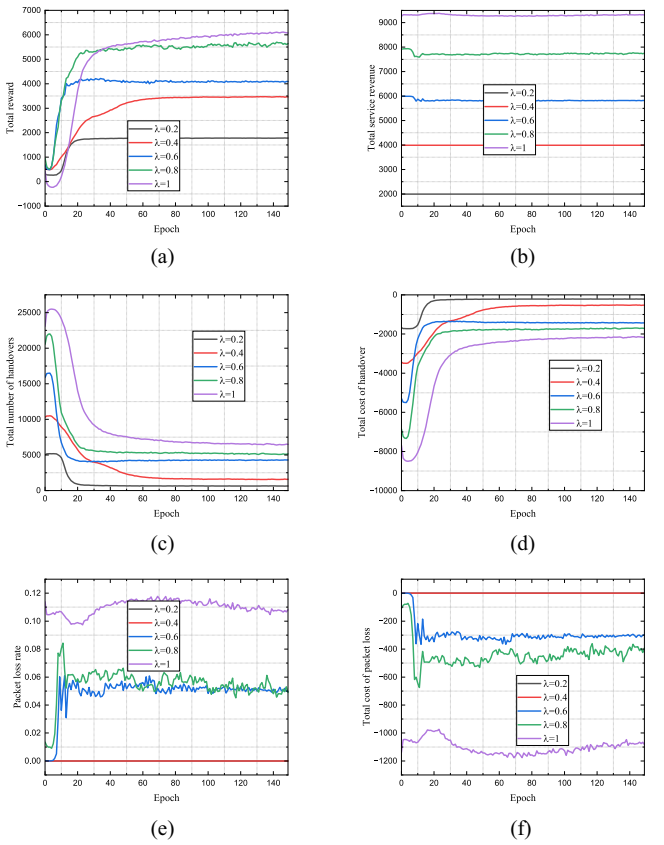


Fig. 14. Learning results of MAF-DDQN-DIH with various packet arrival rates. (a) Total reward. (b) Service revenue. (c) Number of handovers. (d) Cost of handover. (e) Packet loss rate. (f) Cost of packet loss.

the MA-DDQN-DIH achieved almost the same handover performance as the MAF-DDQN-DIH. Similarly, due to the small mutual influence between few MTs, the reward value estimated by the Greedy algorithm is very close to the reward value actually obtained. Therefore, the Greedy algorithm performs quite well. As the number of MTs increases ($K \in \{30, 40, 50\}$), the impact between MTs increases, and the total reward of the Greedy scheme drops sharply, which is only superior to the Min-Load algorithm. For the Max-CNR algorithm, the Max-VIS algorithm, and the Min-Load algorithm, because numerous MTs connect to the same satellite with the highest CNR, the same satellite with the longest remaining visible time, and the same satellite with the smallest load, respectively, the loads of satellites are extremely unbalance, and the network throughput is low. Therefore, the performance of these three handover algorithms is inferior to the MAF-DDQN-DIH algorithm. Due to the frequent handovers, the total reward of the Random algorithm is also less than that of the MAF-DDQN-DIH scheme.

3) Performance With Various Packet Arrival Rates: The performance of the MAF-DDQN-DIH scheme is evaluated with various packet arrival rates, and the packet arrival rate is set as $\lambda \in \{0.2, 0.4, 0.6, 0.8, 1\}$ while $K = 50$ and $(\mu_1, \mu_2, \mu_3) = (1/3, 1/3, 1/3)$.

The training results of the MAF-DDQN-DIH algorithm with various packet arrival rate are given in Fig. 14. It can be found that the MAF-DDQN-DIH algorithm can converge well under various packet arrival rates, which again verifies the convergence of the MAF-DDQN-DIH algorithm. In addition, the trends of total rewards and reward components is similar to those of Fig. 12. This is because, as the packet arrival rate decreases, the number of active MTs at each time step decreases accordingly, which is equivalent to reducing the total number of MTs.

Fig. 15 shows that the MAF-DDQN-DIH outperforms all other handover algorithms. Compared to the MA-DDQN-DIH, the advantage of MAF-DDQN-DIH is that it can achieve similar or better service revenue and packet loss rate performance with fewer handover times. This once again confirms the effectiveness of the designed multi-agent fingerprints. According to the comparison results in Fig. 15(a), the Greedy and MA-DDQN-DIH algorithms performs very closely to the MAF-DDQN-DIH algorithm when the packet arrival rate is low ($\lambda \in \{0.2, 0.4\}$). This is due to the fact that fewer MTs are active during each time step when the packet arrival rate is low. When $\lambda \in \{0.6, 0.8, 1\}$, more MTs are active during each time step, and thus, the performance of the Greedy and MA-DDQN-DIH algorithms drops. Other handover algorithms show the similar performance trends to those in Fig. 13. In summary, the MAF-DDQN-DIH scheme exceeds all other handover schemes in terms of performance and stability at various packet arrival rates.

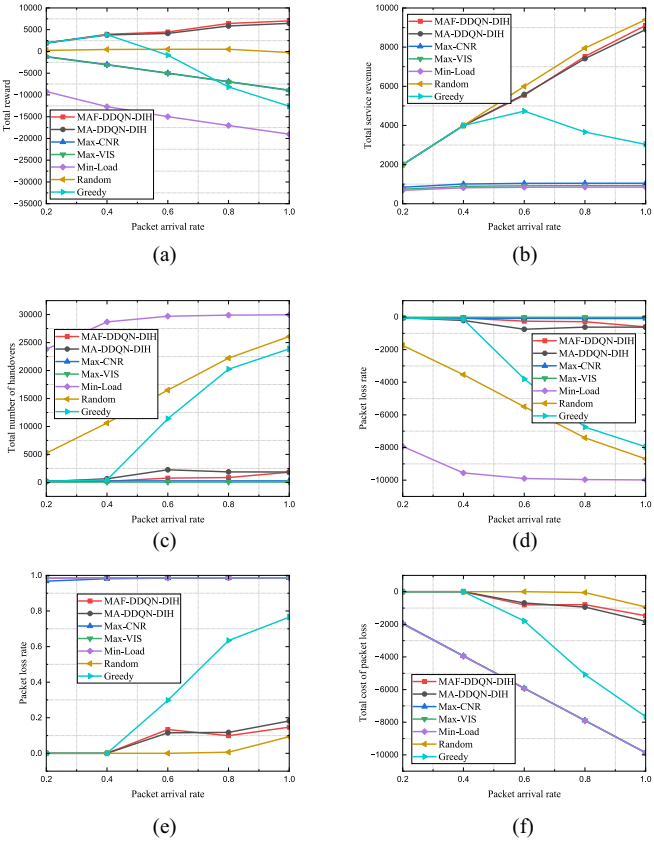


Fig. 15. Performance comparison with various packet arrival rates. (a) Total reward. (b) Service revenue. (c) Number of handovers. (d) Cost of handover. (e) Packet loss rate. (f) Cost of packet loss.

VII. Conclusion

This paper addresses the handover problem in LEO satellite networks serving emerging delay-sensitive applications. The finite bursty traffic model is used realize the evaluation of delay and package loss. The fully decentralized handover decision-making framework with low communication overhead and low decision-making complexity is designed. The handover problem is modeled as an DEC-MDP maximizing the total reward associated with the service revenue and the cost of handover and packet loss. The multi-agent fingerprints-enhanced distributed intelligent handover mechanism named MAF-DDQN-DIH is proposed, which allows each MT to train and use an individual local DDQN to avoid load imbalance between satellites and relieves the non-stationary issue of the environment caused by parallel learning. The handover procedures supporting the proposed distributed intelligent handover algorithm are designed, and the corresponding communication overhead and computational complexity are analyzed. The simulation results show that the multi-agent fingerprint designed in this paper can effectively improve the comprehensive performance of the handover algorithm, and the performance of MAF-DDQN-DIH is superior to other comparison handover algorithms in simulation scenarios with various reward weights, various number of MTs and various package arrival rates.

References

[1] S. Ji, M. Sheng, D. Zhou, W. Bai, Q. Cao, and J. Li, "Flexible and distributed mobility management for integrated terrestrial-satellite networks: Challenges, architectures, and approaches," *IEEE Netw.*, vol. 35, no. 4, pp. 73–81, Jul./Aug. 2021.

[2] B. Di, L. Song, Y. Li, and H. V. Poor, "Ultra-dense LEO: Integration of satellite access networks into 5G and beyond," *IEEE Wireless Commun.*, vol. 26, no. 2, pp. 62–69, Apr. 2019.

[3] R. Wang, M. A. Kishk, and M.-S. Alouini, "Ultra-dense LEO satellite-based communication systems: A novel modeling technique," *IEEE Commun. Mag.*, vol. 60, no. 4, pp. 25–31, Apr. 2022.

[4] 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.

[5] S. Zhang, A. Liu, C. Han, X. Ding, and X. Liang, "A network-flows-based satellite handover strategy for LEO satellite networks," *IEEE Wireless Commun. Lett.*, vol. 10, no. 12, pp. 2669–2673, Dec. 2021.

[6] 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.

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

[8] E. Juan, M. Lauridsen, J. Wigard, and P. Mogensen, "Location-based handover triggering for low-earth orbit satellite networks," in *Proc. IEEE 95th Veh. Technol. Conf.*, 2022, pp. 1–6.

[9] 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.

[10] Y. I. Demir, M. S. J. Solaija, and H. Arslan, "On the performance of handover mechanisms for non-terrestrial networks," in *Proc. IEEE 95th Veh. Technol. Conf.*, 2022, pp. 1–5.

[11] W. Lin et al., "A novel method to determine the handover threshold based on reconfigurable factor graph for LEO satellite internet network," *IEEE Access*, vol. 10, pp. 31907–31921, 2022.

[12] J. Li, K. Xue, J. Liu, and Y. Zhang, "A user-centric handover scheme for ultra-dense LEO satellite networks," *IEEE Wireless Commun. Lett.*, vol. 9, no. 11, pp. 1904–1908, Nov. 2020.

[13] C.-Q. Dai, Y. Liu, S. Fu, J. Wu, and Q. Chen, "Dynamic handover in satellite-terrestrial integrated networks," in *Proc. IEEE Globecom Workshops*, 2019, pp. 1–6.

[14] Y. Liu, X. Tang, Y. Zhou, J. Shi, M. Qian, and S. Li, "Channel reservation based load aware handover for LEO satellite communications," in *Proc. IEEE 95th Veh. Technol. Conf.*, 2022, pp. 1–5.

[15] C.-Q. Dai, J. Xu, J. Wu, and Q. Chen, "Multi-objective intelligent handover in satellite-terrestrial integrated networks," in *Proc. IEEE Int. Conf. Commun. Workshops*, 2022, pp. 367–372.

[16] 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.

[17] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 2018.

[18] W. Wu et al., "Distributed handoff problem in heterogeneous networks with end-to-end network slicing: Decentralized Markov decision process-based modeling and solution," *IEEE Trans. Wireless Commun.*, vol. 21, no. 12, pp. 11222–11236, Dec. 2022.

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

[20] F. Yang, W. Wu, X. Wang, Y. Zhang, and P. Si, "Deep reinforcement learning based handoff algorithm in end-to-end network slicing enabling hetnets," in *Proc. IEEE Wireless Commun. Netw. Conf.*, 2021, pp. 1–7.

[21] H. Liu, Y. Wang, and Y. Wang, "A successive deep Q-learning based distributed handover scheme for large-scale LEO satellite networks," in *Proc. IEEE 95th Veh. Technol. Conf.*, 2022, pp. 1–6.

[22] S. Vasudevan, R. N. Pupala, and K. Sivanesan, "Dynamic eICIC — A proactive strategy for improving spectral efficiencies of heterogeneous LTE cellular networks by leveraging user mobility and traffic dynamics," *IEEE Trans. Wireless Commun.*, vol. 12, no. 10, pp. 4956–4969, Oct. 2013.

[23] J. Foerster et al., "Stabilising experience replay for deep multi-agent reinforcement learning," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 1146–1155.

[24] D. Guo, L. Tang, X. Zhang, and Y.-C. Liang, "Joint optimization of handover control and power allocation based on multi-agent deep reinforcement learning," *IEEE Trans. Veh. Technol.*, vol. 69, no. 11, pp. 13124–13138, Nov. 2020.
[25] H. Elshaer, M. N. Kulkarni, F. Boccardi, J. G. Andrews, and M. Dohler, "Downlink and uplink cell association with traditional macrocells and millimeter wave small cells," *IEEE Trans. Wireless Commun.*, vol. 15, no. 9, pp. 6244–6258, Sep. 2016.
[26] T. Gabel and M. A. Riedmiller, "Reinforcement learning for DEC-MDPs with changing action sets and partially ordered dependencies," in *Proc. 7th Int. Joint Conf. Auton. Agents Multiagent Syst.*, 2008, pp. 1333–1336.
[27] D. S. Bernstein, R. Givan, N. Immerman, and S. Zilberstein, "The complexity of decentralized control of Markov decision processes," *Math. Operations Res.*, vol. 27, no. 4, pp. 819–840, 2002.
[28] 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. 2020.
[29] N. Zhao, Y.-C. Liang, D. Niyato, Y. Pei, M. Wu, and Y. Jiang, "Deep reinforcement learning for user association and resource allocation in heterogeneous cellular networks," *IEEE Trans. Wireless Commun.*, vol. 18, no. 11, pp. 5141–5152, Nov. 2019.
[30] 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.
[31] J. K. Gupta, M. Egorov, and M. Kochenderfer, "Cooperative multi-agent control using deep reinforcement learning," in *Proc. Int. Conf. Auton. Agents Multiagent Syst.*, 2017, pp. 66–83.
[32] G. Tesauro, "Extending Q-learning to general adaptive multi-agent systems," in *Proc. Int. Conf. Adv. Neural Inf. Process. Syst.*, 2003, pp. 871–878.
[33] H. Van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double Q-learning," in *Proc. AAAI Conf. Artif. Intell.*, 2016, pp. 2094–2100.
[34] A. Ignatov et al., "AI benchmark: Running deep neural networks on android smartphones," in *Proc. Eur. Conf. Comput. Vis. Workshops*, 2018, pp. 288–314.
[35] Y.-J. Liu, G. Feng, Y. Sun, S. Qin, and Y.-C. Liang, "Device association for ran slicing based on hybrid federated deep reinforcement learning," *IEEE Trans. Veh. Technol.*, vol. 69, no. 12, pp. 15731–15745, Dec. 2020.
[36] T. Kelso, "Celestrak," 2022. [Online]. Available: https://celestrak.org/NORAD/elements/gp.php?GROUP=oneweb&FORMAT=tle

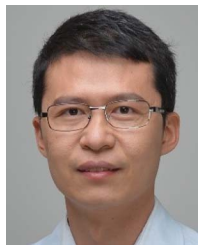
Feng Yang (Graduate Student Member, IEEE) received the M.S. degree in electronic science and technology in 2022 from the Beijing University of Technology, Beijing, China, where he is currently working toward the Ph.D. degree in electronic science and technology. From 2023 to 2024, he visited Keio University, Yokohama, Japan, as a Visiting Ph.D. degree Student funded by China Scholarship Council. His research interests include mobility management, reinforcement learning, deep learning, and game theory.

Wenjun Wu (Member, IEEE) received the B.E. and Ph.D. degrees from the Beijing University of Posts and Telecommunications, Beijing, China, in 2007 and 2012, respectively. From 2012 to 2015, she was a Postdoctoral Researcher with the School of Electronic and Information Engineering, Beihang University, Beijing. She is currently an Associate Professor with the Faculty of Information Technology, Beijing University of Technology, Beijing. Her research interests include mobile edge computing, blockchain, Markov decision process, and deep reinforcement learning.

Yang Gao (Graduate Student Member, IEEE) received the B.S. degree in communication engineering in 2018 from the Beijing University of Technology, Beijing, China, where she is currently working toward the Ph.D. degree in electronic science and technology. Her research interests include mobile edge computing, blockchain, deep reinforcement learning, and wireless resources management.

Yang Sun (Member, IEEE) received the Ph.D. degree from the School of Information and Communication Engineering, Beijing University of Posts and Telecommunications, Beijing, China, in 2018. She is currently a Lecturer with the Beijing University of Technology, Beijing. Her research interests include ultradense heterogeneous network, interference management, massive MIMO, and green telecommunications.

Teng Sun received the M.S. degree in communication and information system from the 54th Research Institute of China Electronics Technology Group Corporation, Shijiazhuang, China, in 2012. He is currently a Senior Engineer with the 54th Research Institute of China Electronics Technology Group Corporation. His research focuses on mobile communications.



Pengbo Si (Senior Member, IEEE) received the B.S. and Ph.D. degrees from the Beijing University of Posts and Telecommunications, Beijing, China, in 2004 and 2009, respectively. In 2009, he joined the Beijing University of Technology, where he is currently a Professor. From 2007 to 2008, he visited Carleton University, Ottawa, ON, Canada. From 2014 to 2015, he was a Visiting Scholar with the University of Florida, Gainesville, FL, USA. His research interests include blockchain, SDN, resource management, and cognitive radio networks. He is an Associate Editor for *International Journal on AdHoc Networking Systems*, an Editorial Board Member of *Ad-Hoc and Sensor Wireless Networks*, and the Symposium Chair of IEEE Globecom 2019. He is a Guest Editor of *Advances in Mobile Cloud Computing*, IEEE TRANSACTIONS ON EMERGING TOPICS IN COMPUTING Special Issue, TPC Co-Chair of IEEE ICCC'13-GMCN, Program Vice Chair of IEEE GreenCom'13, and a TPC Member of numerous conferences.