IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 24 November 2022, accepted 4 December 2022, date of publication 12 December 2022,
date of current version 30 December 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3228804

RESEARCH ARTICLE

An SDN Controller-Based Network Slicing Scheme Using Constrained Reinforcement Learning

MDUDUZI C. HLOPHE, (Member, IEEE), AND BODHASWAR T. MAHARAJ, (Senior Member, IEEE)
Department of Electrical, Electronic and Computer Engineering, University of Pretoria, Pretoria 0002, South Africa
Corresponding author: Mduduzi C. Hlophe (mduduzi.hlophe@tuks.co.za)

This work was supported by the Sentech Chair in Broadband Wireless Multimedia Communications (BWMC) at the University of Pretoria.

ABSTRACT In order to meet the strong diversification of services that demand network flexibility that will be able to serve the dire need for transmission resources, network slicing was embraced as a plausible solution. Reinforcement learning (RL) has been applied in resource allocation (RA) problems, but has not yet marked the translation from traditional optimization approaches primarily due to its inability to satisfy state constraints. The aim of this article is to address this challenge. This article proposes a logical architecture for network slicing based on software-defined networking (SDN), where an SDN controller controls the network slicing process in a centralized fashion, and manages the resource allocation (RA) process with the help of the slice manager. The considered problem jointly addresses power and channel allocation using a hybrid access mode for ultra-reliable low-latency communication (URLLC) and enhanced mobile broadband (eMBB) slices. Proper assumptions on the arrival rates, packet length distributions, as well as power and delay constraints were used to design the behavior of the reward function to realize a constrained RL approach. Here, the Bellman optimality equation was reformulated into a primal-dual optimization problem through the use of Nesterov's smoothing technique and the Legendre-Fenchel transformation. The proposed algorithm shows favorable performance over the traditional RL strategy in attributes favoring eMBB services, i.e., the average bit rate, and significantly outperforms both baselines in attributes favoring URLLC services, i.e., average latency. Systematically, on the power-delay performance evaluation, it shows that it can adapt very well in rapidly time-varying non-Markovian environments and still successfully satisfy the delay constraints of the applications hosted on a slice.

INDEX TERMS 5G, Bellman optimality, constrained reinforcement learning, eMBB, mMTC, network slicing, non-Markovian, power-delay, resource allocation, satisfaction degree, URLLC.

I. INTRODUCTION AND BACKGROUND

The design of traditional mobile and wireless networks has always focused on supporting specific services such as voice, messaging, and internet access. However, with the unprecedented and accelerated development of wireless networks towards the fifth generation (5G), mobile network operators (MNOs) face the ever-escalating challenges of meeting the demands of diverse vertical industry applications [1]. For the 5G new radio (NR) to be able to simultaneously accommodate and meet the demands of these industry applications and services, the network must be able to focus on the requirements of each 5G use case. 5G use cases boast of services with diverse requirements such as ultra-low latency as well as high resilience for real-time control of critical systems. Typical examples of these services include, but not limited to: (i) the enhanced mobile broadband (eMBB), (ii) ultra-reliable and low-latency communication (URLLC), and (iii) massive machine-type communications (mMTC). These services illustrate the wide diversity of their associated requirements, enabling a paradigm shift that can only be handled by slicing the physical network into logical sub-networks - a concept referred to as network slicing. The network slicing concept is not new as it dates back to distributed service architectures such as distributed cloud computing systems [2].

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

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

A network slice can be defined as a virtual network that is implemented on a physical network such that it creates the illusion that slice tenants are operating on their own dedicated physical networks. In network slicing, the physical network is split into numerous logical or virtual network, each logical network tailored to serve particular services or applications. As a result, the network slicing paradigm is predetermined to manage the diversity of emerging applications by enhancing the performance and flexibility requirements of the physical networks [3]. This was a way of keeping up with the tremendous amount of data that is generated by the enormous number of user equipments from the applications they run on a daily basis. However, this has caused remarkable challenges for network designers in terms of making sure that new designs have a considerable positive influence on network slicing performance. Therefore, network slicing determines the self-contained and logical networks consisting of a combination of dedicated instances of resources [4].

With the rise of the internet of things (IoT), more especially the fastest growing IoT paradigm known as the industrial IoT (IIoT), network slicing cuts the physical network into several end-to-end virtual networks. Each virtual network obtains logically independent network resources for supporting the richer services [5]. When the Third Generation Partnership Project (3GPP) released the initial 5G network recommendations, i.e., 5G NR evolution in Release 15 [6], this gave assistance to analytical communications, huge vehicular-to-everything, as well as mMTC. As a result, a corresponding data traffic was anticipated to develop many-folds over the years. The first complete set of the physical layer design allows achievements of improved metrics for latency and reliability that can support new use cases of URLLC. With this rapid development of mobile devices, several appealing applications were developed for the eMBB use case, further enhancing the necessity of increasing quality of service (QoS) [7]. Before a new slice is orchestrated, the mobile network operator (MNO) has to first determine the required slice functionality as well as the required resources. For example, the anticipated applications to be handled by the eMBB slice are those requiring high throughput, which entail faster download speeds for increased seamless experience. The eMBB use case is one of the three defining characteristics of 5G networks with throughput speeds expected to eventually reach 20 Gbps once the millimeter wave (mmWave) frequencies become available [8].

The second use case is anticipated to confront the unmatched demand for reduced latency in communications, i.e., the URLLC slice to handle delay-constrained applications such as automotive communications and remote healthcare [9]. Then, in line with the IIoT and the dictates of the Industry 4.0, the mMTC slice corresponds to the gigantic amounts of data produced by millions of sensors. The MNO may be required to select a slice template that fits the requirements of each slice and parametrize it according to its demands. The resource allocation (RA) and scheduling algorithms applied for admitting traffic in each slice at each transmission time interval (TTI) is evaluated based on the average throughput, fairness, and spectral efficiency [10]. This means that the variables such as bandwidth, packet losses, signal strength, latency, user density, network protocol and topology, that affect throughput, are the ones that need to be investigated. However, given the critical nature of the URLLC applications, its traffic flows have to be given higher priority over the others, which might cause negative performance effects on the other types of applications [11]. Since the mobile and wireless devices are not aware of this interrupt, packet losses for the devices running eMBB and mMTC applications may increase sharply. This is an inevitable coexistence problem that leads to performance degradation, and can only be mitigated using proper mechanisms. In other words, more flexible resource allocation (RA) and scheduling approaches are required to be able to support the eMBB services without compromising the other services. Another possible way to handle this coexistence problem is: when a request is generated for a URLLC service, prompt access to the wireless medium must be granted for immediate packet transmission. Since the sum of demands for transmission resources are expected to be high and very dynamic for all the slice types, especially during peak hours, dynamic RA is required in network slicing. When using dynamic RA, the utility function of each network slice can be maximized while individual users control their transmission powers in order to reduce interference. The objective of this article is to test the effectiveness of the approach in solving the coexistence problem by maximizing the throughput of the eMBB slice, while giving priority to URLLC users.

II. PREVIOUS RELATED WORKS

The discussion of the state-of-the-art algorithms related to this research work begins with a focus on the use of traditional optimization approaches in network slicing. By traditional optimization approaches, reference is made to mathematical programming techniques such as game/queuing theoretic approaches, etc. Then focus will be shifted to learning based strategies, where reinforcement learning (RL) strategies related to this work are discussed. In each case, only a few outstanding research contributions that are in resonance with the objectives of this work will be reviewed.

A. MATHEMATICAL PROGRAMMING APPROACHES

With the existence of different slices and services in the same physical network creating a challenging RA task, the impact of algorithms for maximizing data rate, spectral efficiency, as well as fairness, is reduced. In an attempt to address this challenge, the authors in [10] formulated a URLLC and eMBB RA problem as an optimization problem with the aim of maximizing the average throughput of eMBB applications, while simultaneously satisfying the latency requirements of URLLC applications. Dynamic programming was then applied to achieve an optimal RA for URLLC traffic on a TTI level that minimizes the negative impact on the average throughput of eMBB applications. The authors did this in

addition to maintaining a tolerable level of fairness among eMBB users. Their approach was implemented on heuristic scheduling algorithms where the URLLC traffic punctures the preserved/pre-allocated resources of eMBB users upon arrival. Numerical simulations were conducted in order to evaluate the effectiveness of this approach and the obtained results indicated how the approach was able to minimize the negative impact of URLLC traffic on the performance in terms of the achievable data rate, spectral efficiency, as well as the fairness.

The authors in [12] studied network slicing and slice coordination on the radio access network (RAN) where they formulated the problem as a bi-convex one. However, due to the existence of some complicated couplings between RAN RA for each slice and the coordination between slices sharing the same network resources, slicing of the RAN becomes challenging. In order to cope with this challenge, the authors then designed two algorithms that address the couplings of the bi-convex problem, whose objectives were to minimize two important components, i.e., (i) the aggregate load of gNB slice for load balancing, and (ii) the cost of the backhaul links for all the slices for delay minimization. The second objective captures two important inter-dependencies, i.e., (i) the dependency between the radio bandwidth at the gNB as well as caching slice allocation to minimize the aggregate load of each slice, and (ii) the dependency between the caching and backhaul slice allocation that minimizes the aggregate backhaul delay. Computer simulations were used in evaluating the performance of the proposed algorithms, and their efficiency proved their validity in solving network slicing problems at the RAN side. However, one shortfall was on the realization that the algorithms do not have any global convergence guarantees. Conversely, the results also showed that a global solution can be achieved in a simplified scenario consisting of only two tenants, which compares well with the convergence performance of exhaustive search. When the number of tenants was increased, the simulation results indicated that the two proposed algorithms converge to a similar performance to that of the Interior Point Optimizer (IpOPt) solver, which validates their efficacy.

With energy consumption being a very critical issue for MNOs that deeply impacts the cost of service provisioning, MNOs may have difficulties in coping with high energy costs. Since network slices require different types of resources, which include energy in order to fulfill the requirements of each application, the legacy energy efficiency models are considered a critical concern. In order to address this critical energy consumption issue within the limited radio and power resources, the authors in [13] designed a dynamic energy and cost-efficient RA strategy for IoT services. To achieve the objective energy efficient slicing, the authors proposed a RAN slicing and scheduling scheme that would ensure extreme QoS of differentiated IoT services. Here, a coexistence scenario of URLLC and eMBB services was considered in an software-defined networking (SDN)-enabled wireless RAN for allocating the shared resources. In this regard, the authors focused on guaranteeing the reliability and the latency of sporadic URLLC uplink (UL) traffic while simultaneously improving the QoS of continuous eMBB services such as video. The URLLC traffic was characterized by small and sporadically data packages, whereas eMBB traffic was characterized by large payloads. In order to guarantee sufficiently high and stable image and video quality and content, high peak data rates and high bandwidth were considered. In order to simultaneously support URLLC and eMBB services, the SDN controller was used to allocate corresponding resources to each network slice and to also control the performances of the devices on each slice. A dynamic optimization model for service quality and power consumption was then used to design a cost function in both the time domain and frequency bandwidth for heterogeneous services - constrained by latency. In order to ease the complexity of the model, a novel two-timescale algorithm was designed using Lyapunov optimization. From this, (i) a long-timescale bandwidth allocation, and (ii) short-timescale service control, were the two resulting sub-algorithms. The utility function for this approach was derived using hard latency guarantees, while its theoretical optimality was analyzed according to the relationship between the control parameters and the service performance. The performance of the proposed approach was evaluated through simulations, where the performance analysis explicitly characterized the relationship between the control parameters and the services performance, which included the power consumption and user the satisfaction. In comparison, the proposed algorithm proved to outperform baseline in terms of the total cost and hard latency.

The enhancement of reliability in mobile and wireless communication networks is critical in keeping a high level of energy efficiency in autonomous systems, more especially in emergency situations, such that unmanned aerial vehicles (UAVs) have increasingly become topical among researchers over the past few years. As a result, UAV relay networks are taunted as the most significant complements for terrestrial infrastructure in providing robust network coverage and capacity. To this effect, some researchers began focusing on either eMBB payload communication or URLLC control information, but not both. For instance, the authors in [14] investigated the multiplexing of eMBB payload and URLLC control information communication for multi-UAV relay networks. The proposed multi-UAV comprehensively considered path losses, small-scale channel fading, as well as different QoS requirements for both use cases. With the objective of improving the total data rate, while reducing power consumption, the authors formulated the multiplexing problem as a joint user association, bandwidth, and transmission power optimization. However, the problem seemed non-convex and NP-hard as a result of the coupling of continuous and integer variables. The solution to this problem was made even more challenging by the differences in capacity characteristics and requirements for both use cases. To mitigate these challenges, the authors proposed to equivalently decompose the original optimization problem into two

sub-problems, i.e., a URLLC problem and an eMBB problem. After decomposing it, closed-form expressions for optimal bandwidth and transmission power were derived for the URLLC problem, while an iterative solution framework for alternatively optimizing user association, the bandwidth and transmission power was developed for the eMBB problem.

B. REINFORCEMENT LEARNING-BASED APPROACHES

The reinforcement learning (RL) strategy is used to address the fixed RA mechanisms that may result in a low utilization of resources, hence violate users' QoS demands in specific slices due to network demand fluctuations. It does so by bringing together a resource management system with a dynamic resource adjustment technique known as the Q-learning algorithm. From the analysis in section **II-A** above, it is evident that the process of resource utilization is a complicated one for traditional optimization approaches since they cannot effectively perform resource orchestration. The reason for this is the lack of accurate models, as well as the existence of dynamic hidden structures within the problems. In order to address this resource utilization issue, the authors in [15] formulated the network slicing problem using constrained Markov decision processes (CMDPs) and solved it using constrained RL. The CMDP was used to indicate how the constrained RL has to be applied in a scenario consisting of hidden dynamics. The authors set up a gNB scenario consisting of three types of services, i.e., video, Voice over LTE (VoLTE), and URLLC, with each service having random users. The gNB had a fixed total bandwidth of 100 Mbps, and the task was to allocate bandwidth to all the types of users. Thus, at the start of each time slot, the gNB had to make a decision on the bandwidth allocation based on the number of users currently on each slice, while tracking cumulative and instantaneous constraints. In order to handle these constraints, an adaptive interior-point policy optimization and projection layers was used. The user throughput as well as the user dissatisfaction per slice with respect to the service received were the main attributes in performance evaluation. The performance evaluation through simulations indicated that the proposed constrained RL strategy was effective in the proposed RA objective and in comparison, it outperformed other baseline algorithms.

All the applications in network slicing have opened new business opportunities and business models are required for each slice in the form of a service level agreement (SLA). This forms a tenant-based network slicing scenario, where the MNO offers network slices to tenants to generate revenue. In this case, the QoS of individual users translates to a slice satisfaction degree if the service provision meets the SLA [16]. In this way, every tenant that rents a network slice from the MNO has to pay a fixed amount either a monthly or annual fee for the resources shared according to the contract signed between them. However, such a fixed RA mechanism usually lead to a low utilization of resources and even user QoS violations caused by fluctuations in network demand. In order to address this issues, the authors in [17] introduced a resource management strategy by proposing a dynamic resource adjustment algorithm using the RL strategy from the tenant's perspective. Here, multiple slices were built on the same physical network comprising of the RAN and core network, where virtual network functions (VNFs) and transmission resources were the two different types of resources that were spread across the entire physical network to be shared. Here, three stakeholders, i.e., the MNO/slice provider, the slice tenant, and the end-user, interact in order to realize the end-to-end communication service. The resource management problem for network slicing was modeled as an MDP, and a technique for dynamic resource adjustment that was aimed at maximizing tenants' profits while ensuring that QoS demands for end users were met was developed using Q-learning. The performance of the proposed scheme was evaluated using numerical simulations, where the results demonstrated that the proposed algorithm significantly increases the tenants' profits compared to the existing fixed RA methods while satisfying the QoS requirements of end users.

C. RESEARCH MOTIVATION

The new 5G services, with immersive and high-stake applications, are posing unprecedented challenges for MNOs in terms of both system design and algorithmic solutions. Proper allocation of resources is very crucial in the telco business and it may result in significant reductions in operating costs and increased revenues. Since the allocation of resources is a repetitive task that can be effectively automated using artificial intelligence (AI) strategies, the application of reinforcement learning (RL) results in varying results, which are seldom optimal. The efficient exploration of the state space in the current RL applications in RA still remains a challenge. It cannot carry out deep exploration due to the epsilon (ϵ)-greedy exploration strategy, hence suffers from poor convergence. This challenge consequently results in more cycles required to reach convergence, and becomes unfeasible when the network becomes large with a large state space. When using function approximation in RL, the Bellman optimality equation is applied as a rule of thumb, which has some guarantees of stability when dealing with single-objective problems.

However, when multi-objective optimization problems such as the ones encountered in network slicing are attempted, obtaining a solution for the Bellman optimality equation with stability guarantees becomes as challenge. As a result, performing slice evaluation and management on-the-fly becomes a huge challenge since this requires the collection and correlation of a mixture of variables on network conditions, the slice services, as well as the user requirements. The fundamental difficulty emanates from the fact that the Bellman operator may easily become an expansion, resulting in an oscillation and even divergent behavior for the Q-learning algorithm. This has been a pervasive problem in the RL community for a long time, while the application of RL strategies in other fields continued - completely ignoring

this fact. Therefore, the motivation behind this research work is to apply the RL strategy in network slice RA, using an optimization technique that will solve the Bellman error while maximizing the average throughput for eMBB users and satisfying latency requirements for URLLC users.

D. RESEARCH CONTRIBUTIONS

In order to address the problems associated with RL in order to enable it to handle the QoS requirements of slice users, a constrained RL approach is adopted. The contributions of this article are summarized as follows:

1) THE OPTIMIZATION PROBLEM

An optimization problem for assigning network slice requirements to two 5G use cases so that their total utility can be maximized if formulated. The proposed model considers a physical network that is sliced into two logical sub-networks, where slice 1 provides eMBB services to the internet, while slice 2 offers URLLC to the edge cloud and cache. The constraints associated with the QoS of each network slice request are monitored by a software-defined networking (SDN) controller whose task is to decouple the network control from the data plane, and centralize management of queues. In line with the 5G RAN concepts, slice orchestration and management are controlled by the SDN controller, which uses the northbound application programming interface (API) to obtain slice requests from slice tenants. Also, in this model, a slice manager receives channel quality information and the number of physical resource blocks (PRBs), sends it to the SDN controller, which then determines an assignment policy to the slice manager. The resource scheduling for the latency-critical traffic of the different users of the URLLC slice requires a proper queuing strategy that is incorporated into a properly constrained technique.

2) THE PROPOSED ALGORITHM

The problem was formulated using a constrained reinforcement learning (RL) strategy in order to tackle the challenges faced by the conventional RL strategy in resource allocation. Proper assumptions of Poisson distribution on the arrival rate, exponential distribution on packet lengths, as well as the constraints were conveniently specified in designing the behavior of the reward function. For instance, systems like network slicing where the system has to interact with different slice tenants, different slice users, as well as the service level agreement (SLA), must satisfy safety and reliability constraints [35]. The fundamental difficulty with the traditional RL strategy, with regard to the Bellman optimality equation, was addressed by reformulating the Bellman optimality equation into a primal-dual optimization problem through the use of Nesterov's smoothing technique [20] as well as the Legendre-Fenchel transformation [23]. Then, through the observation of the "$\log - \sum - \exp$" function, a novel constrained RL strategy, which enables the derivation of slice admission control decisions, was realized. The performance evaluation results show that the proposed method can effectively solve the network slicing problems with less complexity that the conventional RL strategy. The power-delay evaluation of the proposed algorithm show that it can also adapt well in rapidly time-varying non-Markovian environments and still successfully satisfy the delay constraints of the hosted applications.

E. NOTATIONS AND ARTICLE OUTLINE

For ease of readability and exposition, the notations used in this article and their descriptions are listed in TABLE 1 below.

TABLE 1. List of notations and their definitions.

Notation	Description
W	System bandwidth
$\mathcal{N}$; $\mathcal{J}$	The set of all gNBs, indexed by n; The set of all PRBs per gNB
$\mathcal{K}$; $\mathcal{K}_i$	The set of all users in the network; The set of users per slice, $i \in [1, 2]$
$\mathcal{S}$	The set of all environmental states
$\mathcal{A}$	The set of all possible actions or commands
$\psi(r)$	Slice policy as a function of the rate r
$\psi_{n,j,k}$	Slice scheduling binary decision variable
r_k	Achievable data rate per-user per slice
$r_{n,j,k}$	Data rate offered by PRB j of gNB n
$\lambda_{1,n}$; $\lambda_{2,n}$	Traffic arrival rates in both slices
$r^{(1)}_{n,j,k}$	Guaranteed data rate for eMBB users at gNB n
$r^{(2)}_{n,j,k}$	Guaranteed data rate for URLLC users at gNB n
$r^{(2)}_{req}$	Minimum required data rate for URLLC users
$p_{n,j,k}$	Transmission power of the k-th user, allocated the j-th PRB on the n-th gNB
$g_{n,j,k}$	Path loss-based channel gain between the k-th user and the n-th gNB
$\gamma_{n,j,k}$	The SINR between the k-th user and the n-th gNB
γ_0	The SINR constraint imposed by each gNB
$\gamma^{th}_{1,n}$	Percentage threshold for eMBB users
$\rho_{1,n}$	Resource percentage threshold based on the number of URLLC users
β; β_{new}	Initial slice admission condition; New slice admission condition
$\beta_{1,n}$	Admission bandwidth threshold for the eMBB slice
ϕ_n	Admission control probability for the n-th gNB
ϱ; χ	Fixed approximation parameters
$q_{2,n}$	Backlog of the URLLC queue
d_k; $d_{k,max}$	Average delay for the URLLC queue; maximum delay
$R_{(\gamma_k)}$	Average satisfaction of slice users
I_k; $\mathcal{I}$	Binary indicator for interference; aggregate interference
ϵ_0	Packet error rate constraint for URLLC services
Π	Set of all possible policies defined by π
$R_{(\gamma_k)}(\pi^*)$	Optimal solution due to optimal policy π^*
α_t; γ^t	The learning rate; Discount factor

The rest of our work is outlined as follows: The system model and problem formulation as well as the resource percentage optimization problem is presented in section **III**. Section **IV** discusses the problem formulation of the baseline and its solution using dynamic resource percentage threshold. Section **V** presents the formulation and solution of the proposed network slicing problem using solution resource allocation solution using constrained RL. Section **VI** discusses the two algorithms together with their respective computational complexities. The simulation results depicting the performance of the proposed algorithm are reported in Section **VII**; and Section **VIII** gives concluding remarks

about the performance that was observed between the two algorithms.

III. PROPOSED NETWORK AND SYSTEM MODEL

Consider a network slicing architecture where a SDN controller uses the north-bound application programming interface (API) to obtain slice requests from slice tenants [24]. The proposed SDN-based network slicing architecture is shown in FIGURE 1 below.

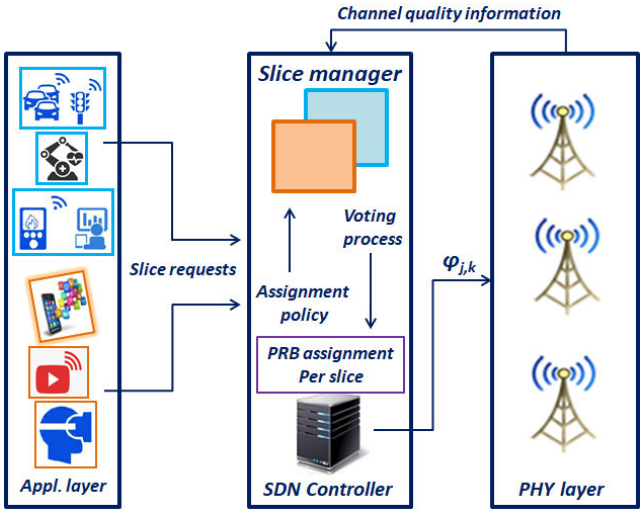

FIGURE 1. The proposed SDN controller-based network slicing scheme.

As shown in FIGURE 1 above, the physical network is sliced into two logical sub-networks, which offer connections between users and specific network components. For instance, slice 1 is offering URLLC services by reserving communication and buffer resources from users to the edge cloud/cache, while slice 2 offers eMBB services between users and the internet. In line with the cpncepts of the 5G RAN, the slice orchestration and management are controlled by an SDN controller, which creates an instance or slice manager for each network slice deployed and also dynamically assigns resources for each network slice by giving its manager the to allocate it to the tenants [25]. An SDN controller is used to decouple the network control from the data plane, hence centralizing the management of queues. Here, the SDN controller is responsible for collecting channel quality information from the gNBs for the RA process. Then, the SDN controller sends the assignment policy to the slice manager, which then controls the slices by abstracting the users from the SDN controller and also coordinates the scheduling decisions [29]. The slice manager uses a voting process to specify the rate policy, $\psi(r)$, to the SDN controller, which then decides on the maximum throughput per slice and then issues a scheduling decision, $\psi_{j,k}(r)$, assigning slice users to appropriate gNBs.

A. PRELIMINARIES

1) PHYSICAL LAYER PARAMETERS

A time-slotted system model is assumed in this work whereby the system is time-slotted with the duration of each time slot corresponding to a long-term evolution (LTE) TTI. A frequency non-selective channel is considered and the orthogonal frequency division multiple access (OFDMA) scheme is adopted [27]. In this case, channel gains are governed by the path loss model defined in [28]. The set of channel states is assumed to be discrete, finite, and constant over the duration of the time slot. In this way, channel conditions can be perfectly estimated and the sequence of their states is modeled using a Markov chain with distinct transition probabilities. It is further assumed that at each transmission instance each user connects to only one gNB, the scheduling decision, $\psi_{j,k}(r)$, translates to the gNB-user association factor, $\psi_{n,j,k}(r)$.

2) SYSTEM LEVEL PARAMETERS

A set, $\mathcal{N} = \{1, 2, \cdots, N\}$ of gNBs, where each gNB operates based on constant power transmission and uses proportional fairness to allocate PRBs from a set $\mathcal{J} = \{1, 2, \cdots, J\}$ [26]. The MNO is offering two pre-defined slice types, i.e., eMBB and URLLC, such that the overall set of network users, $\mathcal{K}$ is split into two subsets of slice users, i.e., $\mathcal{K}_1$ and $\mathcal{K}_2$ such that $\mathcal{K} = \mathcal{K}_1 + \mathcal{K}_2$. Here, the SDN controller controls the network slicing process in a centralized fashion. The arrivals on the eMBB and URLLC slice types are defined by the parameters $\lambda_{1,n}$ and $\lambda_{2,n}$, respectively. As such, the queuing model employed considers both request arrivals and request service (acceptance) as Poisson processes, such that every request in the queue follows the single-server birth-death process, hence the feature of the $M/M/1$ queuing system are directly applied.

IV. MATHEMATICAL PROBLEM FORMULATION

Assuming uniform power allocation technique over all PRBs is assumed, such that the achievable data rate per-user per-slice can be defined as follows:

$$r_k = \sum_{j=1}^{J} \psi_{n,j,k} \cdot r_{n,j,k}, \tag{1}$$

where $\psi_{n,j,k}$ represents a binary decision variable whether the k-th user is allocated the j-th PRB of gNB n or not, given as follows:

$$\psi_{n,j,k} = \left\{ \begin{array}{ll} 1, & \text{if PRB } j \text{ assigned to } k \\ 0, & \text{otherwise,} \end{array} \right\} \tag{2}$$

and

$$r_{n,j,k} = W \cdot \log_2 \left(1 - \frac{p_{n,j,k} \cdot g_{n,j,k}}{\sum_{m \neq k, m \in K} p_{j,m} \cdot g_{n,j,m} + \sigma^2} \right), \tag{3}$$

where W (in Hz) is the system bandwidth, and the signal-to-interference-plus-noise ratio (SINR) is the second term inside the bracket, which, in the sequel is represented by $\gamma_{n,j,k}$. The term $p_{n,j,k}$ is the transmission power of the k-th user on the j-th PRB on gNB n, $g_{n,j,k}$ represents the overall gain between the k-th user and the n-th gNB, which includes

the antenna gain, the shadow fading and the path loss. The expression $\sum_{m\neq k, m\in K} p_{j,m} \cdot g_{n,j,m}$ is the sum of the conglomerated interference from the other user, but the k-th; whereas σ^2 denotes the Gaussian noise power. Due to critical URLLC applications such as autonomous vehicles, where obstacles are frequently encountered along the signal path, the transmission power function is defined by channel gains measured over short time scales, hence Rayleigh fading [30] is considered.

A. THE OPTIMIZATION PROBLEM

Assuming that the lifetime of each slice type to be an exponentially distributed random variable [31], and slice acceptance is based on the preference of the MNO regarding the SINR requirements. Then, using the dynamic resource percentage threshold scheme, a dynamic RA objective to maximize the UL capacity of eMBB services under the constraint of maximum transmission power of users and guaranteed data rate for URLLC users is given as follows:

$$r^{(1)}_{n,j,k} = \frac{\beta \cdot \gamma^{th}_{1,n}}{\lambda_{1,n}} \cdot W \cdot \log_2(1+\gamma_{n,j,k}), \quad \forall k \in \mathcal{K}_1 \tag{4}$$

where $\frac{\beta\cdot\gamma^{th}_{1,n}}{\lambda_{1,n}}$ represents the fraction of the bandwidth remaining for eMBB slice. This means that the eMBB slice receives $\frac{\beta\cdot\gamma^{th}_{1,n}}{\lambda_{1,n}} \cdot W$ of the total gNB resources. Once the resources have been reserved for the URLLC slice, the remaining task is to maximize the per-user data rate in the eMBB slice. To achieve this objective, a single-objective optimization problem can be defined as follows:

$$P : \arg\max_{P} \sum_{n=1}^{N}\sum_{j=1}^{J}\sum_{k=1}^{K} r^{(1)}_{n,j,k} \tag{5}$$

subject to

$$\begin{aligned} C1 &: \sum_{j=1}^{J} p_{j,k} \leq p_{max}, \quad \forall k \in \mathcal{K} \\ C2 &: \sum_{j=1}^{J} r^{(2)}_{n,j,k} \geq r^{(2)}_{req}, \quad \forall k \in \mathcal{K}_2 \end{aligned} \tag{6}$$

where the decision variable $\mathbf{P}$ in (5) represents the set of transmission powers, which has been translated from $\gamma_{n,j,k}$. The constrain **C1** imposes a restriction on the maximum transmission power for all users, i.e., $k \in \mathcal{K}$, allocated PRBs in the n-th gNB. It must be noted that by ensuring that its UL transmission power is less than or equal to the maximum allowed transmission power, p_{max}, the interference conditions will be met. The constraint **C2** ensures that the minimum required data rate for URLLC users, i.e., $k \in \mathcal{K}_2$, is always met. Here, it is assumed that each $\mathcal{K}_2$ user generate only one service flow at a time with a required service rate $r^{(2)}_{req}$. Thus, $r^{(2)}_{req}$ represents the minimum data rate threshold to guarantee the data rate for URLLC users, whereas $r^{(2)}_{n,j,k}$ is the data rate for all $k \in \mathcal{K}_2$.

B. THE DYNAMIC RESOURCE PERCENTAGE THRESHOLD SCHEME

The improved dynamic resource percentage threshold scheme allocates slice resources based on the total number of URLLC users currently being served by each gNB. The admission control condition for the allocation of slice resources to users is employed taking into account their interference levels. Using this condition, when an eMBB user requests an UL connection, the gNB has to check if by accepting the new eMBB connection request it will meet the admission control condition. This is accomplished by calculating the new admission condition, $\beta_{new} = \frac{\beta\cdot\gamma^{th}_{1,n}}{\lambda_{1,n}+1}$, whose sole purpose is to prevent the gNB from being overloaded resulting into low data rate. This admission control condition operates by increasing the number of eMBB users by 1, then compares the result with the admission bandwidth threshold for the eMBB slice, i.e., $\beta^{(1)}_{1,n}$. This admission bandwidth threshold is the minimum equi-spaced channel per user as discussed in [9], and the admission constraint for eMBB users that ensures the protection of URLLC services on slice $\mathcal{K}_2$ is imposed as follows:

$$\beta_{new} \geq \beta^{(1)}_{1,n}. \quad \forall n \in \mathcal{N} \tag{7}$$

This admission constraint can be illustrated using an admission control probability as follows [32]:

$$\phi_n = \left\{ \begin{array}{ll} 1, & \beta_{new} \geq \beta^{(1)}_{1,n} \\ 0, & \text{otherwise} \end{array} \right\}. \tag{8}$$

Therefore, the total number of eMBB users associated with the n-th gNB is be given as follows:

$$\lambda_{1,n} = \frac{\beta \cdot \gamma^{th}_{1,n}}{\beta_{new}} - 1 = \beta\gamma^{th}_{1,n}(\beta^{(1)}_{1,n})^{-1}. \tag{9}$$

So, if $0 \leq \lambda_{1,n} = \frac{\beta\cdot\gamma^{th}_{1,n}}{\beta_{new}} - 1 \leq \beta \cdot \gamma^{th}_{1,n}(\beta^{(1)}_{1,n})^{-1}$, it means that the n-th gNB is under-loaded in terms of eMBB users, thus the admission probability equals 1. However, if $\frac{\beta\cdot\gamma^{th}_{1,n}}{\beta_{new}} - 1 > \beta \cdot \gamma^{th}_{1,n}(\beta^{(1)}_{1,n})^{-1}$, it means the n-th gNB is overloaded and the connection request has to be rejected. Then, in order to balance the load over the set of gNBs, the SDN controller employs an immediate retry procedure, whereby users rejected may attempt admittance and gain service in their respective slices from nearby gNBs. At a departure instant of any connection, the corresponding PRBs are released, and system state transition occurs and admitting another slice user awaits to take the state back to occupied.

C. TRANSFORMATION THROUGH COMPLEMENTARY GEOMETRIC PROGRAMMING

To solve the objective function in (5), which is non-convex due to inter-site interference, it must be first transformed into its linear form. In order to transform the non-convex function into a convex function, power control by complementary geometric programming [33] is employed to transform it into a convex function. Complementary geometric

programming is based on successive approximation and even here we use geometric programming for power allocation. Here, the gNB dynamically adjusts the resources allocated for eMBB users, which first guarantees data rates for URLLC users before allocating the remaining resources to eMBB users. Furthermore, eMBB users will be admitted only if they meet the admission control condition set in (7) and the SINR minimum requirement, γ_0, which is based on the SINR of users. As stated earlier, the eMBB users are supposed to meet this minimum requirement at the gNB in order to be accepted in case there are still resources available in their designated slice. At his point, an SINR-based admission control condition is formulated as follows:

$$\gamma_{n,j,k} \geq \gamma_0, \tag{10}$$

where γ_0 represents the SINR constraint imposed by the gNB. The gNB computes the resource percentage threshold, $\rho_{1,n}$, which is based on the total number of URLLC users that are currently being served and their minimum required data rate, $r_{req}^{(2)}$. This is specifically computed by the n-th gNB for eMBB users is as follows:

$$\gamma_{1,n}^{th} = 1 - \rho_{1,n}, \tag{11}$$

where

$$\rho_{1,n} = \frac{r_{req}^{(2)}}{\sum_{u=1}^{N^{(2)}} \frac{\beta}{N^{(2)}} \log_2(1+\gamma_{n,j,k})}. \tag{12}$$

Once this is done, power control by complementary geometric programming is used to compute the optimized UL capacity for eMBB users, which requires some lower bound substitution and variable transformation. The problem in (5) can then be transformed into its convex equivalent by using a relaxation approach, i.e., introducing alternative variables and approximations. At this point, a lower bound which is tight with equality at a chosen value of γ_0 is obtained as follows:

$$\varrho \cdot \log \gamma_0 + \chi \leq \log(1+\gamma_0), \tag{13}$$

where ϱ and χ are fixed approximation parameters defined as follows:

$$\varrho = \frac{\gamma_0}{1+\gamma_0}, \quad \text{and} \quad \chi = \log(1+\gamma_0) - \varrho \cdot \log \gamma_0. \tag{14}$$

Then, using α and χ in (14), the lower bound of (4) can be reformulated as follows:

$$\hat{r}_{n,j,k}^{(1)} = \frac{\beta \cdot \gamma_{1,n}^{th}}{\lambda_{1,n}} \cdot \varrho \cdot \log_2 \gamma_{n,j,k} + \chi. \tag{15}$$

Therefore, the original optimization problem in (5) can be transformed to maximize the UL capacity under the constraint of maximum power transmission of users and guarantee data rates for URLLC users per gNB. However, (15) is still non-convex and still requires some further transformation. At this point, as stated in [33], the lower bound can be transformed into convex by letting $p_{j,k} = e^{\hat{p}_{j,k}}$ in (15) and $\hat{p}_{j,k} = \ln p_{j,k}$. Then, (15) can be reformulated as follows:

$$Z_{n,j,k} = \frac{\varrho}{\ln 2}\left[\ln g_{n,j,k} + \hat{p}_{j,k} - \varphi\right] + \chi, \tag{16}$$

where

$$\varphi = \ln\left(\sum_{m \neq k} e^{(\hat{p}_{j,m})} g_{n,j,m} + \sigma^2 + \eta\right), \tag{17}$$

where

$$\eta = \sum_{m \in M_n^{neighbor}} e^{\hat{p}_{j,m}} \cdot g_{n,j,m}. \tag{18}$$

Then, by substituting (17) and (18) into (16), a "$\log - \sum - \exp$" function is observed, which was proven to be convex in [33]. Therefore, after the lower bound variable transformation and approximation, the initial optimization problem in (5) can be reformulated as follows:

$$P^* : \arg\max_{P} \sum_{n=1}^{N} \sum_{j=1}^{J} \sum_{k=1}^{K} \tilde{r}_{n,j,k}^{(1)}, \tag{19}$$

subject to

$$\begin{aligned} C1^* &: \sum_{j=1}^{J} p_{j,k} \leq p_{max}, \quad \forall k \in \mathcal{K} \\ C2^* &: \sum_{j=1}^{J} \tilde{r}_{n,j,k}^{(2)} \geq r_{req}^{(2)}, \quad \forall k \in \mathcal{K}_2 \end{aligned} \tag{20}$$

where

$$\tilde{r}_{n,j,k}^{(1)} = \hat{r}_{n,j,k}^{(1)}(e^{\hat{p}_{j,k}}; \varrho, \chi). \tag{21}$$

This means that the variation of resources reserved for eMBB is based on the number of URLLC users that are currently being served, and the conditions stated in (7) and (10) are adopted in this scenario. In this case, the overall SINR at the n-th gNB can be represented as follows:

$$\gamma_{n,j,k} = \frac{p_{j,k} g_{n,j,k}}{\sum_{m \neq k, m \in M_n} p_{j,m} g_{n,j,m} + I_{n,j}^{inter} + \sigma^2}, \tag{22}$$

where $I_{n,j}^{inter}$ is the cluster interference. However, due to the existence of other gNBs, the SINR at gNB n changes, thus substituting (22) into (4), which is a lower-bound substitution, the function is still non-convex. Therefore, the function requires some further transformation into a convex through another substitution and observing the "$\log - \sum - \exp$" function, which is time consuming and computationally complex. In order to address this issue, this problem is solved as a constrained problem discussed in [34].

V. THE PROPOSED CONSTRAINED REINFORCEMENT LEARNING SCHEME

In order to achieve the near-constraint satisfaction of both slices, a constrained RL strategy, which allows the algorithm policy, π^*, to guarantee the slicing policy, $\pi(r)$, behavior throughout the observation period T. The initial stage of the proposed strategy operates similar to the dynamic RPT approach in the sense that it is also based on the SINR requirement, which then translate to power and rate allocation. Thus, at the gNB the SINR can be defined as follows:

$$\gamma_0^{(2)} = \frac{p_0 p_0^{(2)}}{\sum_{k=1}^{K} p_k g_k^{(2)} + \sigma^2}, \tag{23}$$

where p_0 is the transmission power of the gNB, assumed to be at constant power, $g_0^{(2)}$ represents the channel gain between the URLLC users and the gNB, p_k is the transmission power from all users associated with the gNB, and g_k is their corresponding channel gains. The SINR of the k-th user can be defined as follows:

$$\gamma_k^{(1)} = \frac{p_j g_j^{(1)}}{\sum_{j \neq k} p_k g_k^{(1)} + p_0 g_0^{(1)} + \sigma^2}, \tag{24}$$

the term $g_0^{(2)}$ is the channel gain between the URLLC user and the gNB, $g_k^{(2)}$ is the channel gain of the k-th eMBB user to the gNB, $g_0^{(1)}$ is the channel gain between the URLLC user and the j-th gNB, p_j denotes the transmit power of the j-th eMBB user, $g_j^{(1)}$ represents the channel gain of the j-th eMBB user. In order to achieve the objectives of network slicing, the following SINR constraints are imposed:

$$\gamma_0^{(2)} \geq \gamma_0, \quad \text{and} \quad \gamma_k^{(1)} \geq \gamma_k, \quad k \in \mathcal{K}, \tag{25}$$

where γ_0 is the SINR threshold for URLLC users, while γ_k is the SINR threshold for eMBB users, such that the overall transmission power is given, tight with equality, as follows:

$$p_k = \frac{\vartheta_k(\sigma^2 + g_0^{(1)} p_0)}{g_j^{(1)}(1 - \sum_{k=1}^{K} \vartheta_k)}, \quad \text{where} \quad \vartheta_k = \left(1 - \frac{1}{\gamma_k}\right)^{-1}. \tag{26}$$

In order to ensure that the SINR thresholds are met, a valid power allocation expression must be derived. To obtain a valid power allocation expression, the condition $1 - \sum_{k=1}^{K} \vartheta_k > 0$ must first be met. After replacing the eMBB transmission powers from (26) into (23), the SINR constraints in (25) can be represented as follows:

$$\sum_{j=1}^{K} \vartheta_j \alpha_j \leq 1, \quad \text{where} \quad \alpha_j = \left\{ \frac{g_j^{(2)}(\sigma^2 + g_0^{(1)}) p_0}{g_j^{(1)}(g_0^{(2)} p_0/\gamma_0 - \sigma^2)} + 1 \right\}. \tag{27}$$

where α_j is obtained after some algebraic manipulations. Thus, by properly adjusting the parameter γ_k, the transmission rate of devices on the eMBB slice, $r_k^{(1)}$, will automatically be adjusted. At this point, the eMBB users will try to obtain and maintain an optimal power assignment that meets the interference constraints in (25), thereby maximizing their performance. Thus, the problem of finding a transmission power and corresponding bit rate has now become that of finding an optimal SINR, $\hat{\gamma}_k$. Based on the assumption of constant power at the gNB, the term γ_0 is assumed to be constant, as a result γ_k is the one that needs to be adjusted by slice users in order to meet the conditions in (26) and (27). This is achieved by trying to adapt the transmission rate as follows: (i) using the relationship between the modulation scheme and the SINR of $\mathcal{K}_1$ users, and (ii) using the relationship between the delay and outage probability for $\mathcal{K}_2$ users.

A. RESOURCE ALLOCATION MODEL FOR eMBB SERVICES

Due to the transmission latency tolerance of the eMBB communication, the achievable bit rate is a more practical measure of the throughput. For instance, when user $k \in \mathcal{K}_1$ sends a connection request, the bit rate achieved at the gNB can be determined as follows:

$$r_k^{(1)} = \sum_{k \in \mathcal{K}_1} \psi_{j,k} \cdot r_{n,j,k}^{(1)}, \tag{28}$$

where $\psi_{j,k} \in [0, 1]$ is user association as defined earlier in (7), and $r_{n,j,k}$ is the transmission rate as defined in the previous section, which, for eMBB users is given as follows:

$$r_{n,j,k}^{(1)} = W_1 \cdot \log_2(1 + \varsigma \cdot \gamma_{n,j,k}), \tag{29}$$

with W_1 representing the bandwidth allocation for the eMBB slice, the expression $(1 + \varsigma \cdot \gamma_{n,j,k})$ shows the number of bits contained in a modulation symbol. This takes only a small number of integer values in practice, while the constant ς relates the SINR to the target transmit bit error rate (BER) requirement [36]. This means that eMBB devices have to explore the set of available SINRs that also match with their match with both their BER requirement. It must, however, be noted that when the eMBB devices adjust their $\gamma_{n,j,k}$, they consequently adjust their modulation schemes, as well as their transmission rates, $r_{n,j,k}^{(1)}$.

B. RESOURCE ALLOCATION MODEL FOR URLLC SERVICES

A URLLC created slice is a delay-critical slice that serves users with strict delay requirements such as autonomous vehicles [37]. The scheduling of an unexpected packet generation by URLLC users is one of the most important issues in the proposed mechanism, since an arriving URLLC is stored in a specific transmission buffer. The transmission of each packet takes no less than one TTI. However, the stochastic nature of channel conditions, payload size, and availability of resources are the main challenges towards achieving the stringent latency requirements. This challenge may end up forcing the scheduling to increase the TTI of a packet [14]. Therefore, a proper URLLC communication model must be able to ensure high reliability by overcoming the variations in channel conditions. To achieve this, the outage probability for each user on the URLLC slice needs to be achieved using a proper delay constraint [38]. Then, from each data

packet sequence, a maximum expected delay, $d_{k,max}$, must be fulfilled. Assuming that the queue has a backlog of $q_{2,n}(t)$, and the URLLC traffic is arriving at a rate $\lambda_{2,n}$ per slot, the queue revolves as follows:

$$q_{2,n}(t+1) = \max\{q_{2,n}(t) + \lambda_{2,n} - r^{(2)}_{n,j,k}, 0\}. \quad (30)$$

where $t+1$ represents the next slot, and $r^{(2)}_{n,j,k}$ is the slice rate for serving traffic admitted based on the admission control in (24). Using Little's theorem, the average delay of the queue can be obtained as $d_k = (q_t + \lambda_{2,n})/r^{(2)}_{n,j,k}$. Then, the corresponding delay outage probability of of the user $k \in \mathcal{K}_2$ can be given as follows:

$$Pr\{d_k \geq d_{k,max}\} = e^{-(r^{(2)}_{n,j,k} - \lambda_{k,max})d_{k,max}}, \quad (31)$$

where $\lambda_{k,max}$ is the maximum data arrival rate per slot. in $\mathcal{K}_2$.

C. THE OPTIMIZATION PROBLEM

Due to the resource limitation at the gNB, when the resources at the gNB cannot provide service for all the associated users, the optimization goal is to maximize the sum of the total satisfaction degree for all users [39]. In order to maximize the utilization of network resources, the average satisfaction of both slices in the network can be formulated as follows:

$$R_{(\gamma_k)} = \frac{1}{K}\left(\sum_{\forall k \in \mathcal{K}_1} r^{(1)}_k + \sum_{\forall k \in \mathcal{K}_2} r^{(2)}_{n,j,k}\right). \quad (32)$$

Then, maximizing this average satisfaction for both traffic types require an optimization problem formulated as follows:

$$P^{**}: \hat{\pi}_k = \arg\max_{\forall \pi^* \in \Pi} \sum_{i=1}^{K} R_{(\gamma_k)}(\pi^*), \quad (33)$$

subject to

$$\begin{aligned}
C1^{**}&: \sum_{k=1}^{K} \vartheta_{(\gamma_k)}(\pi^*) \leq 1 - \epsilon,\\
C2^{**}&: \sum_{m \neq k} \alpha_m \vartheta_{(\gamma_m)}(\pi^*) \leq 1,\\
C3^{**}&: \sum_{k=1}^{K}\sum_{n=1}^{N} \psi_{n,k} = N, \quad \psi_{n,k} \in \{0,1\},\\
C4^{**}&: \sum_{n=1}^{N} \psi_{n,k} r^{(1)}_{n,j,k} \geq \sigma_0, \quad \forall k \in \mathcal{K}_1,\\
C5^{**}&: Pr\{d_k \geq d_{k,max}\} \leq \epsilon_0, \quad \forall k \in \mathcal{K}_2, \quad (34)
\end{aligned}$$

where Π in P^{**} represents the set of all possible policies, while the constraint in $C1^{**}$ enforces the power allocation condition. The constraint $C2^{**}$ is from (27) and puts an upper-bound on the aggregate interference from the eMBB users - but the k-th. $C3^{**}$ makes sure that all users make full use of the available PRBs; $C4^{**}$ enforces the transmission rate of user k to stay above the transmission rate requirement in the eMBB slice; and the constraint $C5^{**}$ guarantees the outage probability is lower than ϵ_0. This means that the SDN controller must estimate the appropriate transmission rates to make sure that the URLLC outage probability does not reach the packet error constraint, ϵ_0. Thus, at each time instant, the state space of the system is defined as $s(t) = \{I_k(t), \mathcal{I}(t)\}$, where $I_k(t) \in [0, 1]$ is a binary indicator specifying whether or not the k-th user is causing aggregated interference.

D. SOLVING THE CONSTRAINED PROBLEM

Since (33) takes the form of a constrained Markov decision process (CMDP), a constrained policy optimization method, which is a general-purpose policy search algorithm from constrained RL [21], is proposed. At this point, each eMBB device observes the state space, $s(t) \in \mathcal{S}$, and executes an appropriate action by following a policy, $\pi(s(t), a(t))$, $\pi \in \Pi$. The information in $C1^{**}$ and $C2^{**}$ is supposed to be communicated by the eMBB devices to their associated gNB n. Based on this information, the eMBB devices are instructed to adjust their transmission power levels by selecting possible actions from a set of SINRs, where the finite discrete space of candidate SINRs is represented as follows:

$$\mathcal{A} = \{\gamma_1, \gamma_2, \cdots, \gamma_i, \cdots, \gamma_K\}, \quad \gamma_i \in \Gamma \quad (35)$$

where Γ is a finite set of possible SINR levels. By following a strategy, $\pi(s(t), a(t))$, each eMBB device has the task of searching from the finite discrete space containing all candidate SINRs in order to obtain the optimal solution, $R_{(\gamma_k)}(\pi^*)$. This is an immediate return obtained using the command assignment $a(t) \in \mathcal{A}$ while the system is in state $s(t) \in \mathcal{S}$. At this point, the optimization task is to obtain the policy that maximizes the received discounted reward, $\gamma^t R_{(\gamma_k)}(\pi^*)$, where γ^t denotes the discount factor. Letting $s_0 = s(t)$ denote the initial or start-state of the system, the finite horizon expected discounted reward can be represented using a state-value function as follows:

$$V(s(t), \pi^*) = \sum_{t=0}^{T-1} \gamma^t \mathbb{E}\left[R_{(\gamma_k)}(s, a)|\pi^*, s_0\right]. \quad (36)$$

Then, the eMBB slice users can repeatedly make their decisions that finally allow them to obtain their optimal policies that lead to the maximization of the expected sum of discounted rewards. Through the use of the Bellman optimality principle [22], a solution for (36) can be obtained by taking the optimal action assuming that all possible strategies by other devices are optimal. The maximization of the expected sum of rewards is expressed as follows:

$$V^*(s, \pi^*) = \max_a \left[R_{(\gamma_k)} + \gamma^t \sum_{s'} p(s'|s, a) V(s', \pi^*)\right], \quad (37)$$

where $p(s'|s, a)$ represents the transition probability from the current state $s(t)$ towards the next state of the system, $s' = s(t+1)$, after taking optimal action a. In (37), the optimal policy, π^*, is related to the state-value function, $V^*(s, \pi^*)$, as follows:

$$\pi^*(a|s) = \arg\max_a \left\{R_{(\gamma_k)} + \gamma^t \mathbb{E}_{s'|s,a}\left[V^*(s')\right]\right\}, \quad (38)$$

where $\mathbb{E}_{s'|s,a}[\cdot]$ is the conditional expectation operator with respect to the state-value function $V^*(s, \pi^*)$. The state transitions are determined by the power allocation that results in

a return, i.e., either a reward or a penalty. To proceed from here, another version of the Bellman optimality equation, equivalent to (37), is employed and is stated as follows:

$$V(s) = \max_{\pi(\cdot|s)\in\mathcal{P}_{\mathcal{A}}} \mathbb{E}_{a\sim\pi(\cdot|s)}\left[R_{(\gamma_k)} + \gamma^t \mathbb{E}_{s'|s,a}[V(s')]\right], \tag{39}$$

where $\mathbb{E}_{a\sim\pi(\cdot|s)}$ is the expectation of selecting an action while following the policy $\pi(\cdot|s)$. Here, it can be seen that (39) makes the role of the policy, π^*, very explicit, which creates distance discrepancy between the left and right hand-sides. the max operator over $\mathcal{P}_{\mathcal{A}}$ induces some non-smoothness to the objective function such that any slight change in the state-value function, V, causes a large difference in (37). In other words, the max operator causes some instability in the optimization process. In order to minimize this discrepancy, (39) may be jointly optimized over $V^*(s,\pi^*)$ and π^* to lead to a convex function. To achieve this and subsequently arrive at a convex function, the square distance needs to be minimized by minimizing the squared Bellman error as follows:

$$V^*(s) = \min_V \mathbb{E}_{s\sim\delta} \left[\left(\max_{\pi(\cdot|s)\in\mathcal{P}_{\mathcal{A}}} \mathbb{E}_{a\sim\pi(\cdot|s)}\left[R_{(\gamma_k)} + \gamma^t \mathbb{E}_{s'|s,a}[V(s')]\right] - V(s)\right)^2\right], \tag{40}$$

where the parameter δ denotes a distribution such that $\delta(s) > 0, \forall s \in \mathcal{S}$. It must, however, be noted that for $\delta = 0$ is still the original Bellman equation. In this way, the reward has been shaped and a reward function equivalent of an MDP is obtained. Thus, the parameter δ can be viewed as trying to control the degree of smoothing, such that smoothing the Bellman operator requires $\delta > 0$. To solve the instability and discontinuity that is as a result of the max operator, the Nesterov smoothing technique is used to smooth the Bellman operator. However, since the policy, π^*, is a conditional distribution over the set of actions, $\mathcal{A}$, an entropy regularization is used such that (37) is rewritten as follows:

$$V_\delta(s) = \max_{\pi(\cdot|s)\in\mathcal{P}_{\mathcal{A}}} \left(\mathbb{E}_{a\sim\pi(\cdot|s)}(R_{(\gamma_k)} + \gamma^t \mathbb{E}_{s'|s,a}\left[V_\delta(s')\right]) + \delta H(\pi, s)\right), \tag{41}$$

where $H(\pi^*, a) = -\sum_{a\in\mathcal{A}} \pi^*(a,s)\log\pi^*(a|s)$. Since negative entropy is defined as the conjugate of the "$\log-\sum-\exp$" function, the objective in (37) can be equivalently reformulated in accordance with the narrative of Theorem 2.2 in [23] as follows:

$$V_\delta = \delta \log\left(\sum_{a\in\mathcal{A}} \exp\left(\frac{R_{(\gamma_k)} + \gamma^t \mathbb{E}_{s'|s,a}\left[V_\delta(s')\right]}{\delta}\right)\right), \tag{42}$$

where the "$\log-\sum-\exp$" observed in (42) is an effective smoothing approximation of the max operator. However, due to the practical convenience of working with the Q-function instead of the state-value function V^*, the $V^*(s,\pi^*)$ in (37) can be approached by a Q-function that is updated as follows:

$$Q^*(s,a) = \bar{\alpha}_t Q(s,a) + \alpha_t\left[R_{(\gamma_k)} + \gamma^t Q^*(s')\right], \tag{43}$$

where $\bar{\alpha}_t \triangleq (1-\alpha_t)$, $0 < \alpha_t < 1$ is the learning rate, and $Q^*(s')$ is the Q-value of the k-th eMBB device corresponding to the maximum $Q^*(s') = \max_b Q(a,s')$ in the new state s' after selecting and performing the action a.

VI. DISCUSSION AND COMPUTATIONAL COMPLEXITY OF ALGORITHMS

A. DYNAMIC RESOURCE PERCENTAGE THRESHOLD ALGORITHM

The dynamic resource percentage threshold algorithm begins by allocating resources for the URLLC services, then reserves the remainder for eMBB services. This process, which is assumed for both the algorithms, assumes that URLLC services are deserving of high priority treatment. The procedure for the dynamic resource percentage threshold scheme is outlined in **Algorithm 1** below.

Algorithm 1 Procedure for the Dynamic Resource Percentage Threshold Algorithm

Input: $\beta, p_{max}, \sigma^2, \beta_n^{(1)}$
Output: $r_{n,j,k}^{(1)}$
Initialize input parameters
For $n = 1 : N$ **do**
  For $j = 1 : J$ **do**
    For $k = 1 : K$ **do**
      Compute the channel gain, $g_{n,j,k}$
      Use $\gamma_{1,n}^{th}$ and β to compute the number of users that can be admitted into the eMBB slice using (9)
      When a URLLC user is admitted, compute new $\rho_{1,n}$ under the current SINR using (12)
      If eMBB user request uplink **then**
        Compute new admission condition, β_{new}
        If $(\beta_{new} \geq \beta_{1,n}^{(1)})$ **then**
          Accept eMBB user into eMBB slice
          Compute $r_{n,j,k}^{(1)}$ after lower bound and variable transformation.
          Maximize $r_{n,j,k}^{(1)}$.
        Else
          Reject eMBB user
        End If
      End If
      Re-compute $\gamma_{1,n}^{th}$ under the current SINR's of associated users
    End For
  End For
End For

In this algorithm, an admission control policy is first derived in order to obtain the number of users to be admitted into the

eMBB slice. This is achieved through (9). This step is independent of the kind of traffic that each eMBB device is offering to the system, hence is has a computational complexity of order 1, i.e., $\mathcal{O}(1)$. After the SINR-based admission control condition has been obtained, the resource percentage threshold is computed using (12). It must, however, be noted that the objective function in (5) is non-convex, which makes its optimization process NP-hard. Since the proposed dynamic RPT optimization problem is inherently non-convex and NP-hard, it is beneficial to reduce and relax it to transform it into its convex alternative.

In order to achieve convexity, successive convex approximation were performed using a complementary geometric programming technique, which is an efficient two-step iterative approach, whereby for a given power allocation, the first step is to derive the user association. Then, using the obtained user association, the second step derives an optimum power allocation using complementary geometric programming. The complementary geometric programming is employed to optimize the power control problem to maximize the number of successfully admitted users. This operation requires the location indices of the different users with respect to the associated gNB. This process, which appears in **line 10 to line 18** of **Algorithm 1** is the one that contributes more into the increase in computational complexity, more especially the time complexity. The variation of resources reserved for the eMBB slice is finally found in (21) based on the total number of users that are currently being served by the URLLC slice.

The objective function in (5) is still non-convex even after this transformation, this is justified by substituting (22) into (4). Although it is beneficial to reduce and relax the optimization problem in (5) into its convex form, there is certainly a cost associated with doing so. Therefore, the computational cost in terms of longer algorithm convergence times or more cycles required towards convergence are expected. The convex constraint sets that are required always make the complexity grow exponentially with the number of variables such as schedulability constraints. It must, however, be noted that this is similar to a fixed learning policy, whereby the process of learning the states and the actions is a stationary Markovian.

B. COMPLEXITY OF THE PROPOSED CONSTRAINED REINFORCEMENT LEARNING SCHEME

In the RL strategy, the complexity comes with the repeated negotiations, although the strategy/automaton of the game has to be included when computing the complexity in the equilibrium concept. The equilibria differs along three task complexity measures, i.e., (ii) the cardinality of the choice space, where a stage is equivalent to an information set that is facing the player along the path leading to an equilibrium; (ii) the level of iterative knowledge of rationality, and (iii) the level of iterative knowledge of strategy. With the dynamic version of the algorithm, a non-stationary Markovian process results, and the algorithm converges in the sense that there is no history of the learning process. In this case, the process of learning the states and actions eventually becomes stationary Markovian. However, the sampling distribution could be replaced by the stationary distribution of the underlying stationary Markovian process, which is an observation that brought out the need for a dynamic learning algorithm. The procedure for the constrained RL strategy that demonstrates the effectiveness of the CMDP on the Q-learning algorithm in network slicing is outlined in **Algorithm 2** below.

Algorithm 2 Procedure for the Proposed Constrained Reinforcement Learning Algorithm

Input: $\beta, p_{max}, \sigma^2, \beta_{1,n}^{(1)}, \alpha_t, \gamma^t, \epsilon, \delta$
Output: $r_{n,j,k}^{(1)}, r_{n,j,k}^{(2)}, \pi^*, R_{(\gamma_k)}$.
01: Initialize learning parameters, α_t, γ^t, ϵ, and δ
02: $\forall s \in \mathcal{S}$ and $a \in \mathcal{A}$, initialize $Q(s, a) = 0$
03: Using steps 02 to 07 in **Algorithm 1**, obtain the optimal user association
04: **For** each iteration **do**
05: Use the user association to obtain the optimal power allocation using (26)
06: **For** each TTI **do**
07: Observe power control condition using (27) & create system state $s(t) = [I_k(t), \mathcal{I}(t)]$
08: Formulate return using satisfaction in (32)
09: Determine possible action $a(t) \in \mathcal{A}$ in (35)
10: Use the policy $\pi(s(t), a(t))$ to select appropriate action to maximize reward $\gamma^t R_{(\gamma_k)}(\pi^*)$ and observe $V_i^*(s, \pi^*)$ in (37)
11: Use optimal policy in (38) to smooth the Bellman operator, i.e., (39) - (41)
12: Observe the $\log - \sum - \exp$ function in (42)
13: **End For**
14: Update $Q^*(s, a)$ by using (43)
15: Increment timer and move system state to s'
16: **End For**

1) ALGORITHM INITIALIZATION

The formulation of the resource allocation problem using the constrained RL formulation is identical to the CMDP optimization problem of the general finite-horizon CMDP problem in (40). The algorithm parameters are initialized in **line 01** and the initialization of the Q-function to zero, i.e., $Q(s, a)$, in **line 02**, which entails the starting point of the algorithm. This means that the Q-table is initialized with zeros, i.e., no learning history, which is done to avoid undirected exploration such that the algorithm immediately works towards reward maximization. After the initialization, **line 03** of **Algorithm 2** calls **line 02 - 07** of **Algorithm 1** for SINR adjustment and optimal user association which aids in power allocation as stated in **line 05**. This step involves matrix-vector multiplications, which have a complexity of $\mathcal{O}(n^2)$ [42]. In this case, the first Q-value that experiences a change is the one for or related to the action that leads to the reward state. The choice of smaller learning rates, i.e.,

$\alpha_t = 0.1$, is to avoid speeding up the process to convergence at a local optimization solution - typical of higher learning rates. The discount factor, γ^t, which affects how much weight must be given to future rewards in the value function, was set to 0.9. This is because a value $\gamma^t = 0$ results in state-action values that represent immediate rewards, whereas higher values tend to represent the cumulative discounted future rewards an agent is expecting to receive by behaving under the policy π^*.

2) ACTION EXPLORATION AND EPISODIC REWARD FUNCTION

In terms of action selection, a persistent exploration learning policy, π^*, that stores information about the relatedness of states and actions in the Q-table, is used. From the action set, $\mathcal{A}$, of candidate SINRs, the algorithm selects the best action that translates to the optimal transmission power, subsequently to achieving $\gamma^t R_{(\gamma_k)}$. The traditional ϵ-greedy approach is used with respect to the estimated Q-function with a probability as stated in [41]. The reason for this is to make sure that all state-action pairs are explored enough before the algorithm converges to a particular decision. Using the power allocation result, observing the system state is first step of the inner iteration, which happens every TTI. This step avoids too much interference on other users, which leads to the formulation of the satisfaction for users of each slice. The action selection step in **line 09** is followed by **line 10**, which implements the exploration rule that defines the next state the system has to go to next. This includes the Q-values for all the actions, and as it is stated in [43], the number of steps executed by the algorithm is always bounded by an expression that depends only on its initial and current Q-values; whose computational complexity is well discussed in [45]. Using the proposed constrained RL algorithm, the computational complexity of action selection which is well documented in [45] was substantially reduced. Due to the role of the policy, π^*, being made explicit in (39), a distance discrepancy is created resulting into a Bellman error. This discrepancy is minimized by optimizing (39) over V and π, which results in the minimization of the squared Bellman error in (40).

3) UPDATING Q-VALUES AND REACHING THE REWARD STATE

Reaching the reward state requires resolving the Bellman error for better algorithm convergence, which requires a few additional steps from (36). This is done in acknowledgement of the existence of the Bellman error, which leads to unstable solutions. It must be noted that the traditional RL strategy ignores the existence of the Bellman error and only uses a single update step to adjust $Q(s, a)$ and arrive at a solution. Then an immediate reward $R(s, a) \in \mathcal{R}$ is obtained. If the agent starts in state $s \in \mathcal{S}$ and proceeds to execute actions for which it receives immediate reward, R_t, at time step t, the total reward that it would receive over its lifetime is $R_{(\gamma_k)}$, as stated by **line 10**. The smoothing of the Bellman operator in **line 11** leading to the convex observation in **line 12**, the effective finite horizon power control condition is supposed to increase the computational complexity of the algorithm. Then, after resolving the instability issue through the Nesterov smoothing technique, the average reward of the Q-learning algorithm with respect to the episodic allocation of network slice resources is observed through the "$\log - \sum - \exp$" function in (42). Throughout the iterations of the Q-learning algorithm, the Q-function is updated using (43). The calculation indicate that the law of iterated logarithm holds for the learning process underlying the constrained RL strategy leads to (42). Thus, with a discount factor $\gamma^t = 0.9 > 0.5$, the asymptotic convergence of the Q-learning algorithm used by the constrained RL strategy is $\mathcal{O}(n^2)$. Immediately the agent begins approaching the reward state, the number of steps can be exponential in the number of states. Then, the Q-function value of each state-action pair can be augmented with an estimate of its uncertainty to guide exploration, and to achieve faster learning and a higher reward during learning. After the smoothing of the Bellman operator, the state-action pairs are populated and $Q(s, a)$ is updated as shown in **line 14**. This value, i.e., $Q(s, a)$, is then used in the approximation of the optimal total reward received. If $C5^{**}$ is not true, which means URLLC applications will suffer delays, then $Q(s, a)$ is adjusted using information local to the previous state. Then, due to the process of transfer learning, the worst-case computational complexity of this stage becomes quadratic, i.e., $\mathcal{O}(n^2)$, which is lower than the upper bound on the complexity of the Q-learning algorithm. The state-of-the-art RL strategies in RA states that the computation time cannot be upper-bounded by less than $\mathcal{O}(n^3)$.

VII. PERFORMANCE EVALUATION

This section presents the numerical results of the proposed algorithm considering gNBs using hybrid access mode.

A. SIMULATION PARAMETERS

The performance evaluation is for semi-stationary users with a system bandwidth of 100 MHz centered on a component carrier frequency of 3.5 GHz. For the sake of making this work repeatable by other researchers, the non-default simulation parameters used in the proposed algorithm are tabulated in Table 2 below.

TABLE 2. Simulation parameters.

Parameter	Value	Unit
Component carrier frequency	3.5	GHz
Component bandwidth	100	MHz
Number of gNBs, N	20	-
Number of sub-channels	5	-
gNB transmit power	20	dBm
Max No. of users per gNB, K	30	-
Packet size	1500	Bytes
Packet arrival rate, λ	100	packets/ms
Transmission power	20	dBm
Thermal Noise density, σ^2	-174	dBm
Minimum SINR, γ_0	15	dB
Delay constraint, $d_{k,max}$	0.6	ms
Packet error rate constraint, ϵ_0	10^{-6}	-
Default learning rate	0.1	-
Discount factor, γ^t	0.9	-

Here we used a static simulator, *MATLAB*^{TM}, the CVX tool for solving geometric programs is used to solve the NP-Hard optimization problem in (5). The evaluation results reported in this section are based on the objective of maximizing the UL capacity of eMBB users per gNB and utility function of the corresponding slice. It is worth noting that the dynamic resource percentage threshold scheme operates without exclusively relying on the learning rate. For both the RL and constrained RL strategies, a traditional default value for the learning rate, $\alpha = 0.1$ was used as a starting point for the problem, then the learning rate was reduced to $\alpha_t = 0.01$. The performance evaluation of these algorithms is in terms of: (i) convergence performance, and (ii) average slice throughput, with a focus on traffic averaging, similar traffic, as well as dissimilar traffic.

- **Traffic Averaging:** This is the most trivial way of evaluating the algorithm, which is by averaging the accumulated reward. Since reward maximization is the core of all RL strategies, averaging makes both traffic types to share the same reward function. The averaging technique, however, does not provide much insight into the different behaviors that the agent may elicit due to cross-talk error.
- **Similar Traffic:** In this case, the system evolves to maximize the throughput of all admitted traffic without concentrating on the access delay. In this way, both traffic types are treated as similar. However, this technique quickly indicates the need to capture task-specific metrics such as throughput and delays from each episode. This comes in the form of splitting tasks into dissimilar traffic.
- **Dissimilar Traffic:** This is a highly intricate problem, where the scheduling problem is done over a time-varying set of devices with heterogeneous traffic contexts. Here, the ability of the algorithm to schedule traffic of different classes of requirements, which is the objective of network slicing, is evaluated. The metrics were calculated as averages over 100 episodes for each environment in order for each algorithm to obtain statistically significant results.

B. EVALUATING CONVERGENCE PERFORMANCE

It must be noted here that since the RL algorithms implemented by Stable-Baselines have varying parallelization capabilities, they are not compared based on their wall-time consumption. Thus, the convergence rates evaluated in this section refers to time consumption in terms of simulation steps per episode an agent takes to make a decision. Therefore, the time efficiency score is a number of iteration steps instead of percentage values. The evaluation begins with the default learning rate, which is then substantially reduced to observe the generalization accuracy of both the RL strategies. The convergence results for averaged traffic, evaluated at learning rates $\alpha_t = 0.1$ are presented in FIGURE 2 below.

In FIGURE 2 above, the dynamic RPT algorithm exhibits a steady increase in the required steps for convergence as the number of users increase. The traditional RL algorithm required less iterations at $K = 5$, outperforming the proposed algorithm at that epoch. However, the proposed algorithm exhibits a steady decrease in the number of required iterations at each epoch as the number of users increased. The convergence results for averaged traffic are evaluated at learning rate $\alpha_t = 0.01$ in FIGURE 3 below.

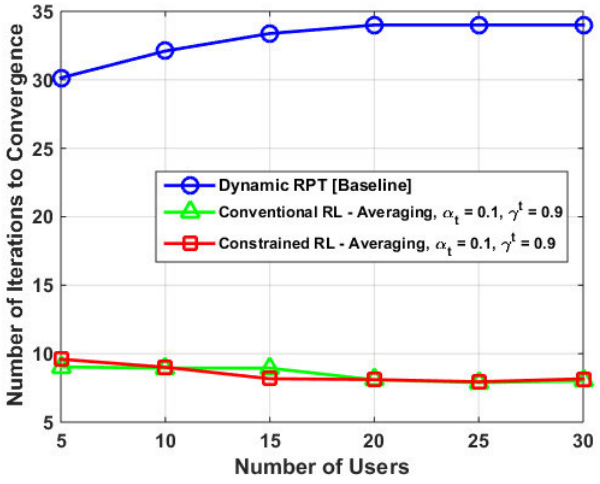

FIGURE 2. Comparison of convergence rates when traffic averaging is used at $\alpha_t = 0.1$.

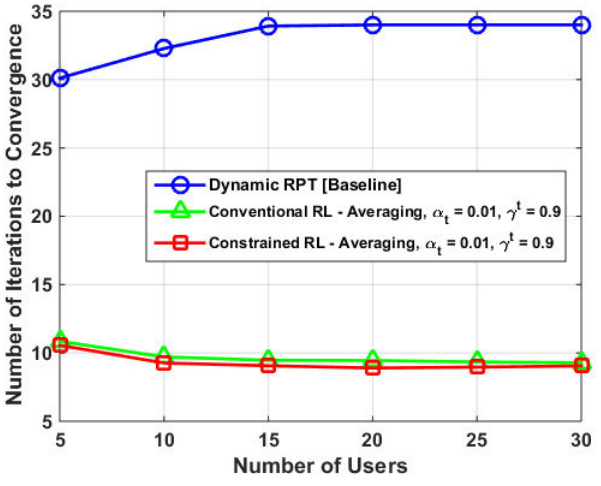

FIGURE 3. Comparison of convergence rates when traffic averaging is used at $\alpha_t = 0.01$.

In FIGURE 3 above, the learning rate was decreased to $\alpha_t = 0.01$ and the algorithms show a steady performance as the number of users increase. Even the baseline algorithm shows a plateau at $15 \leq K \leq 30$. It must be noted that at this point, the Q-learning algorithm has not yet been tasked to separate the traffic received from both network slices, but average it. The performance results when the Q-learning algorithm is learning similar traffic from both slices are shown in FIGURE 4 and FIGURE 5 below.

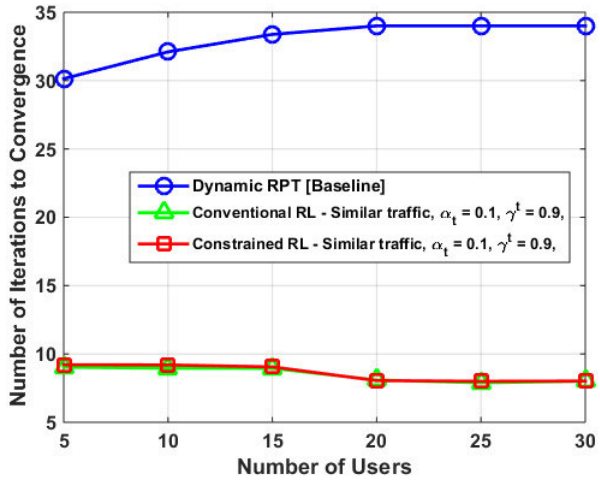

FIGURE 4. Comparison of convergence performance when system is learning similar traffic $\alpha_t = 0.1$.

In FIGURE 4 above, the convergence performance of the traditional RL and the constrained RL is the same throughout the range, constant at $5 \leq K \leq 15$, and tends to exhibit an improvement that matches the traditional RL between $15 \leq K \leq 20$, then becomes constant thereafter. This instability exhibited by these results are due to the correlations present in the sequence of observations, since similar traffic is learned here. The same performance evaluation is repeated, now with a reduced learning rate of $\alpha_t = 0.01$, as shown in FIGURE 5 below.

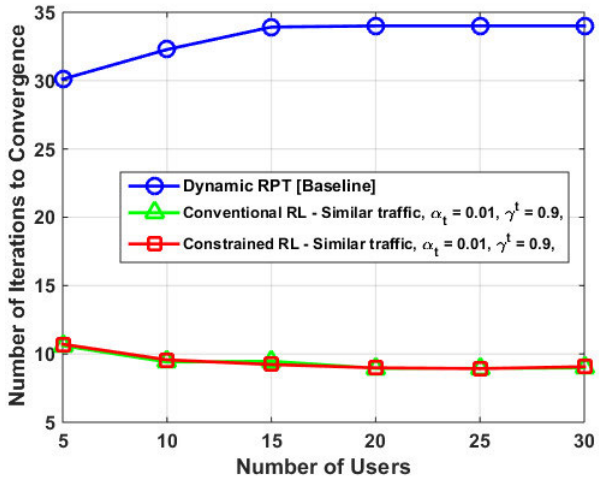

FIGURE 5. Comparison of convergence performance when system is learning similar traffic at $\alpha_t = 0.01$.

The performance consistency brought by the decrease in learning rate from $\alpha_t = 0.1$ to $\alpha_t = 0.01$ can be observed. Also, at a lower learning rate, the instability observed in FIGURE 4 above is addressed. However, as it can be seen, this comes at a cost of increased iterations per epoch. The asymptotic behavior shown by the constrained RL strategies show that the use of a smoothed Bellman operation might have solved the issue of algorithm instability and oscillation. The convergence performance is evaluated using dissimilar traffic in FIGURE 6 below.

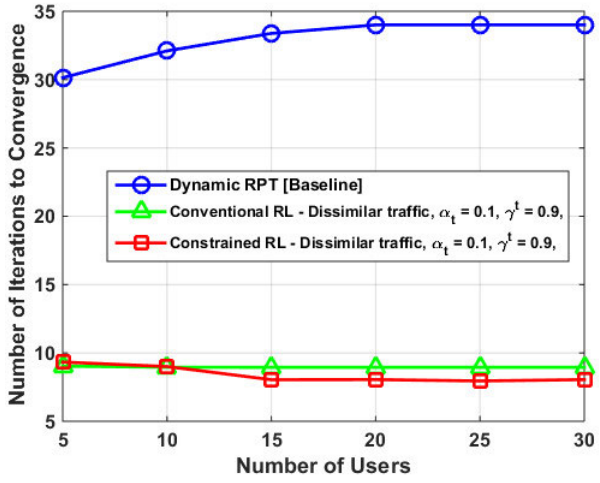

FIGURE 6. Comparison of convergence performance when system is learning dissimilar traffic at $\alpha_t = 0.1$.

The results shown in FIGURE 6 above indicate that at learning rate $\alpha_t = 0.1$ the proposed constrained RL algorithm outperforms the other two baselines with early convergence as the number of users increase. The learning rate is then reduced once again, and the performance is shown in Fig. 7 below.

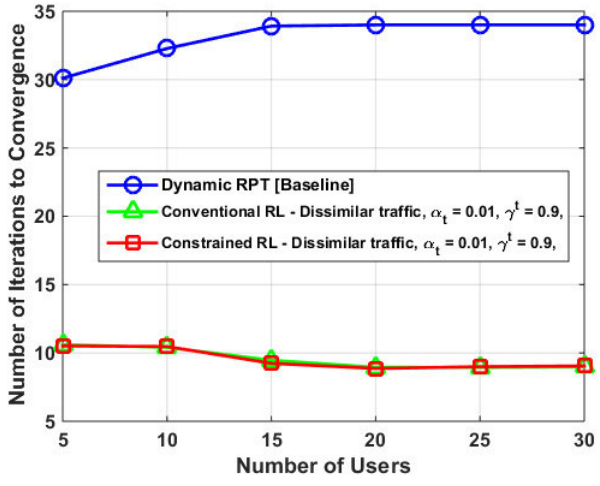

FIGURE 7. Comparison of convergence performance when system is learning dissimilar traffic at $\alpha_t = 0.01$.

The results shown in FIGURE 6 and FIGURE 7 above demonstrate the convergence performance of the algorithms while dealing with dissimilar traffic from the two slices. From the results reported in this section, it was observed that the dynamic RPT algorithm takes ''artificially'' more steps to reach a decision. As mentioned earlier, the convergence of RL strategies is historically unstable given the sparseness of the rewards that are observed from the environment as well as the

difficulty of learning from scratch, i.e., $Q(s, a) = 0$. But the results in this section show that the instability can be handled well using the constrained RL strategy, more especially at lower learning rates. Even though this comes at the cost of more iterations per epoch, the added number of iterations is very small compared with the dynamic RPT algorithm. On overall, the results show the reason why the learning-based approaches, with proper exploration, actually perform better than the traditional methods in terms of time taken to converge based on observed states and action taken. From the results shown above, it can be seen that the proposed algorithm is not very superior to the conventional RL in terms of cycles required to converge. This proves the greediness of the conventional RL strategy can control its rate of convergence, which is a tendency that was avoided by the constrained RL strategy. Greediness is a relevant concern in every optimization algorithm, and was successfully avoided in the propose approach, some positive results to that effect begin to manifest in the following subsection.

C. SLICE THROUGHPUT EVALUATION

In this subsection, a performance comparison of the proposed constrained RL algorithm with two baseline algorithms is done in terms of the average bit rate. The throughput performance is evaluated based on the objective of throughput maximization as a function of an increasing number of admitted users. The performance evaluation using traffic averaging and learning rate, $\alpha_t = 0.1$ is shown in FIGURE 8 below.

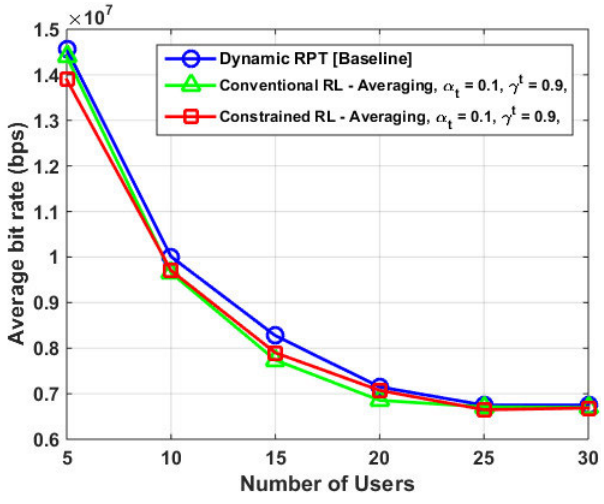

FIGURE 8. Comparison of average bit rate when system is using traffic averaging at $\alpha_t = 0.1$.

In FIGURE 8 above, all the algorithms show a similar trend of a decreasing average bit rate as the number of users increases. However, the proposed constrained RL algorithm seems to be lagging behind in performance compared to the two baseline algorithms until the number of admitted users reaches $K = 10$, where it then shows to outperform the conventional RL, but still lags the dynamic RPT. This shows that even though the dynamic RPT algorithm faces challenges in terms of computational complexity, it is actually a good scheme for task scheduling with resource utilization optimization. However, despite its high computational complexity, the dynamic RPT demonstrates superior performance over the proposed algorithm, which is a behavior that has been reported before in [47]. It can be seen that when $20 \leq K \leq 30$, the performance improvement of the constrained RL scheme matches that of both baselines, which shows that it is able to approach the numerical throughput gain of the dynamic RPT scheme as the number of users increase. The same performance evaluation is conducted with a lower learning rate as shown in FIGURE 9 below.

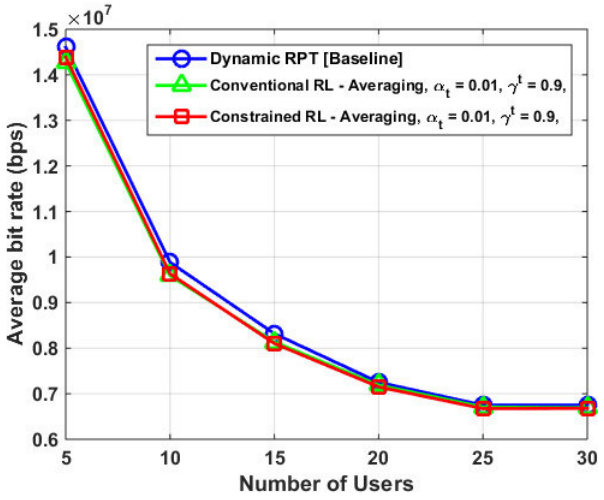

FIGURE 9. Comparison of average bit rate when system is using traffic averaging at $\alpha_t = 0.01$.

In FIGURE 9 above, the learning rate, α_t, was decreased from $\alpha_t = 0.1$ to $\alpha_t = 0.01$, and the proposed constrained RL strategy lags the dynamic RPT algorithm by 0.5%, and outperforms the conventional RL strategy by 0.1%. This shows the sensitivity of the proposed algorithm to a reduction in learning rate as it approaches the performance of the dynamic RPT algorithm better than in FIGURE 8. The reason for this behavior is that at the beginning of the Q-learning algorithm, the value distribution has to demonstrate belief while still working on receiving a better reward. This is because its initialization state did not include some learning history in the form of previous rewards, i.e., $Q(s, a) = 0$. The performance of the algorithms is evaluated for similar traffic with a learning rate of $\alpha_t = 0.1$ on FIGURE 10 below.

The results shown in FIGURE 10 above show that the performance of the proposed constrained RL algorithm is slightly better than the traditional RL algorithm, but still slightly lags the dynamic RPT algorithm. Performance evaluation results for $\alpha_t = 0.01$ are shown in Fig. 11 below:

FIGURE 10 and FIGURE 11 above show results where the task of the agent has been split, but still viewing traffic as similar. The reflection of the constrained RL algorithm here is clearly seen as it better approximates the performance of the dynamic RPT algorithm. It can be seen, however,

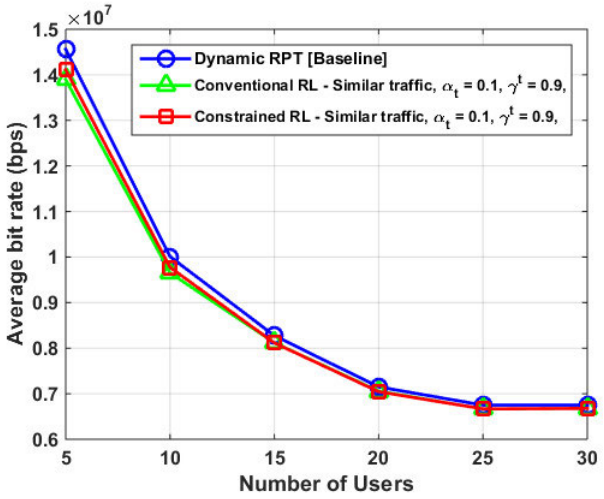

FIGURE 10. Comparison of average bit rate when system is learning similar traffic at $\alpha_t = 0.1$.

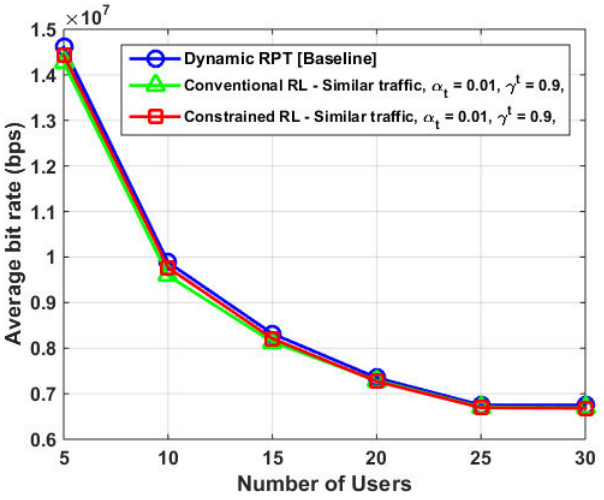

FIGURE 11. Comparison of average bit rate when system is learning similar traffic at $\alpha_t = 0.01$.

in FIGURE 11 above, that the dynamic RPT algorithms outperforms the proposed constrained RL strategy by 0.46%, and the conventional RL strategy by 0.39%. By matching the best existing algorithm up to a factor of horizon dependence, the proposed constrained RL algorithm is showing to be suitable for network slicing problems. This means that researchers working on single network slices can benefit from this technique. However, when working in network slicing, the problems usually require one to deal with at least two slices or three. In that case, one is said to be dealing with the core problem of network slicing. By taking the important advantages of constrained RL strategies one can be able to handle different slices simultaneously. In working with different network slices and treating the traffic as dissimilar is a very intricate, yet exiting, problem. In this case, the expression in (32) is split and each part treated independent of the other. For this task, classification, regression, and decision-making are utilized, and the results for learning rate $\alpha_t = 0.1$ are shown in FIGURE 12 below:

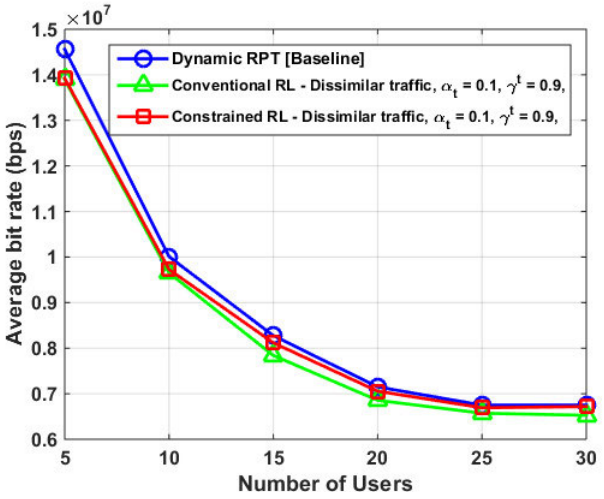

FIGURE 12. Comparison of average bit rate when system is learning dissimilar traffic at $\alpha_t = 0.1$.

The results for learning rate $\alpha_t = 0.01$ are shown in FIGURE 13 below:

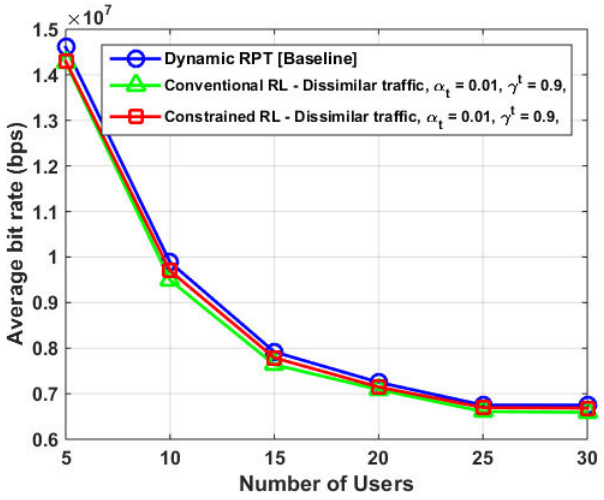

FIGURE 13. Comparison of average bit rate when system is learning dissimilar traffic at $\alpha_t = 0.01$.

The results reported in FIGURE 12 and FIGURE 13 above show the importance of decreasing the learning rate. It must be noted that in this case, the network slicing traffic was treated as dissimilar and the lower learning rate allowed the proposed algorithm to almost match the performance of the dynamic RPT algorithm. The results have shown that at a lower learning rate, there is a significant impact on generalization accuracy of the proposed constrained RL strategy. The results obtained indicate that by starting with the default learning rate, i.e., $\alpha_t = 0.1$, and then reducing it results is a better generalization accuracy. However, this better generalization comes at a cost in the form of the number of iterations

towards convergence at each epoch. However, the cost of an additional few computational iterations can be tolerated since slices are traded over longer intervals. On the basis of these results, it is evident the proposed algorithm, with the help of the SDN-OpenFlow enabled network improves the network efficiency, and performs better when the learning rate, $\alpha_t = 0.01$.

D. LATENCY AND POWER-DELAY EVALUATION

In this section, the proposed algorithm and the baselines are compared in terms of their performance regarding latency.

1) AVERAGE LATENCY VS MEAN ARRIVAL RATE

The evaluation of the average latency as a function of the mean arrival rates at $\alpha_t = 0.1$ is shown in FIGURE 14 below.

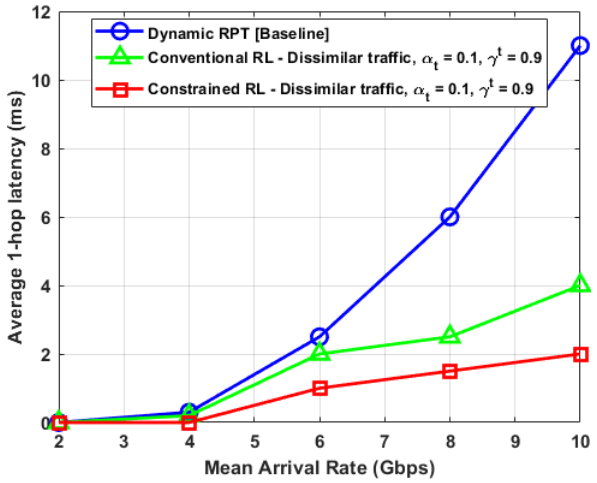

FIGURE 14. Average latency vs mean arrival rates with background traffic of 10 Gbps at $\alpha_t = 0.1$.

In FIGURE 14 above, the average latency increases with the number of connected users, i.e., mean arrival rate. The dynamic RPT algorithm violates the latency constraints, while the both the conventional RL and the proposed algorithm have a much better performance. The reason behind this performance gain is that the delay requirement is satisfied using **C5****, which has a better response when learning-based utility-driven algorithms are used than when the traditional utility-delay trade-off approaches are used. The adaptive intelligence of the constrained RL strategy allows for existing knowledge to either be changed or discarded, while new knowledge is being acquired. For instance, when eMBB devices observe the state space and execute the policy $\pi(s(t), a(t))$, they also communicate the information in **C1**** and **C2**** to the gNB.

It can be seen in FIGURE 15 above, that the proposed algorithm outperforms the conventional RL strategy with a performance difference of 11.7%. the average latency of the proposed algorithm improved by 5.0% from what its performance was at $\alpha_t = 0.1$ in FIGURE 14. The behavior of the

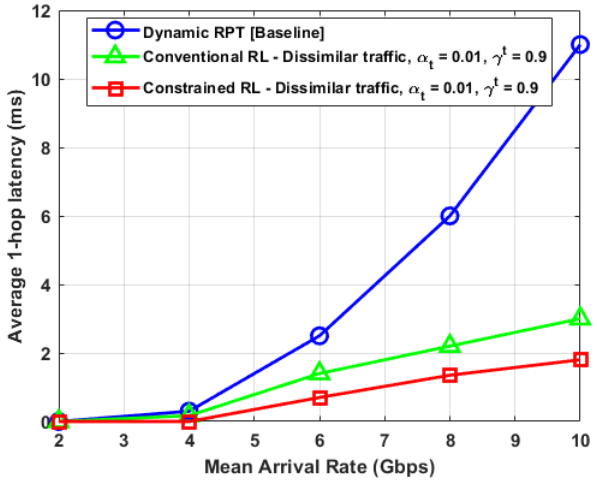

FIGURE 15. Average latency vs mean arrival rates with background traffic of 10 Gbps at $\alpha_t = 0.1$.

target delay level is evaluated as a function of the latency scheme defined by **C5****, as shown in FIGURE 16 below.

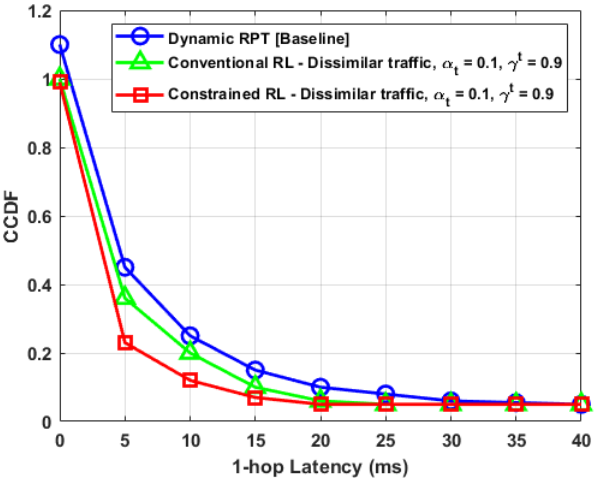

FIGURE 16. Behavior of the tail distribution of latency at $\alpha_t = 0.1$ with background traffic of 10 Gbps.

As shown in FIGURE 16, when $\alpha_t = 0.1$, all the algorithms violate the the latency requirement, but the proposed algorithms gives better performance. However, when α_t is adjusted to 0.01, the proposed algorithm is able to meet the latency requirement, as shown in FIGURE 17 below.

Shown in FIGURE 17 above is the tail distribution of the latency, which shows how the system achieves delays compared to the target delay threshold. As opposed to the average delay, the tail distribution offers useful insights into the URLLC use case. Thus, by imposing the probabilistic latency of $d_{k,max} = 10$ ms on the arrival rate, the violation of the latency constraint becomes easy to trace. From the above results on latency, the inability of dynamic RPT algorithm to adapt to epistemic uncertainties, such as hidden structures, is exposed. Due to this lack of this subsequent ability, it is

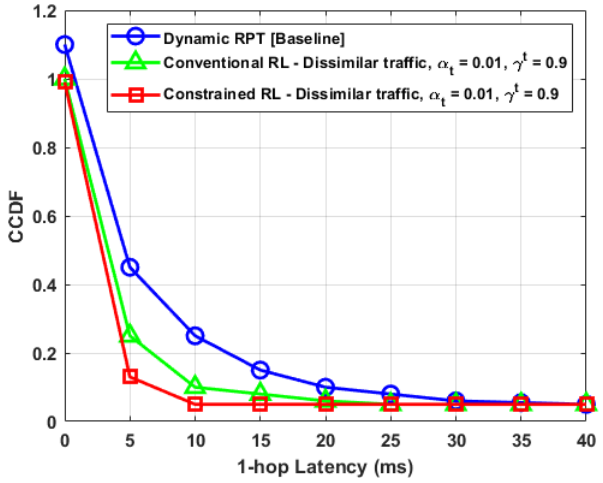

FIGURE 17. Behavior of the tail distribution of latency at $\alpha_t = 0.01$ with background traffic of 10 Gbps.

at a disadvantage in exploring and learning new information from the system.

2) THE POWER-DELAY PERFORMANCE

In this part, the sensitivity of the proposed constrained RL algorithm is examined on non-Markovian system behavior, whose dynamics are very challenging for the conventional RL strategy to adapt to. Here, the power-delay performance is evaluated to observe the time-averaged power consumption per slice, based on the constraints of the stability of the queues. As the physical quantity, d_k, in **C5**** has an upper-bound in practical systems, this evaluation seeks to point out situations when the probabilistic delay levels are shifted below their normal values, thus shifting the system out of the normal Markovian behavior. The power-delay performance of the proposed constrained RL strategy is evaluated for both slice types at $\alpha_t = 0.01$. The constrained RL strategy is further used to evaluate the effects of adjusting the latency thresholds on the different slices on the power consumption of the system. In this evaluation, the learning rate, α_t is fixed at 0.01, only the probabilistic latency is adjusted. The following figures show comparison between eMBB (left y-axis) and URLLC (right y-axis), where each point on the graph corresponds to the average transmission power of that corresponding slice.

In FIGURE 18 above, the latency probabilistic adjustment values for eMBB and URLLC are $\epsilon_0 = 1.5$ and $\epsilon_0 = 0.05$, respectively. Then, the value for eMBB is reduced, while for URLLC is kept constant at $\epsilon_0 = 0.05$, and the performance is shown in FIGURE 19 below.

FIGURE 19 above, the latency values for eMBB and URLLC are $\epsilon_0 = 1.3$ and $\epsilon_0 = 0.05$, respectively. With this adjustment, the average transmission power of the eMBB slice increased tremendously, while a reduced power consumption is seen on the URLLC slice. Then, the latency adjustment is performed for the URLLC slice, keeping the

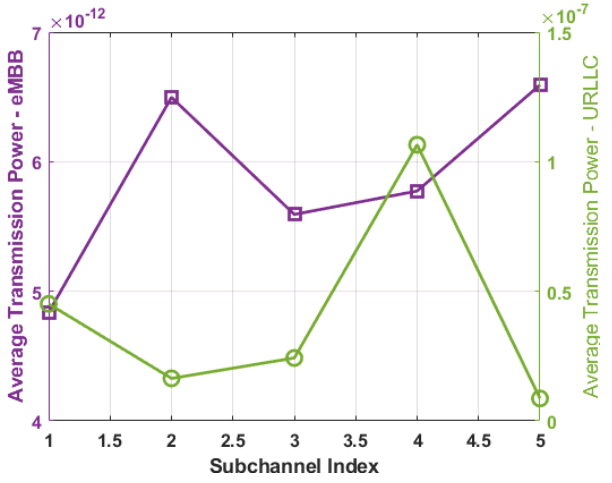

FIGURE 18. Average eMBB/URLLC transmission power (Watts).

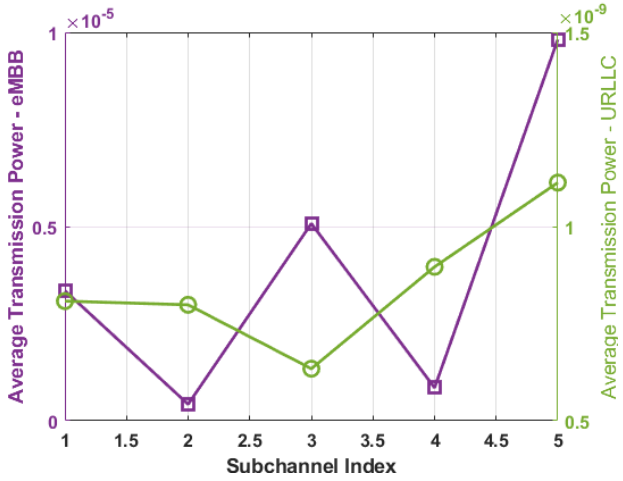

FIGURE 19. Average eMBB/URLLC transmission power (Watts).

eMBB slice at the original value, and the performance results are shown in FIGURE 20 below.

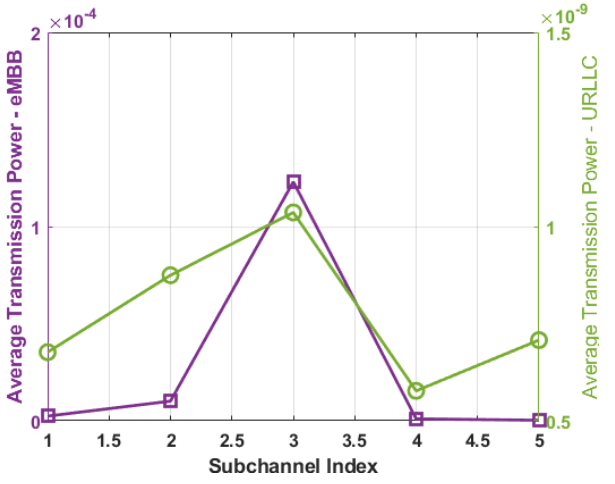

FIGURE 20. Average eMBB/URLLC transmission power (Watts).

FIGURE 20 above, the latency values for eMBB and URLLC are $\epsilon_0 = 1.5$ to $\epsilon_0 = 0.03$, respectively. With the

reduction in the probabilistic latency of the URLLC slice, a corresponding increase in the average transmission power is witnessed. The average power-delay performance shown in the above figures shows that as the probabilistic adjustment of the latency value for a certain slice is reduced, the average transmission power increases corresponding to the sensitivity of that slice to delay requirements. As the reduction of the probabilistic latency values happens on a certain slice, the behavior of that slice becomes increasingly non-Markovian, which causes the system to move to higher power consumption state. These results show that even in rapidly time-varying non-Markovian environments, constrained RL is able to successfully adapt in order to continue satisfying the delay constraints of the applications hosted on that particular slice.

VIII. CONCLUSION AND FUTURE WORK

In this article, the application of dynamic resource percentage threshold in network slicing was presented and compared with the proposed constrained RL strategy. The objective was to improve network slice admission and management by considering the creation and configuration of the two slices with different traffic on demand. Apart from the problem of improving network slice admission and management, the problem presented itself as a non-convex constraint set, with upper bounds on SINR as a function of the transmission power vector in order to reduce the interference to other network users. The proposed constrained RL-based slicing scheme was used to address both the non-convexity of the problem as well as the most pervasive RL problem that leads to an oscillating behavior of Q-learning algorithms. This was done by reformulating the Bellman optimality equation into a primal-dual optimization problem using smoothing and transformation techniques. Using the mathematical techniques from Nesterov and Legendre to perform some approximations and transformations, the constrained RL strategy was obtained, which enabled for the derivation of a policy that ensures that slices are allocated the resources they require.

In validating the feasibility of the proposed constrained RL strategy, the convergence rate and slice throughput in terms of the average bit rate were used. The established complexity upper bounds seem to be the best available - presently - for both sequential and parallel computation and outperform the dynamic RPT algorithm by orders of magnitude. Systematically, the proposed constrained RL algorithm compares favorably with the baseline algorithms at the default learning rate of 0.1, in terms of average bit rate. These complexity upper bounds actually represent significant improvements over even the traditional RL strategy, more especially when the learning rate is decreased from 0.1 to 0.01. The performance evaluation results show that the proposed algorithm can effectively solve network slicing problems with less complexity that the conventional RL strategy. The power-delay evaluation of the proposed algorithm show that it can also adapt well in rapidly time-varying non-Markovian environments and still successfully satisfy the delay constraints of the hosted applications.

A. DISCUSSION OF SUBSTANTIAL IMPACT

In practical network slice management problems, both the dynamics of resource availability (e.g., channel fading) and resource elasticity of the active slices must be considered when allocating resources. The proposed algorithm imbued the dynamic RPT algorithm into the RL strategy to ensure constraint satisfaction, leading to the constrained RL strategy. The advantages of the constrained RL strategy are that: (i) it guarantees the satisfaction of joint constraints with high probability, which is crucial for safety tasks; (ii) it attains the optimal average reward much faster than the conventional RL strategy, which is also justified by the results on convergence. This indicates that the proposed constrained RL algorithm outperforms both dynamic RTP and the conventional RL in terms of the time it takes to converge to the optimal solution; (iii) it has adaptive intelligence that is superior than the conventional RL strategy since it allows for existing knowledge to either be changed or discarded, while new knowledge is being acquired. For instance, when eMBB devices observe the state space and execute the policy $\pi(s(t), a(t))$, they also communicate the information in **C1**** and **C2**** to the gNB.

B. LIMITATIONS AND DISADVANTAGES

However, the proposed algorithm also has several disadvantages in wireless network optimization: (i) **robustness** - performing optimization in a central manner, i.e., with the SDN controller, presents a single isolated point of failure; (ii) **decentralized optimization** - sharing the information collected/stored in the clouds with a centralized controller is not economically efficient due to the time-varying network topology, energy constraints, as well as privacy issues; (i) **convexity/non-convexity** - the way in which the proposed algorithm is handled, which is similar to the heuristic treatment of first justifying the convexity, i.e., achieving $\log - \sum \exp$. This is the same approach used by the dynamic RPT algorithm, which is very efficient, but computationally complex. As a result, of this treatment, the proposed algorithm is seen not to outperform the dynamic RPT in terms of achievable bit rate, but hugely outperforms it in terms of convergence rate. Since with this constrained RL approach, the task of the learning agent becomes slightly different from that of the traditional RL strategy in that it learns the transition dynamics without using the reward information, it is able to address the fundamental limitations of RL.

C. RECOMMENDATIONS FOR FUTURE WORK

The application of this approach needs to be extended to incorporate more slices and even multiple slice objectives. In this way, the envisioned future work comprise of considering the connectivity resources as well as the existence of multiple data centers in complex network slicing models. The current algorithm can be used, either in its current form or as an improved version, to ind the joint optimal power control as well as the dynamic power management policies even when the traffic arrival patterns and statistics of the

propagation channel are unknown. The modeling of such a problem would be structured in a way that action exploration is eliminated in order to enable virtual experience. The aim of this kind of a modeling approach is to dramatically improve the performance of each network slice. The performance of the obtained solution would then be evaluated in terms of computational complexity, scalability, convergence rate, as well as stability.

REFERENCES

[1] E. O'Connell, D. Moore, and T. Newe, "Challenges associated with implementing 5G in manufacturing," *Telecom*, vol. 1, no. 1, pp. 48–67, Jun. 2020.
[2] A. N. Toosi, R. Mahmud, Q. Chi, and R. Buyya, "Management and orchestration of network slices in 5G, fog, edge, and clouds," *Fog Edge Comput., Princ. Paradigms*, vol. 8, pp. 79–96, Jan. 2019.
[3] M. H. Abidi, H. Alkhalefah, K. Moiduddin, M. Alazab, M. K. Mohammed, W. Ameen, and T. R. Gadekallu, "Optimal 5G network slicing using machine learning and deep learning concepts," *Computer Standards Interfaces*, vol. 76, Jun. 2021, Art. no. 103518.
[4] P. Rost, C. Mannweiler, D. S. Michalopoulos, C. Sartori, V. Sciancalepore, N. Sastry, O. Holland, S. Tayade, B. Han, D. Bega, and D. Aziz, "Network slicing to enable scalability and flexibility in 5G mobile networks," *IEEE Commun. Mag.*, vol. 55, no. 5, pp. 9–72, 12 May 2017.
[5] F. Xie, D. Wei, and Z. Wang, "Traffic analysis for 5G network slice based on machine learning," *EURASIP J. Wireless Commun. Netw.*, vol. 2021, no. 1, pp. 1–15, Apr. 2021.
[6] S. Parkvall, Y. Blankenship, R. Blasco, E. Dahlman, G. Fodor, S. Grant, E. Stare, and M. Stattin, "5G NR release 16: Start of the 5G evolution," *IEEE Commun. Standards Mag.*, vol. 4, no. 4, pp. 56–63, Dec. 2020.
[7] I. Bor-Yaliniz, M. Salem, G. Senerath, and H. Yanikomeroglu, "Is 5G ready for drones: A look into contemporary and prospective wireless networks from a standardization perspective," *IEEE Wireless Commun.*, vol. 26, no. 1, pp. 18–27, Feb. 2019.
[8] I. Taboada and H. Shee, "Understanding 5G technology for future supply chain management," *Int. J. Logistics Res. Appl.*, vol. 24, no. 4, pp. 392–406, Jul. 2021.
[9] T.-K. Le, U. Salim, and F. Kaltenberger, "An overview of physical layer design for ultra-reliable low-latency communications in 3GPP releases 15, 16, and 17," *IEEE Access*, vol. 9, pp. 433–444, 2021.
[10] M. Al-Ali, E. Yaacoub, and A. Mohamed, "Dynamic resource allocation of eMBB-uRLLC traffic in 5G new radio," in *Proc. IEEE Int. Conf. Adv. Netw. Telecommun. Syst. (ANTS)*, Dec. 2020, pp. 1–6.
[11] M. A. Siddiqi, H. Yu, and J. Joung, "5G ultra-reliable low-latency communication implementation challenges and operational issues with IoT devices," *Electronics*, vol. 8, no. 9, p. 981, Sep. 2019.
[12] P. L. Vo, M. N. Nguyen, T. A. Le, and N. H. Tran, "Slicing the edge: Resource allocation for RAN network slicing," *IEEE Wireless Commun. Lett.*, vol. 7, no. 6, pp. 970–973, 31 May 2018.
[13] L. Feng, Y. Zi, W. Li, F. Zhou, P. Yu, and M. Kadoch, "Dynamic resource allocation with ran slicing and scheduling for URLLC and eMBB hybrid services," *IEEE Access*, vol. 8, pp. 34538–34551, 2020.
[14] X. Xi, X. Cao, P. Yang, J. Chen, T. Q. Quek, and D. Wu, "Network resource allocation for eMBB payload and URLLC control information communication multiplexing in a multi-UAV relay network," *IEEE Trans. Commun.*, vol. 69, no. 3, pp. 1802–1817, Dec. 2020.
[15] Y. Liu, J. Ding, and X. Liu, "Resource allocation method for network slicing using constrained reinforcement learning," in *Proc. IFIP Netw. Conf.*, Jun. 2021, pp. 1–3.
[16] M. A. Habibi, B. Han, M. Nasimi, and H. D. Schotten, "The structure of service level agreement of slice-based 5G network," 2018, *arXiv:1806.10426*.
[17] Y. Kim, S. Kim, and H. Lim, "Reinforcement learning based resource management for network slicing," *Appl. Sci.*, vol. 9, pp. 1–17, Jun. 2019.
[18] O. Beaumont, L. Eyrauld-Dubois, A. Guermouche, and T. Lambert, "Comparison of static and runtime resource allocation strategies for matrix multiplication," in *Proc. 27th Int. Symp. Comput. Archit. High Perform. Comput. (SBAC-PAD)*, Oct. 2015, pp. 170–177.
[19] S. Miryoosefi and C. Jin, "A simple reward-free approach to constrained reinforcement learning," 2021, *arXiv:2107.05216*.
[20] N. M. Nam, W. Geremew, S. Reynolds, and T. Tran, "Nesterov's smoothing technique and minimizing differences of convex functions for hierarchical clustering," *Optim. Lett.*, vol. 12, no. 3, pp. 455–473, May 2018.
[21] J. Achiam, D. Held, A. Tamar, and P. Abbeel, "Constrained policy optimization," in *Proc. 34th Int. Conf. Mach. Learn. (ICML)*, Aug. 2017, pp. 22–31.
[22] M. G. Bellemare, W. Dabney, and R. Munos, "A distributional perspective on reinforcement learning," in *Proc. Int. Conf. Mach. Learn.*, Jul. 2017, pp. 449–458.
[23] Y. Lucet, "What shape is your conjugate? A survey of computational convex analysis and its applications," *SIAM Rev.*, vol. 52, no. 3, pp. 505–542, 2010.
[24] L. M. M. Zorello, S. Troia, S. Giannotti, R. Alvizu, S. Bregni, and G. Maier, "On the network slicing for enterprise services with hybrid SDN," in *Proc. IEEE Latin-Amer. Conf. Commun. (LATINCOM)*, Nov. 2020, pp. 1–6.
[25] M. A. Habibi, B. Han, F. Z. Yousaf, and H. D. Schotten, "How should network slice instances be provided to multiple use cases of a single vertical industry?" *IEEE Commun. Standards Mag.*, vol. 4, no. 3, pp. 53–61, Sep. 2020.
[26] B. Ramisetti and A. Kumar, "Methods for cellular network's operation in unlicensed mmWave bands," in *Proc. Nat. Conf. Commun. (NCC)*, Feb. 2020, pp. 1–6.
[27] *NR; NR and NG-RAN Overall Description; Stage 2*, document TR 38.300, Version 16.0.0, 3GPP, Dec. 2019.
[28] N. Jaldén, J. Lun, P. Frenger, A. Furuskar, S. Venkatasubramanian, and E. Trojer, "Full coverage with 3GPP technologies on the feasibility of providing full rural cellular coverage," in *Proc. 91st Veh. Technol. Conf.*, May 2020, pp. 1–6.
[29] K. Koutlia, R. Ferrus, E. Coronado, R. Riggio, F. Casadevall, A. Umbert, and J. Pérez-Romero, "Design and experimental validation of a software-defined radio access network testbed with slicing support," *Wireless Commun. Mobile Comput.*, vol. 2019, pp. 1–17, Jun. 2019.
[30] D. Peethala, T. Kaiser, and A. H. Vinck, "Reliability analysis of centralized radio access networks in non-line-of-sight and line-of-sight scenarios," *IEEE Access*, vol. 7, pp. 18311–18318, 2019.
[31] O. Vikhrova, C. Suraci, A. Tropeano, S. Pizzi, K. Samouylov, and G. Araniti, "Enhanced radio access procedure in sliced 5G networks," in *Proc. 11th Int. Congr. Ultra Modern Telecommun. Control Syst. Workshops (ICUMT)*, Oct. 2019, pp. 1–6.
[32] M. C. Hlophe and B. T. Maharaj, "Optimization and learning in energy efficient resource allocation for cognitive radio networks," in *Proc. IEEE 89th Veh. Technol. Conf. (VTC-Spring)*, Apr. 2019, pp. 1–5.
[33] M. Chiang, *Geometric Programming for Communication Systems*. Boston, MA, USA: Now, 2005.
[34] M. C. Hlophe and B. T. Maharaj, "AI meets CRNs: A prospective review on the application of deep architectures in spectrum management," *IEEE Access*, vol. 9, pp. 113954–113996, 2021.
[35] Z. Kotulski, T. W. Nowak, M. Sepczuk, M. Tunia, R. Artych, K. Bocianiak, T. Osko, and J.-P. Wary, "Towards constructive approach to end-to-end slice isolation in 5G networks," *EURASIP J. Inf. Secur.*, vol. 2018, no. 1, pp. 1–23, Dec. 2018.
[36] M. Torabi and D. Haccoun, "Variable-rate adaptive modulation with optimum switching thresholds for cooperative systems with relay selection," *Wireless Commun. Mobile Comput.*, vol. 13, no. 13, pp. 1161–1176, Sep. 2013.
[37] H. Alves, G. Do Jo, J. Shin, C. Yeh, N. H. Mahmood, C. Lima, C. Yoon, N. Rahatheva, O.-S. Park, S. Kim, E. Kim, V. Niemelä, H. W. Lee, A. Pouttu, H. K. Chung, and M. Latva-aho, "Beyond 5G URLLC evolution: New service modes and practical considerations," 2021, *arXiv:2106.11825*.
[38] A. Chagdali, S. E. Elayoubi, and A. M. Masucci, "Slice function placement impact on the performance of URLLC with multi-connectivity," *Computers*, vol. 10, no. 5, p. 67, May 2021.
[39] X. Chen, Y. Tang, M. Zhang, and L. Huang, "Ran slice selection mechanism based on satisfaction degree," in *Proc. IEEE Wireless Commun. Netw. Conf. Workshops (WCNCW)*, Apr. 2020, pp. 1–6.
[40] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *Proc. 35th Int. Conf. Mach. Learn.*, Stockholm, Sweden, vol. 80, Jan. 2018, pp. 1861–1870.
[41] A. G. Barto, "Reinforcement learning: An introduction," *SIAM Rev.*, vol. 63, no. 2, p. 423, Jun. 2021.

[42] N. Shariati, E. Bjornson, M. Bengtsson, and M. Debbah, "Low-complexity polynomial channel estimation in large-scale MIMO with arbitrary statistics," *IEEE J. Sel. Topics Signal Process.*, vol. 8, no. 5, pp. 815–830, Apr. 2014.
[43] S. Koenig and R. G. Simmons, "Complexity analysis of real-time reinforcement learning," in *Proc. AAAI*, 1993, pp. 99–107.
[44] B. Price and C. Boutilier, "Accelerating reinforcement learning through implicit imitation," *J. Artif. Intell. Res.*, vol. 19, pp. 569–629, Dec. 2003.
[45] S. M. Kakade, "On the sample complexity of reinforcement learning," Ph.D. thesis, Gatsby Comput. Neurosci. Unit, Univ. College London, U.K., 2003.
[46] G. Wang, G. Feng, T. Q. S. Quek, S. Qin, R. Wen, and W. Tan, "Reconfiguration in network slicing—Optimizing the profit and performance," *IEEE Trans. Netw. Service Manag.*, vol. 16, no. 2, pp. 591–605, Jun. 2019.
[47] Y. Liu, J. Ding, and X. Liu, "A constrained reinforcement learning based approach for network slicing," in *Proc. IEEE 28th Int. Conf. Netw. Protocols (ICNP)*, Oct. 2020, pp. 1–6.

MDUDUZI C. HLOPHE (Member, IEEE) received the Ph.D. degree (electronic engineering) in wireless communications from the University of Pretoria, South Africa, in 2020. He is currently a Postdoctoral Research Fellow with the Broadband Wireless Multimedia Communications (BWMC), Department of Electrical, Electronic and Computer Engineering, University of Pretoria. His research interests include mathematical modeling of multivariate statistics, classification methods, knowledge discovery, reasoning with uncertainty and inference, and predictive analytics and inference with applications in wireless communications, finance, health, and robotics.

BODHASWAR T. MAHARAJ (Senior Member, IEEE) received the Ph.D. degree (engineering) in wireless communications from the University of Pretoria. He is a Full Professor and the Dean of the Faculty of Engineering, Built Environment and IT, University of Pretoria, where he is the Sentech Chair of the Broadband Wireless Multimedia Communications with the Department of Electrical, Electronic and Computer Engineering. His research interests include OFDM-MIMO systems, massive MIMO, cognitive radio resource allocation, and 5G cognitive radio sensor networks.