LinkSlice: Fine-Grained Network Slice Enforcement Based on Deep Reinforcement Learning

Tianxin Wang, Suhong Chen, Yifei Zhu, Aimin Tang, *Member, IEEE*, and Xudong Wang, *Fellow, IEEE*

***Abstract*— Considering network slicing in a cellular network, one of the most intriguing tasks is slice enforcement over air interfaces across multiple cells. The challenges lie in several aspects. First, resources allocated to different slices must achieve soft isolation at the link level. Second, users' diverse QoS requirements must be satisfied even when communication links experience fading and interference. Third, long-term slicing policies must be conformed, no matter how unbalanced they are. To address these challenges, link-level slice enforcement is first formulated as a resource allocation problem that minimizes radio resource consumption while ensuring link-level soft slice isolation, guaranteeing users' diverse QoS requirements, and conforming to slicing policies. Next, this problem is tackled via a deep reinforcement learning (DRL) based approach, through which LinkSlice is designed as an iterative two-stage algorithm. The first stage determines transmission rates for each link based on DRL. It is embedded with a graph neural network (GNN) to characterize link interference. Based on the transmission rates from the first stage, the second stage allocates resources to each slice. Performance results show that LinkSlice converges quickly to a near-optimal solution. It gracefully tackles the three challenges of link-level slice enforcement while further improving throughput by 18.5%.**

***Index Terms*— Network slice enforcement, slice isolation, slicing policy, deep reinforcement learning, graph neural network.**

I. Introduction

5G NETWORKS are expected to support a variety of communication modes, such as enhanced mobile broadband (eMBB), massive machine-type communications (mMTC), and ultra-reliable low-latency communications (URLLC). Service demands vary significantly from mode to mode. Even under the same communication mode, services tailored to different applications can also have diverse quality of service (QoS) requirements. To meet these diverse service demands, network slicing techniques [1]–[3] have been proposed as a key enabler to provide efficient network management. 5G network operators split a shared physical infrastructure into several virtual networks, each of which consists of multiple network slices. As a building block of an end-to-end virtual network, a network slice spans both core networks (CN) and radio access networks (RAN). Therefore, network slicing falls into two main categories, CN slicing and RAN slicing [4].

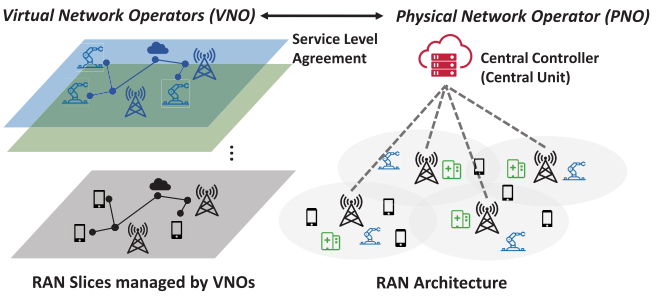


Fig. 1. RAN slicing framework in 5G scenarios.

While CN slicing has already been specified by the 3GPP standards [4], RAN slicing is still in its early stage. In recent years, RAN slicing attracts great attention from both academic and industrial researchers [5]–[8]. As shown in Fig. 1, virtual network operators (VNO) who own and manage network slices need to form payment contracts with physical network operators (PNO) based on the amount of resources allocated to network slices. Such payment contracts are termed service level agreement (SLA) as specified in [4]. Since the users who subscribe to a slice can be distributed across multiple cells, RAN slicing thus involves multiple base stations [9]–[11].

RAN slicing mainly consists of two phases in sequence [11]: slicing policy determination and slice enforcement. In the phase of slicing policy determination, both VNOs and PNOs are involved to determine slicing policies for admitted slices. A slicing policy is a type of SLA that prescribes the percentage of resources reserved on base stations for each admitted slice [11]. In the phase of slice enforcement, given the predetermined slicing policy, physical resource blocks (PRBs) on the involved base stations are allocated to each slice. The problem of slicing policy determination is well addressed by many schemes in the literature, e.g. [12]–[14]. Slice enforcement, however, remains under-explored.

Slice enforcement needs to consider several requirements. First, slice isolation needs to be ensured so that the behavior of one slice does not cause performance degradation in another slice [5]. Thus, interference among different slices, i.e., inter-slice interference, needs to be avoided by resource separation between slices. Second, QoS of users in each slice must be satisfied even under the influence of channel fading and

Manuscript received 16 December 2021; revised 16 March 2022; accepted 11 April 2022. Date of publication 8 June 2022; date of current version 18 July 2022. *(Corresponding author: Xudong Wang.)*

The authors are with the UM-SJTU Joint Institute, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: wxudong@ieee.org).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/JSAC.2022.3180776.

Digital Object Identifier 10.1109/JSAC.2022.3180776

0733-8716 © 2022 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: University of Edinburgh. Downloaded on October 19,2026 at 21:07:55 UTC from IEEE Xplore. Restrictions apply.

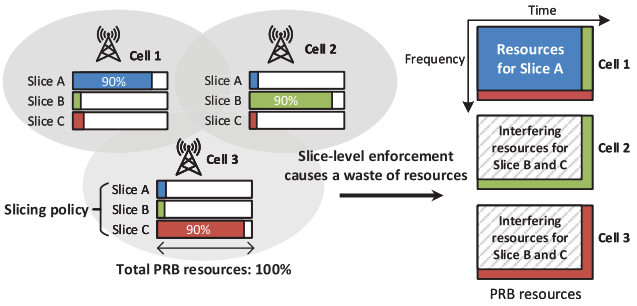


Fig. 2. Slice-level enforcement with unbalanced slicing policies in multi-cell RAN slicing.

interference. Third, the allocation of PRBs needs to conform to a predetermined slicing policy. This requirement is nontrivial, as slicing policies are periodically determined on a much larger timescale than a radio frame, thus requiring long-term conformance.

To satisfy these requirements, slice enforcement needs to be conducted at the link level rather than at the level of a whole slice. The reason is two-fold. First, slice-level enforcement is focused on the aggregated performance of the slice. It cannot guarantee fine-grained QoS performance of each user, while link-level enforcement can satisfy each user's demands via proper handling of fading and interference on the communication link of the user. Thus, it is desired to allocate resources directly to a link. Second, the number of users served by a slice varies from one cell to another, resulting in unbalanced slicing policies in different cells, as shown in Fig. 2. If slice enforcement is carried out per slice (without link-level scheduling) under unbalanced slicing policies, a great amount of resources will be wasted to ensure slice isolation. For example, in Fig. 2, if PRBs are allocated to slice A conforming to its slicing policy in cell 1, then most of these PRB resources (i.e., interfering resources in Fig. 2) cannot be reused in cells 2 and 3 to avoid inter-slice interference. To solve this problem, slice enforcement at the link level is indispensable.

Link-level slice enforcement makes the requirements of slice enforcement even more challenging, particularly in two aspects. First, slice isolation needs to be conducted at the link level. So far there exist two major types of resource separation: hard isolation and soft isolation. Hard isolation allocates dedicated PRBs to each slice in a network-wide manner but at a great sacrifice of spectral efficiency [10]. Soft isolation [11], however, only requires dedicated resources for a slice on interfering base stations. In other words, different slices can reuse the same PRBs unless they are deployed on interfering base stations. Apparently, soft isolation is preferred because of its higher spectrum efficiency. However, how to ensure soft isolation at the link level remains a challenging issue. Second, long-term conformance of slicing policies is closely related to short-term link-level behavior of slices in each radio frame. Thus, while evaluating long-term conformance of a network slice, link-level performance in each radio frame needs to be characterized, which is still an open problem.

To resolve the above challenges, a link-level slice enforcement scheme called *LinkSlice* is developed in this paper for a multi-cell network. It achieves fully soft, fine-grained slice isolation via link-level scheduling of PRBs for each slice. It satisfies QoS requirements demanded by different users. Moreover, *LinkSlice* cannot only ensure long-term (i.e., a number of radio frames) slicing policy conformance, but can also adapt flexibly to different patterns of slicing policy, either balanced or unbalanced.

The design of *LinkSlice* is elaborated as follows. In a multi-cell RAN, a central controller collects channel conditions and user demands from all the BSs, and conducts link-level slice enforcement over multiple air interfaces. To determine PRBs for each slice at the link level, an optimization problem is formulated as minimizing the amount of allocated resources over each period of slicing policy, subject to the constraints that ensure soft slice isolation, QoS guarantee, and slicing policy conformance under link interference among different cells. The optimization problem is hard to solve due to extremely high computation complexity. However, by converting the constraint of slicing policy conformance into a penalty term of the objective function, the optimization problem is transformed into a certain type of optimization problem that can be effectively solved via a deep reinforcement learning (DRL) based approach [15].

By leveraging the DRL-based approach, *LinkSlice* is then designed as an iterative two-stage slice enforcement scheme. More specifically, in the first stage, transmission rates in communication links of different slices are determined via DRL. Interference among links is encoded as an interference graph, and then a graph convolutional network (GCN) is applied to capture the relationships among interfering links. A graph pooling mechanism is further considered to generate a compact and informative state space for the DRL agent. In the second stage, after transmission rates are determined, PRB scheduling per radio frame is conducted for communication links of each slice. It can be solved by a heuristic algorithm called greedy slice enforcement with masking mechanism (G-SEM). The two stages work together iteratively until reaching convergence. By then, an effective link-level slice enforcement tool is obtained.

LinkSlice is evaluated through extensive simulations. Performance results show that it achieves a near-optimal solution to the link-level slice enforcement problem. It can reach convergence in a time interval much shorter than the changing periods of interference graph or user demands. Compared to the state-of-the-art scheme [11], *LinkSlice* improves network throughput by 18.5%, while ensuring users' QoS and maintaining long-term conformance of slicing policies. Moreover, *LinkSlice* handles slice isolation properly no matter if the slicing policies are balanced or unbalanced.

The main contributions of our method are summarized as follows:

- Fine-grained slice enforcement is conducted at the link level. It is formulated as an optimization problem that satisfies three key requirements of RAN slicing, i.e., soft isolation under both balanced and unbalanced slicing policies, users' QoS guarantee, and long-term conformance of slicing policies. The interference management

Authorized licensed use limited to: University of Edinburgh. Downloaded on October 19,2026 at 21:07:55 UTC from IEEE Xplore. Restrictions apply.

with link-level granularity is conducted in multi-cell RAN slicing.

- To tackle the coupling between the long-term objective and short-term scheduling, a DRL-based heuristic approach is developed for the slice enforcement problem. Without depending on the prior knowledge of the time-varying channels and interference conditions, the DRL-based heuristic approach adapts to the dynamic environment and achieves a near-optimal solution via online learning.
- Following the DRL-based heuristic approach, *LinkSlice* is designed as an iterative slice enforcement algorithm consisting of two stages. The first stage determines the transmission rates by capturing link interference via an interference-graph based GNN. Given the transmission rates, the second stage allocates PRBs for each slice by considering QoS requirements and soft slice isolation. *LinkSlice* highly outperforms existing schemes.

The rest of the paper is organized as follows. In Section II, related work is introduced. The system models of RAN slicing are presented in Section III. The link-level slice enforcement problem is formulated and transformed in Section IV. Afterward, the overview of the two-stage framework of *LinkSlice* is presented in Section V. In Section VI, the methodology and design details are presented. Performance evaluation of *LinkSlice* is conducted in Section VII. The paper is concluded in Section VIII.

II. Related Work

A. RAN Slicing

Due to its capability to enable efficient sharing of wireless network resources in 5G systems, RAN slicing has attracted increasing attention in recent years [9], [10], [12]–[14], [16]–[18]. In [18], a programmable, 3GPP-compliant network slicing architecture is designed, which spans from CN to RAN. It is focused on the function block and architecture design. A slice-aware RAN framework proposed in [17] includes a slice scheduler and a MAC scheduler per slice in each cell. This work is focused on the framework design rather than detailed scheduling schemes. In [13], RAN slicing is modeled as a congestion game and the slicing policy is determined for each slice in a distributed manner. In [19], a multi-armed-bandit-based (MAB) inter-slice scheduler allocates resources to multiple slices on a large timescale, which provides average latency and throughput guarantees for each slice in a single cell.

The above work is mainly focused on framework design and high-level scheduling. As a crucial step to map the logical planning of RAN slicing to the physical resources, the slice enforcement problem has also started to gain researchers' attention. In [16], frequency resources are allocated to the flows under the slice constraints in terms of aggregate rate targets. However, only a single base station is considered, and the algorithm cannot be directly extended to multi-cell scenarios. In [10], network-wide dedicated bandwidths are allocated to slices under two specific modes, eMBB and URLLC, to maximize the network operator's revenue while guaranteeing QoS satisfaction. This method assumes hard isolation among slices, and thus leaves a large space for spectral efficiency improvement. For multi-cell scenarios, a service provisioning framework is provided for RAN slicing [9], which jointly optimizes user admission and bandwidth allocation. However, this framework does not consider either the effect of slicing policies or slice isolation. In [11], slice-level enforcement is conducted to mitigate inter-slice interference. It is the first work to ensure soft slice isolation. However, without link-level scheduling, this scheme cannot guarantee QoS performance of each link or adapt to practical unbalanced slicing policies, as stated in Section I.

B. DRL Approaches for Network Slicing

Deep reinforcement learning (DRL) has been demonstrated to be a promising approach to solving sequential decision problems in many fields [15], [20]–[24]. DRL is applied in the field of network slicing in recent years [12], [25]–[32]. A joint network slicing and routing mechanism is introduced in [28]. This work leverages GCN-powered DRL to determine routing paths for each slice and conduct slice-level resource partitioning. In [27], a joint content caching and UE mode selection problem is tackled via DRL, which is considered a critical problem in fog-RAN slicing. A slice-level resource allocation and workload distribution algorithm is designed in [26] for delay-sensitive and delay-tolerant vehicular services. It leverages two-layer constrained DRL to adapt to dynamic vehicle traffic density. A deep Q network is utilized to intelligently conduct slice-level scheduling in a single cell in [12].

Although most of the existing work is focused on slice-level scheduling, some work applies DRL to fine-grained slice enforcement. A hierarchical DRL framework is proposed to conduct slicing configuration adaption and user-level radio resource allocation in [25]. Radio resources are allocated on a small timescale to optimize the QoS performance and spectrum efficiency in a single cell. In [32], an energy-efficient resource allocation algorithm is designed to jointly optimize slicing policy and slice enforcement. RB resources are reserved for each slice on a large timescale, while slice admission, power, and RB allocation are determined on a small timescale. However, only hard isolation is considered in [32], and there lacks interference management among links in both [32] and [25]. The channel scheduling among multiple slices is modeled as a stochastic game in [30] where A DRL-assisted online scheme is designed to conduct channel auction to maximize the network utility in multiple cells. Although user-level scheduling is realized, this work does not consider either long-term resource guarantee for each slice or users' QoS satisfaction.

In spite of the above related work, there lacks fine-grained link-level interference management in multi-cell RAN slicing. Such interference management is indispensable, as the existence of interference directly impacts slice isolation and users' QoS guarantee. Compared with the related work, our work is distinct in two aspects. First, a comprehensive framework is designed for multi-cell RAN slicing on two timescales.

Authorized licensed use limited to: University of Edinburgh. Downloaded on October 19,2026 at 21:07:55 UTC from IEEE Xplore. Restrictions apply.

This framework jointly tackles long-term slicing policy conformance, short-term interference management, and link-level QoS guarantee. Second, interference management is conducted with link-level granularity, where link interference is captured via an interference-graph based GNN.

III. System Model

A. 5G Multi-Cell RAN

A 5G multi-cell RAN is considered in this work. The network includes multiple distributed units (DU) and one central unit (CU, i.e., the central controller of the RAN). All the DUs are connected with the CU via F1 interface [5]. In the rest of the paper, the DU is referred to the base station (BS), and the CU is referred to the central controller. Let $\mathcal{B} = \{1, 2, \dots B\}$ denote the set of base stations. The central controller takes centralized scheduling decisions and sends them to the base stations. All the base stations reuse the radio resources in the RAN.

The set of admitted network slices in the RAN is denoted by $\mathcal{M} = \{1, 2, \dots M\}$. Let $\mathcal{U}$ be the set of users in the RAN. Each user is assumed to be associated with only one BS. Let $\mathcal{U}_{b,m}$ denote the set of users associated with BS b and slice m. The set of users associated with BS b is denoted as $\mathcal{U}_b = \cup_{m\in\mathcal{M}}\mathcal{U}_{b,m}$. Based on the signal strength, a user is predefined as a cell-edge user or cell-center user [33]–[35]. Let $\mathcal{U}_{b,\mathrm{e}}$, $\mathcal{U}_{b,\mathrm{c}}$ denote the sets of cell-edge users and cell-center users associated with BS b respectively. The sets of all cell-edge and cell-center users are denoted by $\mathcal{U}_\mathrm{e}$ and $\mathcal{U}_\mathrm{c}$ respectively, where $\mathcal{U}_\mathrm{e} = \cup_{b\in\mathcal{B}}\mathcal{U}_{b,\mathrm{e}}$, $\mathcal{U}_\mathrm{c} = \cup_{b\in\mathcal{B}}\mathcal{U}_{b,\mathrm{c}}$.

Time and frequency scheduling is conducted given the predetermined power levels P_u for user u. Cell-center and cell-edge users are allocated with different power levels [33]. Specifically,

$$P_u = \begin{cases} P_\mathrm{e}, & \forall u \in \mathcal{U}_\mathrm{e}, \\ P_\mathrm{c}, & \forall u \in \mathcal{U}_\mathrm{c}, \end{cases} \tag{1}$$

and $P_\mathrm{c} \geq P_\mathrm{e}$. We focus on the downlink transmission that adopts orthogonal frequency division multiple access (OFDMA). Let N_f denote the number of PRBs within one slot of a radio frame, and T_f denote the number of time slots. Thus, the available resources in one radio frame are divided into N PRBs with $N = N_\mathrm{f} \cdot T_\mathrm{f}$.

The wireless channel coefficient between the user u and BS b on PRB i is denoted by $h_{u,i}^{(b,m)}$ (the user u is associated with slice m). It is assumed that the channel remains the same within one PRB. Let P_N denote the noise power on each PRB. The actual transmission power of user u is denoted as $\tilde{p}_u$, which depends on PRB scheduling elaborated in Section IV-A. Thus, we derive the signal to interference and noise ratio (SINR) $\gamma_{u,i}^{(b,m)}$ of the user itself on each PRB:

$$\gamma_{u,i}^{(b,m)} = \frac{\tilde{p}_u h_{u,i}^{(b,m)}}{P_\mathrm{N} + \sum_{v\neq u, v\in\mathcal{U}} \tilde{p}_v h_{v,i}^{(b,n)}}. \tag{2}$$

Let $r_{u,i}^{(b,m)}$ denote the transmission rate for user u on PRB i. The transmission rate is chosen from a set of discrete values according to 3GPP standards [5]. First, the SINR is mapped to the channel quality indicator (CQI) based on the CQI table in [5]. Then, a modulation and coding scheme (MCS) is decided based on CQI. The rate $r_{u,i}^{(b,m)}$ is finally calculated based on the MCS chosen on this PRB. Let $l(\cdot)$ denote SINR to CQI mapping, and $f(\cdot)$ denote CQI to MCS mapping. Both functions are discrete step functions. Therefore, the transmission rate can be expressed as the function of SINR as follows:

$$r_{u,i}^{(b,m)} = f(l(\gamma_{u,i}^{(b,m)})). \tag{3}$$

To simplify interference modeling, the interfering links whose interference is below a certain threshold are ignored. Under this assumption, the inter-cell interference experienced by cell-center users is negligible [33], [34]. The cell-edge users, however, are prone to interference and performance degradation. Moreover, SINR $\gamma_{u,i}^{(b,m)}$ in equation (3) can only be obtained based on the results of PRB scheduling, and it cannot be measured in advance.

In the light of the above facts, nominal SNR is further defined as $\beta_{u,i}^{(b,m)}$ according to [36]. It is expressed as the ratio of transmit signal strength and noise:

$$\beta_{u,i}^{(b,m)} = \frac{P_u h_{u,i}^{(b,m)}}{P_\mathrm{N}}, \forall u \in \mathcal{U}, i = 1, \dots, N. \tag{4}$$

Thus, transmission rates for cell-center users are directly calculated using nominal SNR [33], [37]:

$$r_{u,i}^{(b,m)} = f(l(\beta_{u,i}^{(b,m)})), \forall u \in \mathcal{U}_{b,\mathrm{c}}, i = 1, \dots, N. \tag{5}$$

For cell-edge users, the SNR degradation caused by interference is expressed as a discrete value $\delta_u^{(b,m)}$, which is the number of levels below the nominal CQI level $l(\beta_{u,i}^{(b,m)})$. Here, we assume that SNR degradation for one user is homogeneous in interfering PRBs [37], while nominal SNR for each PRB can still be different and is dependent on time-varying channel conditions. Therefore, transmission rates for cell-edge users are computed as follows [33], [37]:

$$r_{u,i}^{(b,m)} = f(l(\beta_{u,i}^{(b,m)}) - \delta_u^{(b,m)}), \forall u \in \mathcal{U}_{b,\mathrm{e}}, i = 1, \dots, N. \tag{6}$$

Since nominal SNR values can be directly measured by the receiver, the determination of transmission rates for each user is converted to the determination of SNR degradation. Naturally, SNR degradation $\delta_u^{(b,m)}$ reflects the level of aggregated interference from other cells based on PRB scheduling. Furthermore, given the transmission rate $r_{u,i}^{(b,m)}$ assigned to user u, the SINR lower bound $\tilde{\gamma}_{u,i}^{(b,m)}$ can be calculated. Let function $F^{-1}(\cdot)$ denote the reverse mapping process as follows:

$$\tilde{\gamma}_{u,i}^{(b,m)} = F^{-1}(r_{u,i}^{(b,m)}). \tag{7}$$

Based on the SINR lower bound, the maximum tolerance interference $I_{u,i}^{(b,m)}$ is derived as follows:

$$I_{u,i}^{(b,m)} = \frac{\tilde{p}_u h_{u,i}^{(b,m)}}{\tilde{\gamma}_{u,i}^{(b,m)}} - P_\mathrm{N}. \tag{8}$$

Authorized licensed use limited to: University of Edinburgh. Downloaded on October 19,2026 at 21:07:55 UTC from IEEE Xplore. Restrictions apply.

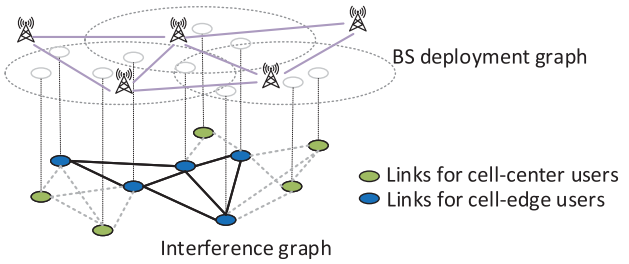


Fig. 3. Base station deployment and interference graph in multi-cell RAN.

B. Slice Service Model

The QoS requirement of slice m is considered to be the minimum data rate R_m for each radio frame. Let $\mathcal{W}_m = \cup_{b\in\mathcal{B}}\mathcal{U}_{b,m}$ denote the set of users who are associated with slice m across all base stations. The users in $\mathcal{W}_m$ are assumed to be homogeneous in the QoS requirement R_m.

During slice enforcement, the long-term slicing policy $p_{b,m}$ needs to be ensured. The range of $p_{b,m}$ is $[0, 1]$, which prescribes the percentage of resources reserved on BS b for admitted slice m [11]. Each slice is deployed on multiple base stations according to this predetermined slicing policy. The slicing policies are assigned periodically, where the slicing policy period is set as T radio frames and it can be much larger than the timescale of slice enforcement (i.e., one radio frame). Each slice needs to conform to the slicing policy during each slicing policy period T, i.e., the resource usage of the slice does not exceed the predetermined percentage of total resources. By conforming to the slicing policy, the traffic load variation of one slice does not affect other slices. Therefore, the slicing policy provides the quality of service on the slice level.

C. Interference Graph Model

Interference graphs are effective models to capture the interference among links [33], [38]–[43], as shown in Fig. 3. Let the undirected interference graph be $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ denotes the set of nodes, and $\mathcal{E}$ denotes the set of edges. The node V_u in $\mathcal{V}$ represents the downlink communication link between user u and its associated BS. The edge between nodes V_u and V_v represents the existence of interference between these two communication links. The estimated interference strength between two communication links is represented as a weight $w_{u,v}$ on its corresponding edge. There is no edge among the transmission links within the same cell, as no intra-cell interference is allowed. Let $\mathcal{V}(u)$ denote the neighboring (i.e., interfering) links of user u's link on the interference graph. Thus, the SINR is reformulated as follows:

$$\gamma_{u,i}^{(b,m)} = \frac{\tilde{p}_u h_{u,i}^{(b,m)}}{P_{\mathrm{N}} + \sum_{v\in\mathcal{V}(u)} \tilde{p}_v h_{v,i}^{(b,n)}}, \tag{9}$$

where $\tilde{p}_v h_{v,i}^{(b,n)}$ is related to $w_{u,v}$ based on PRB scheduling. Note that equation (9) is evolved from equation (2). Equation (2) is the general SINR expression that considers the potential interference from all the other users in the network, while equation (9) only retains the potential interference from neighboring nodes on the interference graph.

IV. Problem Formulation and Transformation

A. Problem Formulation

The fine-grained, link-level slice enforcement problem is studied in a multi-cell network. Slice enforcement is conducted for each radio frame t, which involves multiple types of entities, namely base stations, slices, and users. Given the predetermined long-term slicing policy $p_{b,m}$, PRBs on the involved base stations are allocated to each link. For the benefit of the physical network operator, once link-level QoS requirements are guaranteed, precious radio resources need to be saved as much as possible.

Furthermore, the requirement on slice isolation is stated as follows: interfering links associated with different slices are not allowed to use the same PRB resources for safety and privacy considerations [13]. Since soft slice isolation is conducted across multiple cells, the impact of inter-cell interference on the performance of isolation needs to be considered. For two different slices, if they are allocated with the same set of PRB resources across cells, there can be inter-slice interference caused by inter-cell interference. For a single slice, if it is allocated with the same set of PRB resources across cells, there can also be intra-slice interference caused by inter-cell interference. Different approaches are taken to tackle inter-slice interference and intra-slice interference. Inter-slice interference is avoided by link-level resource separation, which is indispensable due to the slice isolation requirement. However, there is no isolation issue on the physical layer for the same slice. If there exists intra-slice interference, one approach is allocating time/frequency resources to the interfering links to guarantee the user demands, and the other one is performing power allocation among intra-slice links to mitigate this interference. This paper is focused on the former approach.

Based on the above analysis, the objective of slice enforcement is to minimize the long-term PRB usage over each slicing policy period T, i.e., the number of consumed PRBs, for the whole RAN. In the meantime, the long-term slicing policy, the QoS requirement, and slice isolation have to be guaranteed. For the conformance of the slicing policy, the average PRB usage of each slice is smaller than the threshold prescribed by the slicing policy $p_{b,m}$ over each slicing policy period T. For the QoS requirement, in each radio frame t, a user subscribing to the service of slice m has the minimum data rate requirement denoted by R_m. Thus, the resources allocated to user u need to transmit at least $R_m\sigma$ bits in one radio frame, where σ denotes the time duration of one radio frame. Also, the intra-slice interference must be smaller than the maximum tolerance interference $I_{u,i}^{(b,m)}$ in equation (8). For slice isolation, the interfering links associated with different slices are not allowed to use the same PRB resources. Moreover, to avoid intra-cell interference, the transmission links in the same cell must not reuse PRBs.

Let $x_{u,i}^{(b,m)}(t)$ denote the PRB allocation indicator for user u associated with BS b and slice m on PRB i, during the

Authorized licensed use limited to: University of Edinburgh. Downloaded on October 19,2026 at 21:07:55 UTC from IEEE Xplore. Restrictions apply.

radio frame indexed by t. $x_{u,i}^{(b,m)}(t)$ equals 1 when PRB i is allocated to user u, and 0 otherwise. Based on the PRB allocation indicator, the actual transmit power can be expressed by $\tilde{p}_u = x_{u,i}^{(b,m)} P_u$, and the actual interfering power based on the weight $w_{u,v}$ on the interference graph can be denoted as $\tilde{p}_v h_{v,i}^{(b,n)} = x_{v,i}^{(d,n)} w_{u,v}$ (the user v is associated with BS d and slice n).

To simplify the notation, the radio frame index t is omitted for the following formulation except for explicit explanation. The slice enforcement problem is formulated as follows:

Problem 1 (P1):

$$\min_{x_{u,i}^{(b,m)}} \quad \frac{1}{T}\sum_{t=1}^{T}\left[\sum_{u\in\mathcal{U}}\sum_{i=1}^{N} x_{u,i}^{(b,m)}(t)\right], \tag{10}$$

$$\text{s.t.}: \quad \sum_{i=1}^{N} x_{u,i}^{(b,m)} r_{u,i}^{(b,m)} \geq R_m \sigma, \forall u \in \mathcal{U}_{b,m}, b \in \mathcal{B}, m \in \mathcal{M}, \tag{11}$$

$$\sum_{u\in\mathcal{U}_b} x_{u,i}^{(b,m)} \leq 1, \forall b \in \mathcal{B}, i = 1, \ldots, N, \tag{12}$$

$$x_{u,i}^{(b,m)} + \sum_{\substack{v\in\mathcal{V}(u),\\ n\neq m, d\neq b}} x_{v,i}^{(d,n)} \leq 1, \forall u \in \mathcal{U}, i = 1, \ldots, N, \tag{13}$$

$$x_{u,i}^{(b,m)} \sum_{v\in\mathcal{V}(u), d\neq b} w_{u,v} x_{v,i}^{(d,m)} \leq I_{u,i}^{(b,m)}, \quad \forall u \in \mathcal{U}, i = 1, \ldots, N, \tag{14}$$

$$\frac{1}{T}\sum_{t=1}^{T}\left[\sum_{u\in\mathcal{U}_{b,m}}\sum_{i=1}^{N} x_{u,i}^{(b,m)}(t)\right] \leq N p_{b,m}, \quad \forall b \in \mathcal{B}, m \in \mathcal{M}, \tag{15}$$

$$x_{u,i}^{(b,m)} \in \{0,1\}, i = 1, \ldots, N, \tag{16}$$

where objective (10) optimizes the network performance measured over each slicing policy period T; constraint (11) describes the QoS requirement; constraint (12) guarantees that there is no intra-cell interference; constraint (13) ensures the link-level soft slice isolation; constraint (14) guarantees that the intra-slice interference is below the maximum tolerance interference; constraint (15) guarantees the long-term enforcement of slicing policy over each slicing policy period T.

B. Problem Transformation

Slice enforcement is a decision-making problem over multiple time steps, i.e., radio frames, to maximize a long-term performance metric. This problem is extremely hard to solve considering the following perspectives. First, the objective function aims at minimizing the resource usage spanning multiple time steps, while resources are allocated per time step. Second, there is a mixture of long-term constraints and short-term constraints. Third, there is a coupling effect among the long-term objective, the long-term constraints, and the short-term constraints. Last but not least, the computation complexity of even a sub-problem is NP-hard as we will demonstrate in Section V.

Deep reinforcement learning (DRL) is a perfect candidate to handle such a long-term optimization problem, where the agent in DRL aims to maximize the long-term rewards received from the environment through multi-step actions. Thus, DRL can tackle the coupling between the long-term objective and short-term decision in slice enforcement problem. Furthermore, maximizing the long-term network performance requires future information on time-varying channels and interference conditions, which needs complicated mathematical modeling. The model-free DRL algorithm does not require any prior information on the environment. The DRL agent adapts to the dynamic environment and learns the optimal policy by interacting with the environment. Therefore, mathematical modeling and prediction are avoided.

However, when DRL is applied to solve problem P1, one of the obstacles is that there is a long-term constraint (15) on the slicing policy conformance. In this case, it is difficult to decompose problem P1 into multiple separate time steps. To tackle this issue, we define the penalty term

$$L_{b,m}(t) = \max\{0, \sum_{u\in\mathcal{U}_{b,m}}\sum_{i=1}^{N} x_{u,i}^{(b,m)}(t) - N p_{b,m}\}, \tag{17}$$

for frame t to transform the constraint (15) into the objective function. If there is no violation of the slicing policy, the penalty term $L_{b,m}$ equals zero. If slicing policy violation takes place, the corresponding penalty is exerted on the objective function. Let λ denote the penalty coefficient associated with this penalty term. We thus present an approximate problem P2 to problem P1:

Problem 2 (P2):

$$\min_{x_{u,t}^{(b,m)}} \quad \frac{1}{T}\sum_{t=1}^{T}\left[\sum_{u\in\mathcal{U}}\sum_{i=1}^{N} x_{u,i}^{(b,m)}(t) + \lambda \sum_{b\in\mathcal{B}}\sum_{m\in\mathcal{M}} L_{b,m}(t)\right], \tag{18}$$

$$\text{s.t.}: \ \text{Eqs.}(11)-(14),(16).$$

Although problem P2 is an approximated version of P1, it is obvious that the optimal solution to P1 is also the optimal solution to P2. We can also prove that problem P2 can be guided by DRL reward design to approach the optimal solution to P1. With a large enough penalty coefficient λ, the DRL agent can always receive larger rewards when the actions generated by DRL fall into the feasible region of P1 (the reward design is elaborated in Section VI-A). This is shown as follows.

Theorem 1: When the local optimal solution to problem P2 falls out of the feasible region $\mathcal{F}$ of problem P1, there exists $\lambda \in \mathbb{R}$ such that, the optimal objective value of P2 must be larger than that of P1.

Proof: The detailed proof is given in Appendix. ■

V. LinkSlice Overview

As discussed in the Section IV, DRL can be utilized to handle multi-step optimization problem. However, if DRL is directly applied to solve problem P2 by generating the PRB allocation indicators as the actions, it is difficult for the DRL agent to learn the desired policy in practice. The reasons lie in two aspects: 1) since there exist a large number of

Authorized licensed use limited to: University of Edinburgh. Downloaded on October 19,2026 at 21:07:55 UTC from IEEE Xplore. Restrictions apply.

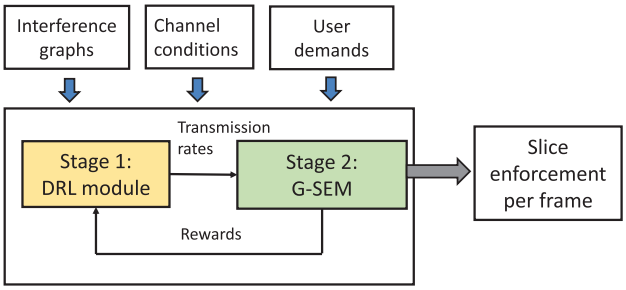


Fig. 4. *LinkSlice* overview: a DRL-based heuristic solution for slice enforcement.

constraints in problem P2, guiding the DRL agent towards feasible actions through reward engineering is notoriously difficult [44]; 2) even though DRL can reach feasible actions, converging to a near-optimal solution is still challenging, as the combinatorial nature of P2 leads to a large discrete action space [45]. Therefore, the direct application of DRL is not practical.

To this end, we design a two-stage framework: DRL is applied to obtain the transmission rates in the first stage, and slice enforcement is conducted in the second stage given the transmission rates. Especially, for the second stage, the remaining PRB scheduling problem for each radio frame is still computationally hard. The following sub-problem needs to be solved:

Problem 3 (P3):

$$\min_{x_{u,i}^{(b,m)}} \quad \sum_{u\in\mathcal{U}}\sum_{i=1}^{N} x_{u,i}^{(b,m)} + \lambda \sum_{b\in\mathcal{B}}\sum_{m\in\mathcal{M}} L_{b,m},$$
$$\text{s.t. :} \quad \text{Eqs.}(11)-(14),(16). \tag{19}$$

Problem P3 is an integer quadratic-constrained programming problem (IQCP). The variable space of P3 grows exponentially with the numbers of users and of PRBs. A brief proof of NP-hard property of P3 is given by theorem 2. It is computationally expensive to obtain the optimal solution as the size of the network increases. Therefore, a low-complexity heuristic algorithm named greedy slice enforcement with masking (G-SEM) is proposed to solve problem P3.

Theorem 2: Problem P3 is NP-hard.

Proof: The detailed proof is given in Appendix. ■

The two-stage framework of *LinkSlice* is shown in Fig. 4. The interference graphs, channel conditions, and user demands are collected for the two stages. In the first stage, the DRL outputs the SNR degradation, which is then transformed into transmission rates through the mapping function in Equation (6). In the second stage, given the transmission rates for each link in each radio frame, a low-complexity greedy algorithm G-SEM is designed to obtain the near-optimal solution to problem P3. G-SEM conducts slice enforcement per radio frame. The rewards are collected from the second stage and fed back into the first stage. The two stages work together iteratively until reaching convergence.

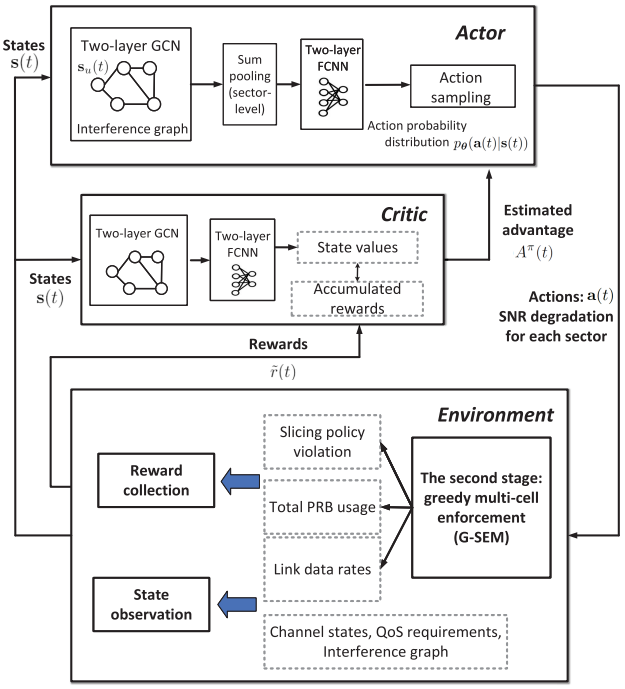


Fig. 5. *LinkSlice* detailed design: two-stage slice enforcement.

VI. Two-Stage Slice Enforcement: A DRL-Based Heuristic Solution

Compared with the standard DRL architecture, *LinkSlice* is unique in two aspects. First, the actor and critic networks contain embedded GCNs to capture the interference among communication links. Then, a pooling mechanism is leveraged to reduce the size of action space. Second, the optimization algorithm G-SEM is embedded into the DRL environment to conduct PRB scheduling. The detailed blocks are shown in Fig. 5. The design of *LinkSlice* is elaborated in this section.

A. The First Stage: A DRL Approach for Interference-Aware Rate Determination

A DRL problem is generally based on a Markov decision process (MDP) consisting of the state space, action space, reward space, transition probability set, and initial state distribution, i.e., $\{\mathcal{S},\mathcal{A},\mathcal{R},\mathcal{P},p_0\}$. The overall architecture and detailed design of DRL used in *LinkSlice* are presented in the following.

1) Overall Architecture of DRL: As illustrated in Fig. 5, an actor-critic architecture is utilized. Considering each radio frame as a time step in DRL, the global states $\mathbf{s}(t)$, actions $\mathbf{a}(t)$ and instantaneous rewards $\tilde{r}(t)$ are modeled for the task of SNR degradation estimation. The central controller in the RAN is viewed as the global agent. At the start of each time step, the DRL agent (i.e., the actor and critic) observes the states $\mathbf{s}(t)$ for each node (i.e., one communication link between a user and its associated BS) on the interference graph. The states are then embedded by a two-layer GCN and sum-pooled. The pooled embeddings are fed into a two-layer fully-connected neural network (FCNN), which outputs the probability distribution of actions after SoftMax activation. The actor then samples the

Authorized licensed use limited to: University of Edinburgh. Downloaded on October 19,2026 at 21:07:55 UTC from IEEE Xplore. Restrictions apply.

actions according to the probability distribution $\pi(\mathbf{a}|\mathbf{s})$ (i.e., the policy). After interacting with the environment, the agent collects rewards to update both the actor and critic networks.

In the following part, we present the design of state, action, reward, and the training algorithm of DRL.

2) State: The node state $s_u(t)$ in time step t consists of three parts, including the channel condition, the link data rate in the last time step, and the gap between this link data rate and QoS requirement. For clarity, The channel condition is denoted by a vector of nominal SNR $\boldsymbol{\beta}_u(t)$. Each element of $\boldsymbol{\beta}_u(t)$ is a nominal SNR measured on one PRB's bandwidth. The link data rate during the last frame $\tilde{R}_u(t-1)$ can be computed by adding the transmission bits on each PRB together. The gap between the link's QoS requirement and the actual data rate describes the level of QoS satisfaction, which is denoted as $\Delta_{\mathrm{R},u}(t-1) = \tilde{R}_u(t-1) - R_m$ (the user u is associated with slice m). Based on these three parts, the node state is formulated as

$$\mathbf{s}_u(t) = \left[\boldsymbol{\beta}_u(t), \tilde{R}_u(t-1), \Delta_{\mathrm{R},u}(t-1)\right]^{\mathsf{T}}, \tag{20}$$

where $\tilde{R}_u(t-1)$ is formulated as

$$\tilde{R}_u(t-1) = \sum_{i=1}^{N} x_{u,i}^{(b,m)}(t-1) r_{u,i}^{(b,m)}(t-1). \tag{21}$$

The state $\mathbf{s}(t)$ observed by the agent consists of all the node states, which is presented as

$$\mathbf{s}(t) = \left[\mathbf{s}_1(t), \mathbf{s}_2(t), \ldots, \mathbf{s}_U(t)\right]^{\mathsf{T}}. \tag{22}$$

3) State Aggregation and Pooling: Given all the node states in $\mathbf{s}(t)$, one straightforward way is to concatenate them to generate a long vector that is used as the input of the DRL networks. However, there are two obvious drawbacks of this design. First, it is not scalable with increasing users. When the number of users increases, the dimension of network inputs also increases. Second, this design ignores the relationships among the interfering links encoded in the interference graph.

Graph neural networks (GNN) [46]–[49] are proposed to provide representations for the graph data structure. The GNN embeds the node feature and its structural information into the embedding space. In *LinkSlice*, an interference-aware DRL network is designed, where GCNs are utilized to capture the relationships among nodes and embed the interference graphs into the states of DRL. Let $\boldsymbol{W}_{\mathrm{G}(1)}$ and $\boldsymbol{W}_{\mathrm{G}(2)}$ denote the parameters of two layer GCNs respectively. Let d_s denote the dimension of the original node state, and d_l denote the dimension of the embedded node state, and the parameter matrices $\boldsymbol{W}_{\mathrm{G}(1)}$, $\boldsymbol{W}_{\mathrm{G}(2)}$ have the dimensions $d_s \times d_{\mathrm{G}}$, $d_{\mathrm{G}} \times d_l$ (let d_{G} denote the dimension of the hidden layer). The aggregation weights $w_{u,v}$ and the neighboring set $\mathcal{V}(u)$ of each node u are provided by the given interference graph. Let g_u denote the degree of user u in the interference graph. Two GCN layers are stacked for generating state embeddings $\tilde{\mathbf{s}}_u(t)$ for each node u, which captures the states of one-hop and two-hop neighboring nodes. The weighted aggregation process in the two GCN layers is presented as follows:

$$\tilde{\mathbf{s}}_u(t) = \sum_{v \in \mathcal{V}(u) \cup \{u\}} \frac{w_{u,v}}{\sqrt{g_u g_v}} \sigma_{\mathrm{R}}\left[\sigma_{\mathrm{R}}(\mathbf{s}_u(t)^{\mathsf{T}} \boldsymbol{W}_{\mathrm{G}(1)}) \boldsymbol{W}_{\mathrm{G}(2)}\right], \tag{23}$$

where $\sigma_{\mathrm{R}}(\cdot)$ is the ReLU activation function. During the state aggregation stage, the agent collects the states of all the nodes on the interference graph and conducts state aggregation via GCNs. The aggregated embeddings are represented by a matrix $\tilde{\boldsymbol{S}}$ (with time step t omitted for simplicity of notation) with the dimension $U \times d_l$. Each row of $\tilde{\boldsymbol{S}}$ corresponds to one node's embedding.

A graph pooling mechanism is added after state aggregation. Each cell is uniformly divided into N_{s} sectors. For example, the cell can be sectorized into six sectors, with $\pi/3$ radians each. The total number of sectors is denoted as K with $K = N_s B$ (B denotes the number of cells). In sector k, the aggregated embeddings are further sum pooled to generate one representation for sector k. The pooled embeddings are denoted by $\tilde{\boldsymbol{S}}_{\mathrm{P}}$ with the dimension $K \times d_l$ (with time step t omitted). Each row of $\tilde{\boldsymbol{S}}_{\mathrm{P}}$ corresponds to one sector's embedding.

The "sector" in our design is a virtual concept for graph pooling, instead of "cell sector" from directional antennas. The sector determines the range of sum pooling. It is practical to sum pool the node embeddings in the same sector, as the cell-edge users in the same sector have similar interference topologies that lead to similar SNR degradation. Although the sectorization scheme causes inaccuracy to the interference modeling that will impact the network performance, it can shrink the size of action space significantly. Instead of outputting actions for each node, the actor network can output actions for each sector, as elaborated in the following design of action space. The impact of the sectorization scheme on *LinkSlice* is evaluated in Section VII-D.

4) Action: The pooled embeddings are then fed into a two-layer FCNN. Let d_a denote the dimension of output layer of FCNNs, i.e., the number of actions that can be chosen from. The two-layer FCNN parameters are represented by two matrices $\boldsymbol{W}_{\mathrm{F}(1)}$, $\boldsymbol{W}_{\mathrm{F}(2)}$ with the dimensions $d_l \times d_{\mathrm{F}}$, $d_{\mathrm{F}} \times d_a$ (let d_{F} denote the dimension of the hidden layer). The action probability distribution output by the FCNNs is denoted as $\sigma_{\mathrm{S}}\left[\sigma_{\mathrm{R}}(\tilde{\boldsymbol{S}}_{\mathrm{P}} \boldsymbol{W}_{\mathrm{F}(1)}) \boldsymbol{W}_{\mathrm{F}(2)}\right]$, with the dimension $K \times d_a$ (let $\sigma_{\mathrm{R}}(\cdot)$ denote the ReLU activation function, and $\sigma_{\mathrm{S}}(\cdot)$ denote the SoftMax activation function). This process should be noted with two features. First, the action is conducted for each sector instead of each node, and all the nodes in the same sector share the selected action. Second, the embeddings for each sector are stacked into $\tilde{\boldsymbol{S}}_{\mathrm{P}}$, and FCNN parameters are shared among all the sectors, as revealed in the matrix multiplication in $\sigma_{\mathrm{S}}\left[\sigma_{\mathrm{R}}(\tilde{\boldsymbol{S}}_{\mathrm{P}} \boldsymbol{W}_{\mathrm{F}(1)}) \boldsymbol{W}_{\mathrm{F}(2)}\right]$ where $\boldsymbol{W}_{\mathrm{F}(1)}$, $\boldsymbol{W}_{\mathrm{F}(2)}$ are operated on each row of $\tilde{\boldsymbol{S}}_{\mathrm{P}}$. This design is similar to graph convolutional reinforcement learning proposed in [49].

Since the graph pooling based on the sectorization scheme is leveraged, the actions are sampled for each sector instead of each node. Therefore, the joint action space is shrunk from $(d_a)^U$ to $(d_a)^K$, with K equal to the number of sectors. The

Authorized licensed use limited to: University of Edinburgh. Downloaded on October 19,2026 at 21:07:55 UTC from IEEE Xplore. Restrictions apply.

action taken by the agent $\mathbf{a}(t)$ consists of actions sampled for each sector. This is presented as

$$\mathbf{a}(t) = [a_1(t), a_2(t), \ldots, a_K(t)], \tag{24}$$

where $a_k(t)$ is defined as the SNR degradation for sector k sampled from d_a possible discrete values. Originally, the SNR degradation level $\delta_u^{(b,m)}$ is defined on the link level, which describes the channel degradation for each link between user u and its associated BS b. After the pooling mechanism is introduced, the SNR degradation level selected for each sector describes the average channel degradation for all the cell-edge links in this sector. Given the SNR degradation level, the transmission rate is determined based on equation (6).

5) Reward: The long-term objective in problem P2 can be naturally mapped to the accumulated reward in DRL design. However, during the learning stage, it is possible that the second stage is not feasible (i.e., the users' QoS requirements cannot be satisfied) given the transmission rates from the first stage, as shown in Section VI-B. To solve this problem, when the DRL agent falls in the infeasible status, an extremely large negative reward is imposed on the agent. Thus, the infeasible status can be efficiently avoided during DRL training.

The instantaneous reward $\tilde{r}(t)$ for each time step t consists of two parts: the objective value in one step of problem P2 (including network-wide PRB usage and slicing policy violation), and the negative reward for infeasible status. Therefore, we have

$$\tilde{r}(t) = -\mathbb{1}\{\mathcal{H} \neq \emptyset\} W(t) - \mathbb{1}\{\mathcal{H} = \emptyset\}\mu, \tag{25}$$

$$W(t) = \sum_{u\in\mathcal{U}} \sum_{i=1}^{N} x_{u,i}^{(b,m)}(t) + \lambda \sum_{b\in\mathcal{B}} \sum_{m\in\mathcal{M}} L_{b,m}(t), \tag{26}$$

$$= \sum_{b\in\mathcal{B}} \left[\sum_{u\ \in\mathcal{U}_b} \sum_{i=1}^{N} x_{u,i}^{(b,m)}(t) \ + \ \lambda \sum_{m\in\mathcal{M}} L_{b,m}(t) \right], \tag{27}$$

where $\mathcal{H}$ denotes the feasible set of the second stage (i.e., problem P3), $-\mathbb{1}\{\mathcal{H} = \emptyset\}\mu$ denotes the penalty for the infeasibility of the second stage (the indicator function $\mathbb{1}$ of an event takes value 1 when the event is true and value 0 otherwise). In practice, μ is set as the maximum value of $W(t)$ collected in the environment.

6) Training Algorithm: The DRL agent with stochastic policies is considered: a policy $\pi(\mathbf{a}|\mathbf{s}) : \mathcal{S} \rightarrow P(\mathcal{A})$ represents the mapping from state space $\mathcal{S}$ to probability distributions over action space $\mathcal{A}$. The policy is parameterized by the actor and critic networks, denoted as $\boldsymbol{\theta}_{\mathrm{A}}$ and $\boldsymbol{\theta}_{\mathrm{V}}$ respectively. Both the actor and critic networks contain two layer GCNs for node embedding, and two layer FCNNs. Note that the output dimension of the critic network is one. The actor and critic networks are trained separately. The action taken by the agent leads to an instantaneous reward $\tilde{r}(t) \in \mathbb{R}$.

The learning process of DRL is based on the policy gradient algorithm. The goal is to maximize the expected long-term reward $\mathbb{E}\left[\sum_{t=1}^{T} \tilde{r}(t)\right]$ over one episode (i.e., T timesteps). Generalized advantage function (GAE) [50] is estimated as

$$A^{\pi}(t) := \tilde{r}(\mathbf{s}(t), \mathbf{a}(t)) + V^{\pi}(\mathbf{s}(t+1)) - V^{\pi}(\mathbf{s}(t)). \tag{28}$$

The advantage function $A^{\pi}(t)$ under policy π reveals how much the current action $\mathbf{a}(t)$ is better than average given the current state $\mathbf{s}(t)$. The value function $V^{\pi}(t)$ is the discounted reward-to-go approximated by the critic network as follows:

$$V^{\pi}(\mathbf{s}(t)) = \mathbb{E}\left[\sum_{t=t'}^{T} \gamma^{t'-t} \tilde{r}(\mathbf{s}(t'), \mathbf{a}(t'))\right], \tag{29}$$

where γ is the discounted factor.

Once the advantage is estimated, the policy gradient algorithm is applied to update the actor network. The objective of optimizing DRL's policy is:

$$\max_{\boldsymbol{\theta}} \mathbb{E}\left[\sum_{t=1}^{T} p_{\theta}(\mathbf{a}(t)|\mathbf{s}(t)) A^{\pi}(t)\right], \tag{30}$$

where $p_{\boldsymbol{\theta}}(\mathbf{a}(t)|\mathbf{s}(t))$ is the probability of taking action $\mathbf{a}(t)$ based on the policy $\pi_{\boldsymbol{\theta}}(\mathbf{a}|\mathbf{s})$.

The gradient of the objective function is taken as follows:

$$\nabla \mathbb{E}\left[\sum_{t=1}^{T} p_{\boldsymbol{\theta}}(\mathbf{a}(t)|\mathbf{s}(t)) A^{\pi}(t)\right] \tag{31}$$

$$= \mathbb{E}\left[\sum_{t=1}^{T} A^{\pi}(t) \nabla \log(p_{\boldsymbol{\theta}}(\mathbf{a}(t)|\mathbf{s}(t)))\right]. \tag{32}$$

In the first stage, the DRL module outputs the SNR degradation as actions, which are then converted to transmission rates via equation (6). Given the transmission rates generated by the DRL module, a greedy priority-based algorithm is designed to conduct slice enforcement per radio frame, which is elaborated in the following part.

Theorem 3: The complexity of greedy multi-cell enforcement algorithm is $O(B \log B + N^2 + U \log \frac{U}{B} + UN \log N + BN)$, where B, U, and N represent the number of BSs, users, and PRBs, respectively.

Proof: The proof is given in Appendix. ∎

B. The Second Stage: Multi-Cell Slice Enforcement Through Link-Level Scheduling

When slice enforcement is conducted over multi-cell air interfaces, two base stations whose coverage regions do not overlap with each other can enforce slices independently. The BS coverage, i.e., BS topology, is known in advance [33], [51].

A greedy slice enforcement algorithm with masking mechanism (G-SEM) is designed with three main blocks. The first block of G-SEM identifies the clusters of BSs that can enforce slices in parallel, which is achieved by Welsh–Powell algorithm [52]. Welsh–Powell algorithm is a greedy graph coloring algorithm based on the degrees of nodes.

For the BSs assigned with the same color, slice enforcement can be conducted in parallel. In the second block, a priority-based algorithm is designed for slice enforcement in a single cell. The allocation priority of users is determined by the following rule. First, the cell-edge users have a higher priority than the cell-center users. Then, for both the cell-edge and cell-center users, those with higher QoS requirements are assigned higher priorities. Last, for the users with the same QoS

Authorized licensed use limited to: University of Edinburgh. Downloaded on October 19,2026 at 21:07:55 UTC from IEEE Xplore. Restrictions apply.

Algorithm 1 G-SEM: Greedy Slice Enforcement With Masking Mechanism

Input: The estimated transmission rates $r_{u,i}^{(b,m)}$ output by DRL, the interference graph $\mathcal{G}$, long-term slicing policies $\mathcal{P}$, channel conditions of links $h_{u,i}^{(b,m)}$, the BS topology.
Initialization:
Identify the sets of base stations $\{\mathcal{B}_1, \ldots, \mathcal{B}_C\}$ that operate in parallel by applying Welsh-Powell algorithm on the BS topology. Let $\{1, \ldots, C\}$ denote the index of these sets.
for $c = 1$ to C **do**
  Conduct slice enforcement in parallel for base stations in the set $\mathcal{B}_c$;
  for $b \in \mathcal{B}_c$ **do**
    for $u \in \mathcal{U}_{b,\text{edge}}$ **do**
      Identify free PRBs for each user via the masking mechanism.
    end for
  end for
  for $b \in \mathcal{B}_c$ **do**
    Generate the scheduling priority for each user in the set $\mathcal{U}_b$;
    for $u \in \mathcal{U}_b$ **do**
      Sort the free PRBs based on transmission rates $r_{u,i}^{(b,m)}$ in the descending order;
      Allocate free PRBs according to the above order until each user's QoS requirement is satisfied.
    end for
  end for
end for
if this is the training stage of *LinkSlice* **then**
  Collect the results of slice enforcement and generate the instantaneous reward $\tilde{r}$.
end if
Output: A PRB allocation scheme $\mathcal{X}$ for slice enforcement.

Algorithm 2 : Two-Stage Slice Enforcement Algorithm

Input: The interference graph $\mathcal{G}$, the long-term slicing policies $\mathcal{P}$.
Initialization: Initialize the actor parameters $\boldsymbol{\theta}_{\mathrm{A}}$, the critic parameters $\boldsymbol{\theta}_{\mathrm{V}}$, the number of episodes K, the number of time steps in each episode T.
for $k = 1$ to K **do**
  Clear the replay buffer for the critic;
  for $t = 1$ to T **do**
    The actor network generates the action, i.e., the estimation of SNR degradation: $\mathbf{a}(t) \leftarrow$ the action sampled by $\log p_{\boldsymbol{\theta}_{\mathrm{A}}}(\mathbf{a}|\mathbf{s})$;
    Convert the SNR degradation into transmission rates according to equation (6);
    Run the heuristic algorithm G-SEM to obtain PRB allocation results for the current time step;
    Collect the instantaneous reward $\tilde{r}(t)$ from the environment according to equations (25)(26)(27);
    Compute the state value $v \leftarrow V^{\pi}(\mathbf{s}, \mathcal{G}; \boldsymbol{\theta}_{\mathrm{V}})$;
    Store the tuple $(\log p_{\boldsymbol{\theta}_{\mathrm{A}}}(\mathbf{a}|\mathbf{s}), \mathbf{s}, \mathcal{G}, \mathbf{a}, v, r)$ in the replay buffer;
    Update the state $\mathbf{s}(t)$ to the next state $\mathbf{s}(t+1)$;
    Reset the gradients $d\boldsymbol{\theta}_{\mathrm{A}} \leftarrow 0, d\boldsymbol{\theta}_{\mathrm{V}} \leftarrow 0$.
  end for
  $\mathrm{Loss}_{\mathrm{a}} = \sum_{t=1}^{T} A^{\pi}(t) \log p(\mathbf{a}(t)|\mathbf{s}(t))$;
  $\mathrm{Loss}_{\mathrm{v}} = \sum_{t=1}^{T} \left[\sum_{t'=t}^{T} \gamma^{t'-t} \tilde{r}(t') - V^{\pi}(\mathbf{s}(t))\right]^2$;
  Update $\boldsymbol{\theta}_{\mathrm{A}}$ using gradients wrt. Loss_a;
  Update $\boldsymbol{\theta}_{\mathrm{V}}$ using gradients wrt. Loss_v.
end for
if users' QoS requirements cannot be satisfied **then**
  Conduct user admission control, i.e., discard the unsatisfied users and start a new round of learning.
end if
Output: A well-trained PRB allocation scheme for slice enforcement.

requirements, i.e., associated with the same slice, those with larger degrees on the interference graph are assigned higher priorities. Once the user priorities are sorted, slice enforcement is carried out sequentially. For each user, the PRBs that can be allocated are sorted based on the transmission rates in the descending order, and PRBs are then allocated according to this order until each user's QoS requirement is satisfied.

The third block of G-SEM is a masking mechanism, which resolves the dependence among BSs with different colors. Taking one cell as an example, we define free PRBs for each link between a user and its associated BS: free PRBs are those that can be allocated to the user conforming to the predetermined transmission rates and ensuring slice isolation at the same time. The key idea of the masking mechanism is to identify all the free PRBs for each user. For cell-center users which are interference-free, all the PRBs are counted as free PRBs. However, it is not the case with cell-edge users.

How to identify free PRBs for a particular cell-edge user is illustrated as follows. A cell-edge user's link is deemed as the central node. The one-hop neighbors of the central node is traversed on the interference graph. If the neighbors are associated with different slices from the central node, we mask the PRBs that are already allocated to these neighbors, i.e., directly set the PRB indicators to zero. If the neighbors are with the same slice as the central node, the masking mechanism is more delicate. We check the interference level on each PRB for both the central node and its neighbors, to see whether the interference will exceed the maximum tolerance interference (equation (8)) on that PRB. This is equivalent to ensuring that there is no violation of the predetermined transmission rates. The PRB allocation indicator is masked to be zero if the interference on that PRB is intolerable. The masking mechanism identifies all the free PRBs that can be allocated in the second block of *LinkSlice*.

The whole procedure of G-SEM is shown in algorithm 1. The three blocks designed in G-SEM efficiently solve the multi-cell slice enforcement problem with a polynomial time complexity, which is analyzed in theorem 3. By prioritizing links and PRBs, G-SEM reduces the number of PRBs as much as possible for each slice, and thus reduces the overall

Authorized licensed use limited to: University of Edinburgh. Downloaded on October 19,2026 at 21:07:55 UTC from IEEE Xplore. Restrictions apply.

PRB usage and slicing policy violation. G-SEM achieves a near-optimal solution to problem P3, which is evaluated in Section VII.

C. Overall Algorithm of LinkSlice

As shown in algorithm 2, the overall algorithm of two-stage slice enforcement is presented. The DRL agent is trained with each episode equal to the slicing policy period T, and the next episode is obtained by sliding the entire policy period by one radio frame. During the training process of K episodes, the actor observes the states and generates actions sampled for each sector. Actions, i.e., estimated SNR degradation, are converted to transmission rates, which are the given inputs of the second stage. G-SEM algorithm is executed to obtain PRB allocation schemes. Rewards are derived based on PRB scheduling for each frame. The DRL algorithm collects the rewards to update actor and critic networks. After K episodes of training, the two stages work together iteratively until reaching convergence. By then, an effective link-level slice enforcement tool is obtained.

At the end of DRL training, if the QoS requirement of some users cannot be satisfied, then the network is considered overloaded, and user admission needs to be taken into account. In this case, the algorithm will automatically discard the users whose data rates are below their QoS requirements and start a new round of slice enforcement. This admission control part is incorporated into algorithm 2 for the integrity of *LinkSlice*.

VII. Performance Evaluation

A. Simulation Setup

In this section, the performance of *LinkSlice* is evaluated via extensive simulations. A multi-cell RAN aligning with 3GPP standards [5], [53] is simulated. A frequency division duplexing (FDD) system is considered, and this paper is focused on downlink scenarios. In the simulation, 20 MHz bandwidth is divided into 612 subcarriers with the subcarrier spacing set as 30 kHz. The radio frame size is 10 ms, consisting of 20 slots each with 0.5 ms. The total number of PRBs per slot is 51, and thus the number of PRBs within each radio frame is 1020 (i.e., $N = 1020$).

For a downlink channel from a BS to its associated users, the large-scale fading follows the UMi-Street canyon model in [53], and the small-scale fading is Rician fading. The coherence time is assumed to be comparable with the length of one radio frame. There exists no frequency selectivity within one PRB's bandwidth. Modulation and coding schemes (MCS) are selected based on the standard link adaptation mechanism, and the CQI table and MCS index tables are specified in TS 38.214 [54].

Users are distributed over each cell following uniform distribution unless specified differently. Users are differentiated by data rates (i.e., the metric of QoS requirement). In our simulations, four types of data rates, i.e., 100 kbps, 500 kbps, 1 Mbps, and 5 Mbps, are considered to represent services with low, medium, and high data rate requirements. User density is represented by the number of users per cell. By increasing user density, the traffic load in the network increases.

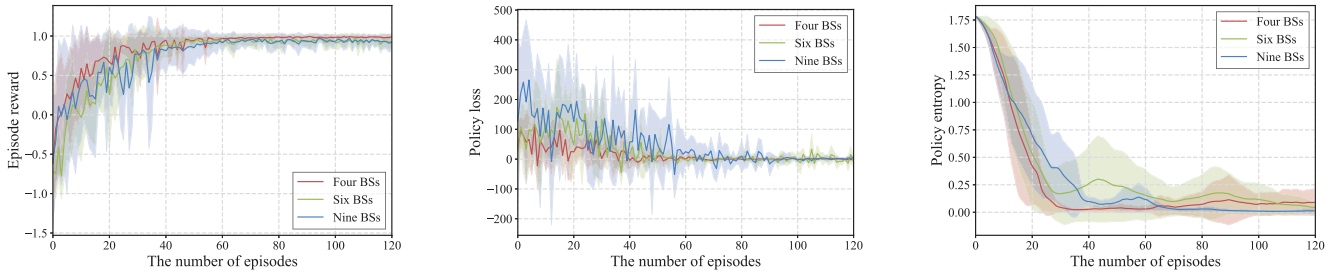
There are four types of slices in the simulations, and without loss of generality, one slice instance is considered in each type of slice. Within one slice instance, the user type is assumed to be homogeneous in data rates. Based on the previous analysis in section III, the slicing policy on a BS, e.g. BS b, is represented by a four-dimension tuple, i.e., $\mathbf{p_b} = [p_{b,1}, p_{b,2}, p_{b,3}, p_{b,4}]$. Since slicing policies can be unbalanced, a parameter q_b called unbalance index is defined to quantify the level of unbalance as follows. Considering BS b as a central BS, the sum of Euclidean distances between its slicing policy vector $\mathbf{p}_b$ and that of its neighboring base stations is determined and then normalized. Thus, $q_b = \sum_{d\in\mathcal{N}(b,m)} \|\mathbf{p_b} - \mathbf{p_d}\|/(\sqrt{2}\,|\mathcal{N}(b,m)|)$, where $\mathcal{N}(b,m)$ is the set of neighboring base stations of BS b, and $|\mathcal{N}(b,m)|$ is the cardinality of the set. Apparently, a higher q_b indicates more unbalanced in slicing policies. Network-wide unbalance index is the average of all the base stations' unbalance indices. In the simulations, six different levels of network-wide unbalance index are considered, i.e., $0, 0.2, 0.4, 0.6, 0.8, 1.0$. The unbalance index is set to 0.8 if not specified explicitly. In subsection VII-E, the performance under various unbalanced policies is evaluated.

As explained in Section VI, the DRL agent is trained with each episode equal to the slicing policy period T, and the next episode is obtained by sliding the entire policy period by one radio frame. In the training process, $T = 10$, i.e., each episode corresponds to 100 ms of physical time in the network. However, considering two consecutive episodes, the overall time interval is only increased by 10 ms, i.e., one radio frame, because of the sliding operation of the policy period.

The hyper-parameters for DRL training are summarized as follows. The maximal number of episodes in the training process is set to 300. For the actor network design, the two-layer GCN that is used for state embedding has an input dimension of 53 (51 neurons for channel states per PRB in one frame, and 2 neurons for link rate and QoS satisfaction). The sizes of GCN layers are $\{53\times64\}$ and $\{64\times128\}$ respectively. The sizes of FCNN layers are $\{128 \times 64\}$ and $\{64 \times 16\}$. The final output dimension, i.e., action dimension d_a, is set as 16, which equals the number of SNR degradation levels. The reason is as follows. As stated in 3GPP specification TS 38.214 [54], there are 16 CQI indices in all CQI tables, corresponding to 16 discrete SNR levels. If the degraded SNR level of a link falls out of the range of CQI table, the lowest transmission rate (i.e., MCS) is selected to serve that link. Finally, the SoftMax activation is used for generating the probability distribution of actions. For the critic network design, the only change is to modify the output dimension of FCNN as one. The learning rates for actor and critic networks are set as 0.0002 and 0.0001 respectively. The discount factor for accumulated rewards is set as 0.98.

1) Performance Metrics: To evaluate performance of *LinkSlice*, several performance metrics are considered. The first one is network throughput, which is defined as the total supported date rates of all users in the network. When QoS satisfaction of users is ensured, a slice enforcement scheme is considered better if its achieved network throughput is higher. A related performance metric is saturation throughput, which

Authorized licensed use limited to: University of Edinburgh. Downloaded on October 19,2026 at 21:07:55 UTC from IEEE Xplore. Restrictions apply.



(a) Episode reward (b) Policy loss (c) Policy entropy

Fig. 6. Convergence performance of *LinkSlice* in the first 120 episodes with a different number of base stations.

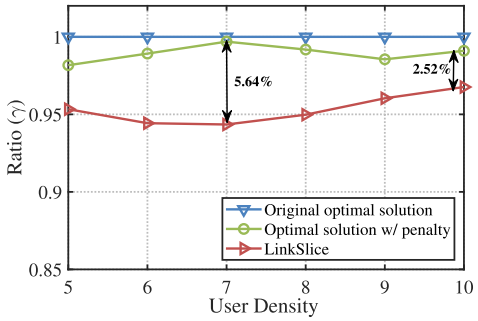


Fig. 7. Performance comparisons with the optimal solutions to P1 and P2.

TABLE I
SIMULATION PARAMETERS

Parameter	Value	Parameter	Value
Large-scale fading	UMi-Street canyon	Fast fading	Rician
Carrier frequency	4 GHz	Subcarrier spacing	30 kHz
Bandwidth	20 MHz	The number of PRB	51/slot
MCS index table	[54]	Base station	4,6,9
Frame size	10 ms	Slot size	0.5 ms

is obtained when the network is saturated, i.e., no more users can be admitted into the network. The second one is QoS satisfaction of users per slice, which is evaluated and compared between *LinkSlice* and other schemes. The third performance metric is slicing policy conformance. It is measured over one slicing policy period (i.e., 10 radio frames), and is then averaged over 1000 periods. The third metric is the number of reused PRBs across interfering links of different slices (i.e., interfering PRBs), which is used to quantify the performance of slice isolation. The smaller is this number, the better is slice isolation, and the best case is zero.

2) Other Schemes: To fully illustrate the performance of *LinkSlice*, it is compared with several other schemes. The first one is the optimal scheme. Its results are obtained directly from P1 by an optimization solver. The suboptimal scheme from P2 is also compared, and its results are also obtained from an optimization solver. The next scheme is random enforcement in which the required number of PRBs is selected randomly from a predetermined set. The users in the same cell are allocated resources sequentially. The state-of-the-art scheme to compare is the Most-Linked-First (MLF) scheme in [11]. It is designed based on the linked index that is defined as the amount of reused PRBs for a single slice across multiple cells. The slice-level enforcement is conducted so that the sum of linked indices is maximized.

B. Convergence of LinkSlice

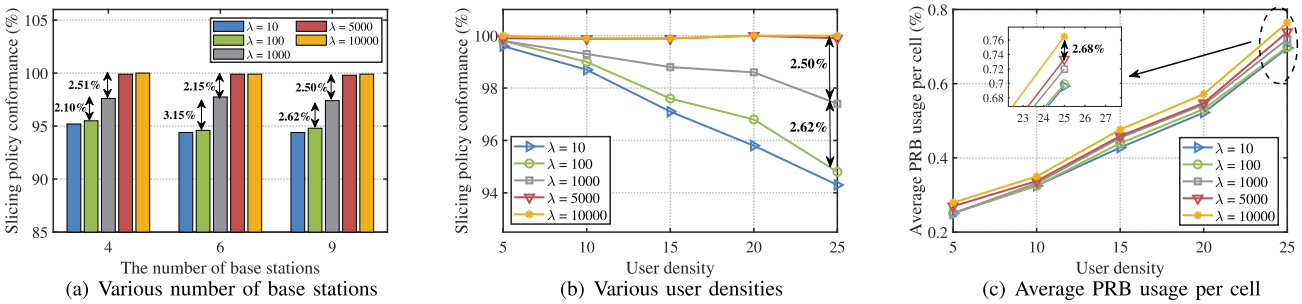
The convergence of two-stage slice enforcement is shown in Fig. 6. Note that the solid line represents the mean value that is computed out of 500 runs, and the shaded area represents the standard deviation. For a 4-BS network, the episode reward increases to reach a plateau that is close to one after 60 training episodes, as shown in Fig. 6(a). Here the reward is normalized into the range $[-1, 1]$, which is a common practice in DRL [55]. Furthermore, as shown in Fig. 6(b) and Fig. 6(c), both the policy loss and policy entropy decrease to reach a plateau that is close to zero after 60 training episodes.

The DRL agent reaches convergence after 60 episodes of learning, where one episode represents 10 radio frames. Thus, when the DRL algorithm runs online, it can converge within 0.69 s (i.e., 100 ms + 59 $\times$ 10 ms). In a cellular network, user mobility does not lead to a fast change in the interference pattern between communication links. In other words, the interference graph remains unchanged for tens of seconds. Thus, the training time of *LinkSlice* is much shorter than inference time during which *LinkSlice* is applied to conduct slice enforcement. In addition, when the number of nodes increases (i.e., the number of base stations is set as 6 and 9 while the user density is fixed), the convergence time of *LinkSlice* is still around 60 to 65 episodes. As a result, *LinkSlice* is highly adaptive to different interference graphs.

C. Performance Compared With Optimal Solutions

The performance of *LinkSlice* is compared with the optimal solution to problems P1 and P2. Both of the optimization problems, P1 and P2, are IQCP problems, which can be solved by Gurobi Optimizer, a business optimization solver designed for large-scale problems [56]. For a 4-BS network, there are millions of integer variables in both P1 and P2. Gurobi Optimizer can obtain the optimal solution to such a problem but it takes several hours or even days. The minimal resource usage, i.e., the minimal number of PRBs, is compared among *LinkSlice* and the optimal solutions to problems P1 and P2. The parameter γ is defined as the ratio of the performance gap (in the minimal resource usage) as compared to the optimal objective value of problem P1.

Authorized licensed use limited to: University of Edinburgh. Downloaded on October 19,2026 at 21:07:55 UTC from IEEE Xplore. Restrictions apply.



(a) Various number of base stations (b) Various user densities (c) Average PRB usage per cell

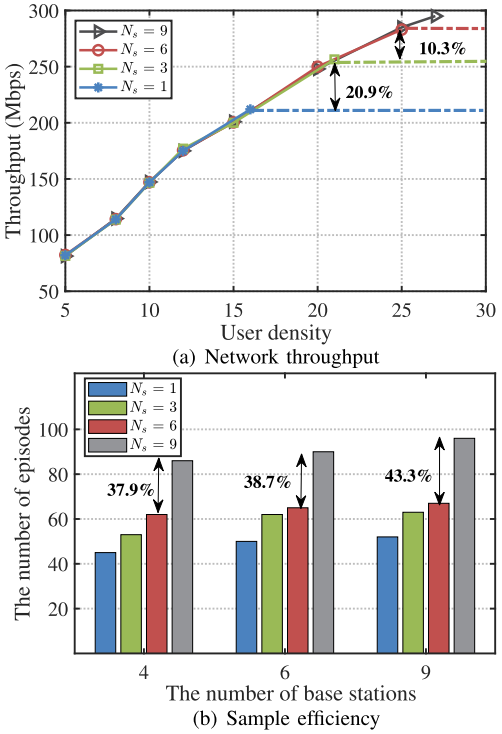
Fig. 8. Slicing policy conformance and network performance with various penalty coefficients.

For a 4-BS network, the comparison is conducted among the best solutions that can be achieved by tuning the penalty coefficient and other parameters. As shown in Fig. 7, the optimal solution with the penalty (P2) approaches the original optimal solution (P1) closely with a gap of less than 3%. Moreover, The largest gap for *LinkSlice* and the original optimal solution is around 5.6%, while the smallest gap is around 3.4%. This result indicates that *LinkSlice* can achieve a near-optimal result of slice enforcement.

D. Key Parameter Determination

1) Slicing Policy Conformance With Various Penalty Coefficients: Since *LinkSlice* is designed based on the transformed optimization problem P2 instead of the original optimization problem P1, the performance of slicing policy conformance is closely related to the penalty coefficient λ in P2. To determine an appropriate value of the penalty coefficient, slicing policy performance is evaluated with respect to various penalty coefficients under a different number of BSs.

As the penalty coefficient varies from 10 to 10000, there is a tradeoff between slicing policy conformance and system performance in terms of PRB resource usage, which is shown in Fig. 8(b) and Fig. 8(c). The experiments are conducted in a 9-BS scenario with uniform user distribution, and the unbalance index is set as 0.8. When the penalty coefficient λ increases from 10 to 10000, the slicing policy conformance increases by 5.60% with the user density equal to 25. Especially, when the penalty coefficient reaches 5000 or above, 99.8% conformance is guaranteed over 1000 periods of slicing policy. However, more PRB resources are needed to fulfill the same user demands as the penalty coefficient increases. In other words, the spectrum efficiency is compromised with a large penalty coefficient. When the penalty coefficient reaches 10000, the network needs 6.85% more PRB resources per cell on average than that with $\lambda = 10$, and 2.68% more resources than that with $\lambda = 5000$. Furthermore, the impact of the number of users and base stations on the penalty coefficient determination is evaluated. As shown in Fig. 8(a), as the number of base stations increases, the slicing policy conformance with the same penalty coefficient varies within 1.2%. Thus, the impact of the number of base stations is negligible. As the user density increases, there exists a gap in the slicing policy conformance caused by various penalty coefficients, as shown in Fig. 8(b). Especially, when the user density is 25, the slicing policy conformance increases 5.6% as the penalty coefficients increases from 10 to 10000. Thus, to guarantee slicing policy conformance, a larger penalty coefficient needs to be used as the user density increases.



(a) Network throughput (b) Sample efficiency

Fig. 9. The tradeoff between network throughput and sample efficiency with various sector sizes.

To balance the slicing policy conformance and system performance, the penalty coefficient is set to 5000 in the learning process[1]. In this case, the system ensures 99.8% slicing policy conformance, with a tolerable decrease in the spectrum efficiency.

2) Graph Pooling With Various Sector Sizes: As mentioned in Section VI, GCNs are incorporated into the DRL agent to perform state embedding. To reduce the action space, a graph pooling mechanism is used to pool all embeddings within one sector as one. The sample efficiency of DRL algorithms is defined as the reciprocal of the number of episodes needed for DRL agents to reach the same performance.

[1]In practice, at first, we choose the penalty coefficient λ so that the penalty term is 5 to 10 times larger than the average PRB usage per cell. Afterward, if the slicing policy conformance does not increase during the learning process, the penalty coefficient is tuned to a larger value, until the policy conformance is ensured.

Authorized licensed use limited to: University of Edinburgh. Downloaded on October 19,2026 at 21:07:55 UTC from IEEE Xplore. Restrictions apply.

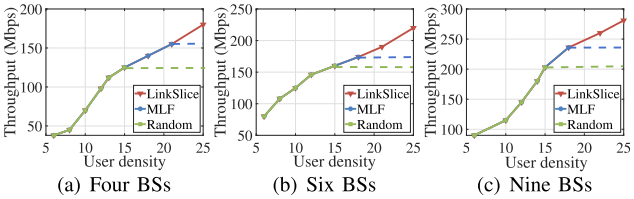


Fig. 10. Network throughput in uniform user distribution.

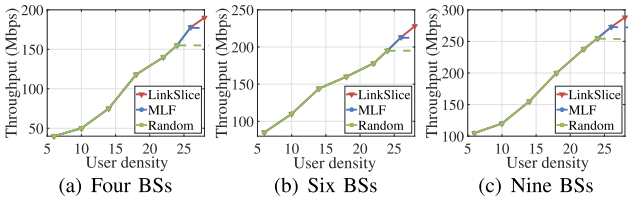


Fig. 11. Network throughput in cell-center-biased user distribution.

To evaluate the impact of the number of sectors per cell on the performance of *LinkSlice*, the tradeoff between the sample efficiency and network performance in terms of throughput is analyzed. As the number of sectors increases, the network throughput increases, while more episodes are needed for DRL convergence. Especially, as shown in Fig. 9(a), when the number of sectors N_s per cell is six, the network throughput is 30.3% higher than that with $N_s = 1$, and 4.2% lower than that with $N_s = 9$. As shown in Fig. 9(b), when the number of sectors is larger than 6, at least 37.9% more episodes are needed to train the DRL agent, but the performance improvement is limited. Therefore, in the following simulations, the number of sectors is set to 6.

E. Performance Compared With Existing Schemes

The performance of *LinkSlice* is evaluated and compared with other schemes in terms of network throughput, QoS satisfaction, slicing conformance, and slice isolation.

1) Network Throughput: Since user distribution impacts inter-slice interference of *LinkSlice*, three different user distribution scenarios are considered, i.e., uniform, cell-center-biased, and cell-edge-biased. In the uniform scenario, users are uniformly distributed in the cell. For the cell-center-biased scenario, users are more densely distributed in cell-center regions, while for the cell-edge-biased scenario, users are more densely distributed in cell-edge regions.

We first consider the network throughput in the uniform user distribution scenario, with an unbalance index of 0.8 in the slicing policies. The multi-cell network goes from the unsaturated to saturated state as the user density increases from 5 to 25. The simulation results are illustrated in Figs. 10, 11, and 12 for the uniform, center-biased, and edge-biased scenarios. In all scenarios, the network throughput increases with the user density and reaches the plateau (i.e., saturation) at a certain user density. For ease of presentation, all curves are extended via a flat dash line to cover the same range of user density. As shown in these results, *LinkSlice* supports the highest user density when saturation starts, so it can achieve the highest saturation throughput as compared to the existing schemes, but the performance gain depends on user distribution. More specifically, in the uniform scenario, *LinkSlice* achieves a

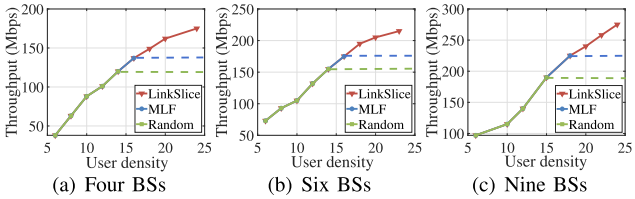


Fig. 12. Network throughput in cell-edge-biased user distribution.

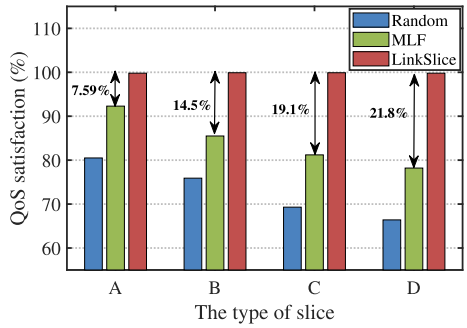


Fig. 13. QoS satisfaction in the saturation scenario of *LinkSlice*.

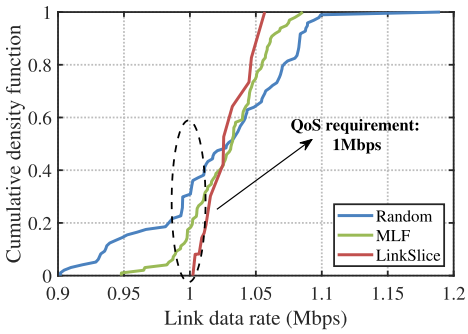


Fig. 14. The CDF of link data rate distribution for slice C (1Mbps) in the saturation scenario of *LinkSlice*.

network throughput of 18.5% higher than the MLF scheme and 40.5% higher than the random enforcement. In the cell-center-biased scenario, however, the performance gain of *LinkSlice* is not as significant as that in the uniform case. The reason is that *LinkSlice* is advantageous at scheduling interfering links, but there exist few interfering links when most users are distributed in the cell-center region.

2) QoS Satisfaction: The QoS satisfaction is compared among these three schemes under the highest user density that LinkSlice can support (i.e., saturation scenario). As shown in Fig. 13, the users' QoS satisfaction per slice is evaluated for four types of slices with the required data rates 100kbps, 500kbps, 1Mbps, and 5Mbps respectively (namely Slice A, B, C, and D). LinkSlice can well guarantee QoS requirements for the users associated with four different slices. Especially, the empirical CDF is used to demonstrate the average link data rate of users associated with slice C, as shown in Fig. 14.

3) Slicing Policy Conformance: Slicing policy conformance is evaluated when the network reaches saturation. The long-term slicing policy conformance is compared among these three schemes. As is illustrated in Fig. 15, slicing policy conformance of *LinkSlice* reaches almost 99.8% conformance, while random enforcement can only guarantee 91.8% conformance. Since the MLF scheme is tailored to ensure slicing

Authorized licensed use limited to: University of Edinburgh. Downloaded on October 19,2026 at 21:07:55 UTC from IEEE Xplore. Restrictions apply.

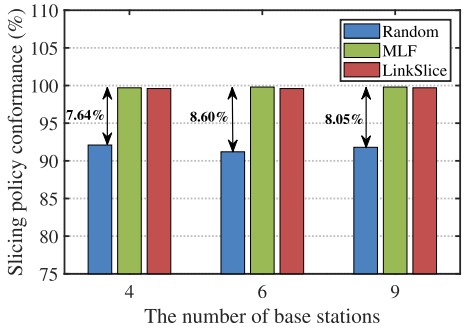


Fig. 15. Comparison of slicing policy conformance.

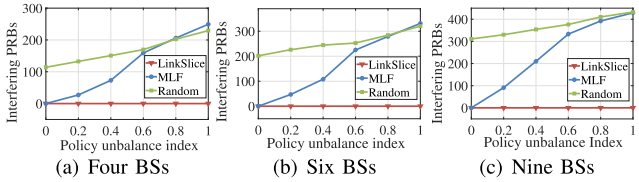


Fig. 16. The total number of interfering PRBs among different slices with various unbalance indices.

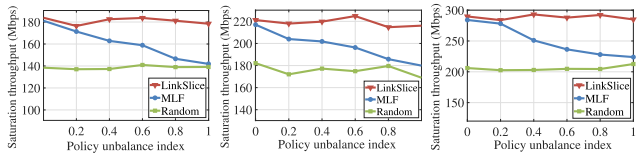


Fig. 17. Saturation throughput under slicing policies with various unbalance indices.

policy at the slice level instead of the link level, conformance to the slicing policies is not impacted by channel fading. Thus, it can always achieve high slicing policy conformance.

4) Slice Isolation: The relationship between slice isolation and the unbalance of slicing policies is revealed in Fig. 16. As slicing policies become more unbalanced, *LinkSlice* can always maintain nearly perfect slice isolation, i.e., the number of interfering PRBs among different slices is zero. However, for the random and the MLF schemes, it is difficult for them to ensure slice isolation under unbalanced slicing policies. The significant performance gain achieved by *LinkSlice* illustrates the advantage of link-level slice enforcement. As shown in Fig. 16, the MLF scheme achieves better isolation than the random scheme at a low unbalance index. However, when the unbalance index is higher than 0.8, the MLF scheme becomes worse and even the worst in slice isolation among all schemes. This result shows that a slice-level enforcement scheme like MLF is ineffective in ensuring slice isolation for unbalanced slicing policies.

5) Throughput Under Unbalanced Slicing Policies: Now the performance of *LinkSlice* under different slicing policies, either balanced or unbalanced, is evaluated. As is shown in Fig. 17, in all three multi-cell network scenarios, the MLF scheme cannot properly handle unbalanced slicing policies, while *LinkSlice* and random algorithm remain stable (within a small fluctuation range). As the unbalance index increases from 0 to 1, saturation throughput drops by 18.5% in the MLF scheme.

VIII. Conclusion

A link-level slice enforcement scheme called *LinkSlice* was developed for network slicing in a multi-cell radio access network. It satisfies the key requirements of network slicing, i.e., slice isolation, QoS guarantees, long-term conformance to slicing policies. As compared to the state-of-the-art, *LinkSlice* is distinct with the capability of achieving high throughput and perfect slice isolation under unbalanced slicing policies. As link-level slice enforcement needs to handle channel fading and interference between communication links, DRL and GCN were leveraged to design *LinkSlice*. GCN helped capture the interactions among interfering links. By embedding GCN into its agent and then working together with an optimization block, DRL accomplished resource allocation for fine-grained slice enforcement through a training process. Since *LinkSlice* can be trained within a time interval that is much shorter than the changing period of interference graph or user demands, it can be applied online effectively.

Several interesting topics need to be further investigated. First, how to obtain an interference graph was not studied in this paper. Although related work exists on this topic, it remains an interesting problem to obtain an interference graph accurately in a timely manner. Second, spatial reuse was not considered in this paper. In a beyond 5G network, beamforming and multi-user MIMO are common physical layer technologies. Thus, how to integrate *LinkSlice* with beamforming and multi-user MIMO is an interesting topic for future research. Third, joint optimization of user admission control, power allocation, and time/frequency allocation was not addressed in this paper. Power allocation has an impact on intra-slice interference management. Although one simple admission control scheme was incorporated in *LinkSlice*, it did not ensure fairness among users. Therefore, how to conduct fair user admission and power allocation together with *LinkSlice* is an interesting topic subject to future work.

Appendices

Proof of Theorem 1

Proof: It is shown in [57] that problems P1 and P2 must be equivalent when the coefficients of the penalty function go to infinity. The optimal objective values of P1 and P2 are denoted by $f(\mathbf{x}^*)$ and $g(\mathbf{y}^*)$ respectively, where $\mathbf{x}^*$ and $\mathbf{y}^*$ are the optimal solutions. Let $\mathcal{F}$ and $\mathcal{H}$ denote the feasible region of P1 and P2. Obviously, $\mathbf{x}^* \in \mathcal{F}$ leads to $\mathbf{x}^* \in \mathcal{H}$. When $\mathbf{y}^* \in \mathcal{H}$ and $\mathbf{y}^* \in \mathcal{F}$ are simultaneously satisfied, problems P1 and P2 have the same optimal solution $\mathbf{x}^* = \mathbf{y}^*$.

There exists an upper bound for the optimal value $f(\mathbf{x}^*)$, which is the total number of PRB resources in multiple cells BN (B is the number of cells, and N is the number of PRBs in one radio frame).

When $\mathbf{y}^*$ falls out of the feasible region $\mathcal{F}$, the penalty term $\sum_{b\in\mathcal{B}}\sum_{m\in\mathcal{M}} L_{b,m}$ must be at least one. Here, if λ takes on values larger than BN, it is guaranteed that $g(\mathbf{y}^*)$ must be larger than $f(\mathbf{x}^*)$. Since the reward is set to be the negative

Authorized licensed use limited to: University of Edinburgh. Downloaded on October 19,2026 at 21:07:55 UTC from IEEE Xplore. Restrictions apply.

of objective values, the DRL agent always receives smaller rewards if the sampled actions fall out of the feasible region $\mathcal{F}$ of problem P1. ■

Proof of Theorem 2

Proof: If transmission rates $r_{u,i}^{(b,m)}$ are assumed to be uniformly distributed over all the PRBs and the constraint (14) is removed in P3, graph coloring that optimizes the chromatic number can be reduced to problem P3. Since it is NP-hard to compute the chromatic number, problem P3 is proven to be NP-hard. ■

Proof of Theorem 3

Proof: Let B denote the number of BSs, U denote the number of users, and N denote the number of PRBs. For the following analysis, we assume that each BS has $\frac{U}{B}$ users.

The complexity of Welsh-Powell algorithm can be divided into two parts: calculate the degree of nodes $O(B \log B)$ and sort nodes by degrees $O(N^2)$. The masking mechanism is checking the neighbors of each cell-edge user, which is described by the complexity $O(\frac{U^2 N}{B})$. For the greedy allocation part, the priority order of users can be obtained in $O(\frac{U}{B} \log \frac{U}{B})$. Then PRBs are sorted by rates and allocated sequentially, which has the complexity $O(U \log \frac{U}{B} + UN \log N + BN)$. Therefore, the second stage algorithm has the order of complexity of $O(B \log B + N^2 + U \log \frac{U}{B} + UN \log N + BN)$. ■

REFERENCES

[1] I. Afolabi, T. Taleb, K. Samdanis, A. Ksentini, and H. Flinck, "Network slicing and softwarization: A survey on principles, enabling technologies, and solutions," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 3, pp. 2429–2453, 3rd Quart., 2018.

[2] X. Foukas, G. Patounas, A. Elmokashfi, and M. K. Marina, "Network slicing in 5G: Survey and challenges," *IEEE Commun. Mag.*, vol. 55, no. 5, pp. 94–100, May 2017.

[3] H. Zhang, N. Liu, X. Chu, K. Long, A. Aghvami, and V. C. M. Leung, "Network slicing based 5G and future mobile networks: Mobility, resource management, and challenges," *IEEE Commun. Mag.*, vol. 55, no. 8, pp. 138–145, Aug. 2017.

[4] 3rd Generation Partnership Project (3GPP), *Management and Orchestration of Networks and Network Slicing; 5G Core Network (5GC) Network Resource Model (NRM); Stage 1*, 3GPP, document TS 28.542, 2018.

[5] 3rd Generation Partnership Project (3GPP), *Study on New Radio Access Technology: Radio Access Architecture and Interfaces*, 3GPP, document TR 38.801, 2017.

[6] 3rd Generation Partnership Project (3GPP), *Management and Orchestration; 5G Network Resource Model (NRM); Stage 2 and Stage 3*, 3GPP, document TS 28.541, Dec. 2021.

[7] S. D'Oro, F. Restuccia, and T. Melodia, "Toward operator-to-waveform 5G radio access network slicing," *IEEE Commun. Mag.*, vol. 58, no. 4, pp. 18–23, Apr. 2020.

[8] R. Ferrus, O. Sallent, J. P. Romero, and R. Agusti, "On 5G radio access network slicing: Radio interface protocol features and configuration," *IEEE Commun. Mag.*, vol. 56, no. 5, pp. 184–192, May 2018.

[9] Y. Sun, S. Qin, G. Feng, L. Zhang, and M. A. Imran, "Service provisioning framework for RAN slicing: User admissibility, slice association and bandwidth allocation," *IEEE Trans. Mobile Comput.*, vol. 20, no. 12, pp. 3409–3422, Dec. 2021.

[10] J. Tang, B. Shim, and T. Q. S. Quek, "Service multiplexing and revenue maximization in sliced C-RAN incorporated with URLLC and multicast eMBB," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 4, pp. 881–895, Apr. 2019.

[11] S. D'Oro, F. Restuccia, A. Talamonti, and T. Melodia, "The slice is served: Enforcing radio access network slicing in virtualized 5G systems," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Apr. 2019, pp. 442–450.

[12] Y. Hua, R. Li, Z. Zhao, X. Chen, and H. Zhang, "GAN-powered deep distributional reinforcement learning for resource management in network slicing," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 2, pp. 334–349, Feb. 2020.

[13] S. D'Oro, F. Restuccia, T. Melodia, and S. Palazzo, "Low-complexity distributed radio access network slicing: Algorithms and experimental results," *IEEE/ACM Trans. Netw.*, vol. 26, no. 6, pp. 2815–2828, Dec. 2018.

[14] C. Sexton, N. Marchetti, and L. A. DaSilva, "On provisioning slices and overbooking resources in service tailored networks of the future," *IEEE/ACM Trans. Netw.*, vol. 28, no. 5, pp. 2106–2119, Oct. 2020.

[15] H. Zhu, V. Gupta, S. S. Ahuja, Y. Tian, Y. Zhang, and X. Jin, "Network planning with deep reinforcement learning," in *Proc. ACM SIGCOMM Conf.*, Aug. 2021, pp. 258–271.

[16] S. Mandelli, M. Andrews, S. Borst, and S. Klein, "Satisfying network slicing constraints via 5G MAC scheduling," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Apr. 2019, pp. 2332–2340.

[17] T. Guo and A. Suárez, "Enabling 5G RAN slicing with EDF slice scheduling," *IEEE Trans. Veh. Technol.*, vol. 68, no. 3, pp. 2865–2877, Mar. 2019.

[18] A. Ksentini and N. Nikaein, "Toward enforcing network slicing on RAN: Flexibility and resources abstraction," *IEEE Commun. Mag.*, vol. 55, no. 6, pp. 102–108, Jun. 2017.

[19] L. Zanzi, V. Sciancalepore, A. Garcia-Saavedra, H. D. Schotten, and X. Costa-Perez, "LACO: A latency-driven network slicing orchestration in beyond-5G networks," *IEEE Trans. Wireless Commun.*, vol. 20, no. 1, pp. 667–682, Jan. 2021.

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

[21] D. Shi, L. Li, T. Ohtsuki, M. Pan, Z. Han, and V. Poor, "Make smart decisions faster: Deciding D2D resource allocation via Stackelberg game guided multi-agent deep reinforcement learning," *IEEE Trans. Mobile Comput.*, early access, Jun. 1, 2021, doi: 10.1109/TMC.2021.3085206.

[22] L. Liang, H. Ye, G. Yu, and G. Y. Li, "Deep-learning-based wireless resource allocation with application to vehicular networks," *Proc. IEEE*, vol. 108, no. 2, pp. 341–356, Feb. 2020.

[23] 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.

[24] R. Li, C. Wang, Z. Zhao, R. Guo, and H. Zhang, "The LSTM-based advantage actor-critic learning for resource management in network slicing with user mobility," *IEEE Commun. Lett.*, vol. 24, no. 9, pp. 2005–2009, Sep. 2020.

[25] J. Mei, X. Wang, K. Zheng, G. Boudreau, A. B. Sediq, and H. Abou-Zeid, "Intelligent radio access network slicing for service provisioning in 6G: A hierarchical deep reinforcement learning approach," *IEEE Trans. Commun.*, vol. 69, no. 9, pp. 6063–6078, Sep. 2021.

[26] W. Wu *et al.*, "Dynamic RAN slicing for service-oriented vehicular networks via constrained learning," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 7, pp. 2076–2089, Jul. 2021.

[27] H. Xiang, S. Yan, and M. Peng, "A realization of fog-RAN slicing via deep reinforcement learning," *IEEE Trans. Wireless Commun.*, vol. 19, no. 4, pp. 2515–2527, Apr. 2020.

[28] T. Dong *et al.*, "Intelligent joint network slicing and routing via GCN-powered multi-task deep reinforcement learning," *IEEE Trans. Cognit. Commun. Netw.*, vol. 8, no. 2, pp. 1269–1286, Jun. 2022.

[29] R. A. Addad, D. L. C. Dutra, T. Taleb, and H. Flinck, "Toward using reinforcement learning for trigger selection in network slice mobility," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 7, pp. 2241–2253, Jul. 2021.

[30] X. Chen *et al.*, "Multi-tenant cross-slice resource orchestration: A deep reinforcement learning approach," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 10, pp. 2377–2392, Oct. 2019.

[31] 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.

[32] Y. Azimi, S. Yousefi, H. Kalbkhani, and T. Kunz, "Energy-efficient deep reinforcement learning assisted resource allocation for 5G-RAN slicing," *IEEE Trans. Veh. Technol.*, vol. 71, no. 1, pp. 856–871, Jan. 2022.

Authorized licensed use limited to: University of Edinburgh. Downloaded on October 19,2026 at 21:07:55 UTC from IEEE Xplore. Restrictions apply.

[33] R. Y. Chang, Z. Tao, J. Zhang, and C. C. J. Kuo, "Multicell OFDMA downlink resource allocation using a graphic framework," *IEEE Trans. Veh. Technol.*, vol. 58, no. 7, pp. 3494–3507, Sep. 2009.

[34] Y. Yu, E. Dutkiewicz, X. Huang, and M. Mueck, "Downlink resource allocation for next generation wireless networks with inter-cell interference," *IEEE Trans. Wireless Commun.*, vol. 12, no. 4, pp. 1783–1793, Apr. 2013.

[35] M. Rahman and H. Yanikomeroglu, "Enhancing cell-edge performance: A downlink dynamic interference avoidance scheme with inter-cell coordination," *IEEE Trans. Wireless Commun.*, vol. 9, no. 4, pp. 1414–1425, Apr. 2010.

[36] M. Eslami Rasekh, D. Guo, and U. Madhow, "Joint routing and resource allocation for millimeter wave picocellular backhaul," *IEEE Trans. Wireless Commun.*, vol. 19, no. 2, pp. 783–794, Feb. 2020.

[37] D. López-Pérez, X. Chu, A. V. Vasilakos, and H. Claussen, "On distributed and coordinated resource allocation for interference mitigation in self-organizing LTE networks," *IEEE/ACM Trans. Netw.*, vol. 21, no. 4, pp. 1145–1158, Aug. 2013.

[38] W. Li, J. Zhang, and Y. Zhao, "Conflict graph embedding for wireless network optimization," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, May 2017, pp. 1–9.

[39] Y. Zhao, W. Li, J. Wu, and S. Lu, "Quantized conflict graphs for wireless network optimization," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Apr. 2015, pp. 2218–2226.

[40] M. Eisen and A. Ribeiro, "Optimal wireless resource allocation with random edge graph neural networks," *IEEE Trans. Signal Process.*, vol. 68, pp. 2977–2991, 2020.

[41] Y. Shen, Y. Shi, J. Zhang, and K. B. Letaief, "A graph neural network approach for scalable wireless power control," in *Proc. IEEE Globecom Workshops (GC Wkshps)*, Dec. 2019, pp. 1–6.

[42] M. Lee, G. Yu, and G. Y. Li, "Graph embedding-based wireless link scheduling with few training samples," *IEEE Trans. Wireless Commun.*, vol. 20, no. 4, pp. 2282–2294, Apr. 2021.

[43] X. Zhou, Z. Zhang, G. Wang, X. Yu, B. Y. Zhao, and H. Zheng, "Practical conflict graphs in the wild," *IEEE/ACM Trans. Netw.*, vol. 23, no. 3, pp. 824–835, Jun. 2015.

[44] S. Paternain, L. F. O. Chamon, M. Calvo-Fullana, and A. Ribeiro, "Constrained reinforcement learning has zero duality gap," in *Proc. Adv. Neural Inf. Process. Syst., Annu. Conf. Neural Inf. Process. Syst. (NeurIPS)*, 2019, pp. 7553–7563.

[45] I. Bello, H. Pham, Q. V. Le, M. Norouzi, and S. Bengio, "Neural combinatorial optimization with reinforcement learning," in *Proc. 5th Int. Conf. Learn. Represent. (ICLR) Workshop*, 2017, pp. 1–5.

[46] H. Gao, Z. Wang, and S. Ji, "Large-scale learnable graph convolutional networks," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2018, pp. 1416–1424.

[47] W. Liu, J. He, and S. Chang, "Large graph construction for scalable semi-supervised learning," in *Proc. 27th Int. Conf. Mach. Learn. (ICML)*, 2010, pp. 679–686.

[48] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," 2016, *arXiv:1609.02907*.

[49] J. Jiang, C. Dun, T. Huang, and Z. Lu, "Graph convolutional reinforcement learning," in *Proc. 8th Int. Conf. Learn. Represent. (ICLR)*, 2020, pp. 1–12.

[50] J. Schulman, P. Moritz, S. Levine, M. Jordan, and P. Abbeel, "High-dimensional continuous control using generalized advantage estimation," 2015, *arXiv:1506.02438*.

[51] L. Zhou *et al.*, "A dynamic graph-based scheduling and interference coordination approach in heterogeneous cellular networks," *IEEE Trans. Veh. Technol.*, vol. 65, no. 5, pp. 3735–3748, May 2016.

[52] D. J. A. Welsh and M. B. Powell, "An upper bound for the chromatic number of a graph and its application to timetabling problems," *Comput. J.*, vol. 10, no. 1, pp. 85–86, 1967.

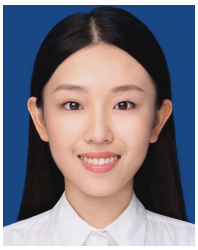
[53] 3rd Generation Partnership Project (3GPP), *Study on New Radio Access Technology Physical Layer Aspects*, 3GPP, document TR 38.802, 2017.

[54] 3rd Generation Partnership Project (3GPP), *NR; Physical Layer Procedures for Data*, 3GPP, document TS 38.214, 2021.

[55] P. Henderson, R. Islam, P. Bachman, J. Pineau, D. Precup, and D. Meger, "Deep reinforcement learning that matters," in *Proc. 32nd AAAI Conf. Artif. Intell.* AAAI Press, 2018, pp. 3207–3214.

[56] Gurobi Optimization, LLC. (2021). *Gurobi Optimizer Reference Manual*. [Online]. Available: https://www.gurobi.com

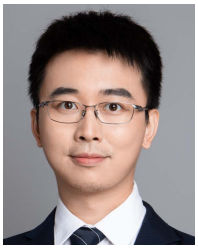
[57] Ö. Yeniay, "Penalty function methods for constrained optimization with genetic algorithms," *Math. Comput. Appl.*, vol. 10, no. 1, pp. 45–56, 2005.



Tianxin Wang received the B.S. degree in information science and engineering from Southeast University, Nanjing, China, in 2020. She is currently pursuing the Ph.D. degree with the Wireless Networking and Artificial Intelligence Laboratory, Shanghai Jiao Tong University, Shanghai, China. She was an Exchange Student at The Hong Kong Polytechnic University in 2018. Her research interests include wireless networks and reinforcement learning.



Suhong Chen received the B.S. degree in electronic engineering from the Tsinghua University, Beijing, China, in 2020. He is currently pursuing the master's degree with the Wireless Networking and Artificial Intelligence Laboratory, Shanghai Jiao Tong University, Shanghai, China. His current research interests include edge computing and distributed systems.



Yifei Zhu received the B.Sc. degree from Xi'an Jiaotong University, Xi'an, China, in 2012, the M.Phil. degree from The Hong Kong University of Science and Technology, Hong Kong, China, in 2015, and the Ph.D. degree in computer science from Simon Fraser University, BC, Canada, in 2020. He is currently an Assistant Professor at Shanghai Jiao Tong University. His research interests include edge computing, multimedia networking, distributed machine learning systems, and network economics.



Aimin Tang (Member, IEEE) received the B.S. and Ph.D. degrees in information and communication engineering from Shanghai Jiao Tong University, Shanghai, China, in 2013 and 2018, respectively. He was a Visiting Ph.D. Student at the University of Washington in 2016. He is currently a Research Assistant Professor with the University of Michigan-Shanghai Jiao Tong University (UM-SJTU) Joint Institute, Shanghai Jiao Tong University. His current research interests include B5G networks, joint communication and sensing systems, full duplex communications, and coded caching. He received the Best Paper Award of IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC) in 2021.



Xudong Wang (Fellow, IEEE) received the Ph.D. degree in electrical and computer engineering from the Georgia Institute of Technology in 2003. He is currently the John Wu and Jane Sun Professor of engineering with the UM-SJTU Joint Institute, Shanghai Jiao Tong University. He is also an affiliate Professor with the Department of Electrical and Computer Engineering, University of Washington. He was a Senior Research Engineer, a Senior Network Architect, and a Research and Development Manager with several companies. His research interests include wireless communication networks, joint communications and sensing, edge computing, and federated learning. He was a Voting Member of IEEE 802.11 and 802.15 Standard Committees. He was the General Chair of 2017 IEEE 5G Summit in Shanghai and the TPC Co-Chair of the 32nd International Conference on Information Networking. He was also the demo Co-Chair of the ACM International Symposium on Mobile Ad Hoc Networking and Computing (ACM MOBIHOC 2006), the Technical Program Co-Chair of Wireless Internet Conference (WICON) in 2007, and the General Co-Chair of WICON in 2008. He was an Editor of the IEEE TRANSACTIONS ON MOBILE COMPUTING, IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, *Ad Hoc Networks* (Elsevier), and *China Communications*. He was also a guest editor for several international journals.

Authorized licensed use limited to: University of Edinburgh. Downloaded on October 19,2026 at 21:07:55 UTC from IEEE Xplore. Restrictions apply.