Reinforcement Learning for Dynamic Resource Optimization in 5G Radio Access Network Slicing

Yi Shi[1,2], Yalin E. Sagduyu[2], and Tugba Erpek[1,2]

[1]Virginia Tech, Blacksburg, VA 24061, USA
[2]Intelligent Automation, Inc., Rockville, MD 20855, USA

***Abstract*—The paper presents a reinforcement learning solution to dynamic resource allocation for 5G radio access network slicing. Available communication resources (frequency-time blocks and transmit powers) and computational resources (processor usage) are allocated to stochastic arrivals of network slice requests. Each request arrives with priority (weight), throughput, computational resource, and latency (deadline) requirements, and if feasible, it is served with available communication and computational resources allocated over its requested duration. As each decision of resource allocation makes some of the resources temporarily unavailable for future, the myopic solution that can optimize only the current resource allocation becomes ineffective for network slicing. Therefore, a Q-learning solution is presented to maximize the network utility in terms of the total weight of granted network slicing requests over a time horizon subject to communication and computational constraints. Results show that reinforcement learning provides major improvements in the 5G network utility relative to myopic, random, and first come first served solutions. While reinforcement learning sustains scalable performance as the number of served users increases, it can also be effectively used to assign resources to network slices when 5G needs to share the spectrum with incumbent users that may dynamically occupy some of the frequency-time blocks.**

***Index Terms*—5G security, network slicing, radio access network, network optimization, reinforcement learning.**

I. Introduction

The demand for high-rate cellular communications is ever growing with emerging applications such as virtual/augmented reality and Internet of Things. *5G communications* has evolved to meet the increasing user demand with its high throughput and low latency promises. For cellular communications systems prior to 5G, static allocation of the network resources has been considered to support different types of user applications. However, this solution is inefficient and falls short of meeting the desired user quality of experience (QoE) in 5G systems.

5G introduces the *Radio Access Network (RAN) slicing* capability, where the physical network infrastructure is shared among mobile virtual network operators. The static allocation of resources (such as frequency, power, and computational resources) is replaced by reserving them on the fly with *network slicing* based on the dynamic user demand. The use cases to be supported by network slicing are categorized as enhanced Mobile Broadband (eMBB), massive machine-type communications (mMTC), and ultra-reliable low-latency communications (urLLC) based on the throughput and latency requirements. The details on resource allocation as part of RAN slicing are not defined yet in the 3GPP standards. To address this gap, the ongoing research has focused on how to allocate the resources as part of RAN slicing [1]–[4].

A. Related Work

In RAN slicing, the complex network dynamics make the underlying network optimization problem challenging. In [5], both network slicing and mobile edge computing (MEC) technologies were considered and the resource allocation that was formulated to minimize the interference among different mobile virtual network operators was shown to be NP-hard. Recently, *machine learning* was applied to solve the RAN optimization problem as an alternative to the model-based optimization that becomes easily intractable due to the complexity of dynamics involving resources and requests. In [6], it was shown that in-network deep learning is promising for application and device specific identification and traffic classification problems. Deep learning was also used in [7] to manage network load efficiency and network availability.

As data may not be readily available to train complex deep learning structures for resource allocation in response to network slicing requests, a model-free approach such as *reinforcement learning* (RL) has emerged as a practical solution to learn from the 5G network performance and update resource allocation decisions for network slicing. For resource allocation as part of network slicing, RL was compared to the static and round-robin scheduling methods in [8]. Both bandwidth and computational resources were considered in [9]. Resource allocation with RL was compared to heuristic, best-effort, and random methods in [10]. A prototype of network slicing was implemented on an end-to-end mobile network system in [11]. RL was applied in [12] for power-efficient resource allocation in cloud RANs by considering multiple transmitters and receivers at the same frequency (rather than 5G time-frequency blocks). In [13], resource allocation with RL was studied by exploiting prediction on communication requests. RL was also used extensively for resource allocation in wireless applications other than network slicing [14]–[16].

The communication and computation resources available to be allocated, the objectives pursued for requests in terms

This effort is supported by the U.S. Army Research Office under contract W911NF-20-P0035. The content of the information does not necessarily reflect the position or the policy of the U.S. Government, and no official endorsement should be inferred.

978-1-7281-6339-0/20/$31.00 ©2020 IEEE

of latency, throughput (5G rate) and priority, and the states, actions, and rewards used in RL have not been always fully or explicitly specified or addressed in the above works on 5G network slicing. Furthermore, the constraints imposed on resources due to potential spectrum utilization of incumbent signals in the same band have not been considered before.

B. Contributions

In this paper, we present the RL solution based on *Q-learning* algorithm to dynamically allocate resources for 5G network slicing. The motivation is that a myopic solution that only considers resources and demands at a given instant cannot be effective in optimizing resources over a time horizon, as current resource allocation will make some resources unavailable for future, thereby coupling the current and future resource allocation problems. We consider dynamic allocation of the *resource blocks (RBs)*, *transmit power* and *computational resources* to support *downlink* communications from a 5G base station, gNodeB, to user equipments (UEs). Each network slicing request from a UE is associated with priority, throughput, CPU usage and latency (deadline) requirement, and needs to be served for a specific duration. These requests compete for frequency-time blocks, transmit powers of the gNodeB, and computational resources spent to support their applications.

In our Q-learning solution, the states correspond to the available resources that transition over time depending on how they are occupied (for granted network slicing requests) or released (for completed requests). The actions are the assignments of resources to requests. The reward is the network utility measured as the weighted sum of requests satisfied, where weights correspond to relative priorities of these requests.

We show that Q-learning successfully allocates resources over a time horizon and provides major gains in network utility compared to *myopic*, *random* and *first come first served (FCFS)* resource allocation algorithms. As the number of UEs increases or priorities of network slicing request change over time, we show that Q-learning successfully adapts to dynamic user demands.

We further show that Q-learning can be effectively used to dynamically allocate spectrum resources in response to *incumbent users coexisting with 5G users*. Since spectrum is a scarce resource, in an effort to develop better utilization of spectrum and support the increasing data rate requirements for emerging wireless applications, the Federal Communications Commission (FCC) adopted rules for shared commercial use of the 3550-3700 MHz band which is also known as 3.5-GHz *Citizens Broadband Radio Service (CBRS) band* [17]. A three-tiered access and authorization framework is developed to accommodate the shared federal and non-federal use of the band. 5G and radar system (incumbent user) will need to coexist in these bands once 5G systems are deployed. It is required that 5G systems do not interfere with the operation of an incumbent user such as radar.

One approach to achieve this spectrum coexistence goal is to continuously sense the spectrum by the Environmental Sensing Capability (ESC) sensors and turn off the operation once an incumbent signal is detected from the commercial system perspective. However, depending on the radar bandwidth, a naive solution to pause the operation of the 5G network may be infeasible. A better approach would be not allocating the corresponding spectrum by the Spectrum Access System (SAS) to the commercial users of 5G once an incumbent signal is detected at a specific resource block.

It is not clear how to reallocate spectrum resources to network slices in 5G in response to incumbent user dynamics that would change the availability of frequency-time blocks over time. As RL takes the time-horizon into account instead of pursuing a myopic objective, it emerges as a viable solution to reconfigure network resources in order to support spectrum coexistence applications for 5G. We show that the proposed Q-learning solution adapts to spectrum sharing dynamics successfully and allocates resources for 5G RAN slicing effectively in response to dynamic spectrum utilization of incumbent users.

The rest of the paper is organized as follows. Section II describes the resources and the network slicing requests. Section III defines the optimization problem for dynamic resource allocation and presents the Q-learning solution. Section IV provides the performance evaluation results. Section V concludes this paper.

II. Resources and Network Slicing Requests

The system model is shown in Figure 1. Suppose that there is one gNodeB and there are N UEs in a 5G network. Each UE needs to connect to the gNodeB to have network access for its applications. 5G network supports three types of traffic: eMBB, URLLC, and mMTC. Consequently, UEs may request downlink communication service (from the gNodeB to the UEs) with dynamic QoE levels regarding different throughput, CPU usage and latency (deadline) requirements, and different priorities (relative importance).

Requests are handled by the gNodeB and then appropriate network slices with corresponding resource blocks are assigned to requests. If a request is not answered yet, it stays in a queue until its deadline (time limit from the request arrival until the service starts) expires. The objective of such

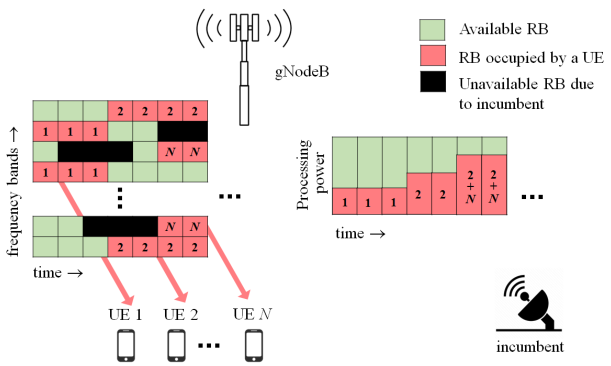

Fig. 1: System model for RAN resource allocation with network slicing.

assignments is to maximize the weighted number of supported requests or the total provided services, where weights represent priorities of these requests. As network resources, we consider bandwidth, communication (transmit) power, and CPU usage. At time t, there are a set of active requests $A(t)$ that includes either requests that have just arrived or requests that are in the waiting list (i.e., requests that are not satisfied yet and their deadlines have not expired). The QoE can be measured in different forms, where we consider throughput, CPU usage, and latency requirements. The CPU usage requirement of UE i for its network slicing request j is

$$P_{ij}^C \geq p_{ij}^C, \quad (i,j) \in A(t), \tag{1}$$

where P_{ij}^C is the assigned computational resource (measured by CPU usage) and p_{ij}^C is the minimum required resource. For throughput, the QoE requirement of UE i for its request j is

$$D_{ij} \geq d_{ij}, \quad (i,j) \in A(t), \tag{2}$$

where D_{ij} is the achieved data rate and d_{ij} is the minimum required rate. D_{ij} is determined by the assigned bandwidth F_{ij}, the assigned transmit power P_{ij}^T (for downlink traffic) at the gNodeB, the modulation coding scheme used for communications between the gNodeB and UE i, and channel effects including interference and path loss. Note that each antenna of the gNodeB serves a different user through spatial multiplexing. For 5G NR, the approximate data rate (bps) for a given number of aggregated carriers in a band or band combination is computed as follows [18]:

$$r = \sum_{k=1}^{K} \left(v_{Layers}^{(k)} Q_m^{(k)} f^{(k)} R_{max} \frac{N_{PRB}^{BW(k),\mu} 12}{T_s^\mu} \left(1 - O^{(k)}\right) \right), \tag{3}$$

where K is the number of aggregated component carriers (CCs) in a band or band combination, and $R_{max} = 948/1024$. For the kth CC, $v_{Layers}^{(k)}$ is the maximum number of supported layers, $Q_m^{(k)}$ is the maximum supported modulation order, $f^{(k)}$ is the scaling factor that can take values 1, 0.8, 0.75 or 0.4, μ is the numerology defined in [19], T_s^μ is the average OFDM symbol duration in a subframe, where $T_s^\mu = \frac{10^{-3}}{14 \cdot 2^\mu}$ for normal cyclic prefix, $N_{PRB}^{BW(k),\mu}$ is the maximum resource block allocation in the UE-supported maximum bandwidth $BW^{(k)}$ in the given band (or combination), and $O^{(k)}$ is the overhead (equal to 0.08 for the uplink in frequency range 1). Assuming a single antenna UE with QPSK modulation, 60 kHz subcarrier spacing and 10 MHz bandwidth, (3) becomes $r = c \cdot K$, where constant c is approximately 12.59×10^6.

The achieved data rate r is further reduced by the corresponding bit error rate (BER) assuming that LDPC coding is used as forward error correction. To calculate the BER at different SNR levels, we simulated the performance of QPSK signal in AWGN channel with LPDC coding (see Figure 2). The BER value is 0 for SNR values higher than -1 dB. The data rate r is scaled based on the BER performance at a given SNR. Thus, when K_{ij} is the number of aggregated CCs and BER_{ij} is the BER of UE i for its request j, (2) becomes

$$c \cdot K_{ij} \cdot (1 - \mathit{BER}_{ij}) \geq d_{ij}, \quad (i,j) \in A(t). \tag{4}$$

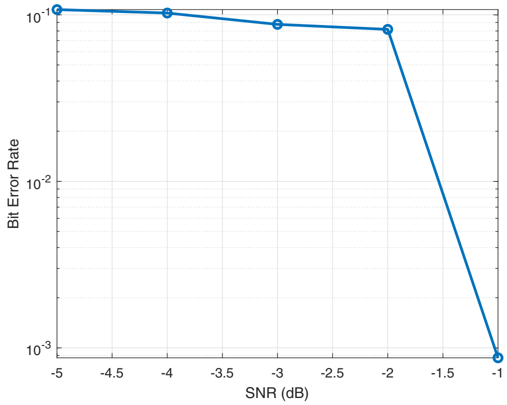

Fig. 2: BER vs. SNR for QPSK signal with LDPC coding in AWGN channel.

III. Optimization Problem for Dynamic Resource Allocation to Network Slices

Denote $x_{ij}(t)$ as the binary indicator on whether UE i's request j is satisfied at time t. The constraints of resource assignments to network slices are given as follows.

$$\sum_{i,j} F_{ij} x_{ij}(t) \leq F(t), \ (i,j) \in A(t), \tag{5}$$

$$\sum_{i,j} P_{ij}^C x_{ij}(t) \leq P^C(t), \ (i,j) \in A(t), \tag{6}$$

$$\sum_{i,j} P_{ij}^T x_{ij}(t) \leq P^T(t), \ (i,j) \in A(t), \tag{7}$$

where $F(t)$, $P^C(t)$, and $P^T(t)$ are available communication (frequency-time blocks), computational and transmit power resources, respectively, at the gNodeB at time t. Some of the gNodeB's resources may have been already assigned to some requests, which are not terminated yet, and thus some resources may not be available.

The myopic objective that optimizes the resource allocation at time t only is to select $F_{ij}(t)$, $P_{ij}^C(t)$ and $P_{ij}^T(t)$ for

$$\max \sum_{ij} w_{ij} x_{ij}(t), \ (i,j) \in A(t) \tag{8}$$

subject to (1), (4)–(7), where w_{ij} is the weight for UE i's request j to reflect its priority. In (8), the myopic objective function is the reward of network slicing allocation at time t.

Next, we consider the optimization problem for a time horizon. The resources are updated from time $t-1$ to time t as follows.

$$F(t) = F(t-1) + F_r(t-1) - F_a(t-1), \tag{9}$$

$$P^C(t) = P^C(t-1) + P_r^C(t-1) - P_a^C(t-1), \tag{10}$$

$$P^T(t) = P^T(t-1) + P_r^T(t-1) - P_a^T(t-1), \tag{11}$$

where $F_r(t-1)$, $P_r^C(t-1)$ and $P_r^T(t-1)$ are released resources at time $t-1$ on frequency, CPU usage, and transmit power, respectively, and $F_a(t)$, $P_a^C(t)$ and $P_a^T(t)$ are allocated resources at time t on frequency, CPU usage and transmit power, respectively. Each request has a lifetime l_{ij} and if it is satisfied at time t (namely, the service starts at time t), this request will end at time $t + l_{ij}$. Denote $R(t)$ as the set of requests ending (completed or expired) at time t. The released and allocated (communications and computational) resources at time t are given by

$$F_r(t) = \sum_{(i,j)\in R(t)} F_{ij}, \quad (12)$$

$$P_r^C(t) = \sum_{(i,j)\in R(t)} P_{ij}^C, \quad (13)$$

$$P_r^T(t) = \sum_{(i,j)\in R(t)} P_{ij}^T, \quad (14)$$

and

$$F_a(t) = \sum_{i,j} F_{ij}, \quad (15)$$

$$P_a^C(t) = \sum_{i,j} P_{ij}^C, \quad (16)$$

$$P_a^T(t) = \sum_{i,j} P_{ij}^T. \quad (17)$$

Then, the optimization problem changes to

$$\max \sum_t \sum_{ij} w_{ij} x_{ij}(t), \;\; (i,j) \in A(t) \quad (18)$$

subject to (1), (4)–(7), (9)–(17). In (18), the objective function is the reward of network slicing allocation over a time horizon. Under an unrealistic assumption that the gNodeB knows all future requests, this problem can be solved offline. Future requests can be predicted by a deep neural network [20]. Then, resources can be allocated using RL [13]. In this paper, we solve this problem by RL without any knowledge on future requests, as we will discuss in Section III-A.

We also extend this formulation to dynamic spectrum sharing of 5G with incumbent legacy communication systems. One example is the CBRS band [17]. The FCC authorized the use of the CBRS band for wireless service provider commercialization. In this dynamic spectrum access setting, radar is the incumbent access user (primary user) and 5G is the priority access user (secondary user). The ESC sensor needs to detect the radar signal with spectrum sensing (potentially using statistical [21] and machine learning [22] techniques). When the radar signal is detected, the SAS needs to reconfigure the 5G resource allocation to assign network slices to frequency bands that are not yet occupied by the radar. In our formulation, the available resources $A(t)$ are updated by possibly removing some frequency blocks temporarily due to radar signal occupancy.

We now consider the case that there are multiple gNodeBs, each with its own transmit power and processing power, which can be allocated independently. However, frequency resource blocks should be jointly allocated due to potential interference relationships. In addition to RL considered per gNodeB, the following pre-processing and post-processing stages need to be performed:

1) *Pre-processing*. A resource block is unavailable if (i) it was already allocated to some earlier requests and (ii) it was used by neighboring gNodeBs. Each gNodeB needs to sense the spectrum and identifies the resource blocks that are used by neighboring gNodeBs. Alternatively, a gNodeB can also obtain such information in the post-processing stage.
2) *Post-processing*. Neighboring gNodeBs may plan to assign the same resource blocks in their RL algorithms. To avoid interference among these assignments, a central controller can be used to resolve such conflicts (by removing some assignments) and to broadcast remaining assignments. A distributed scheme is also possible by letting each gNodeB locally broadcast its planned assignments and comparing with planned assignments from neighboring gNodeBs to decide whether a planned assignment should be kept or not.

Once the available frequency-time blocks are determined and the unavailable ones are marked (as done in the spectrum sharing case), each gNodeB runs its RL algorithms as before to allocate its resources.

A. Reinforcement Learning Algorithm

We use Q-learning as the model-free RL algorithm to learn the policy that determines which action (resource assignment) to take under a given state (available resources and requests) for the gNodeB. The gNodeB applies Q-learning to compute the function $Q : S \times A \to \mathbb{R}$ to evaluate the quality of action A producing reward R at state S. Note that the gNodeB maintains Q as the Q-table. At each time t, the gNodeB selects an action a_t, observes a reward r_t, and transitions from the current state s_t to a new state s_{t+1} (this transition depends on current state s_t and action a_t), and updates Q.

Starting Q as a random matrix and using the weighted average of the old value and the new information, Q-learning performs the value iteration update for Q as follows:

$$\begin{aligned} Q(s_t, a_t) \leftarrow\; & Q(s_t, a_t) \quad (19)\\ & + \alpha \cdot \left(r_t + \gamma \cdot \max_a Q(s_{t+1}, a) - Q(s_t, a_t) \right), \end{aligned}$$

where α is the learning rate ($0 < \alpha \le 1$) and γ is the discount factor ($0 \le \gamma \le 1$) for rewards over time. In (19), $\max_a Q(s_{t+1}, a)$ refers to the estimate of the optimal future value of Q.

In dynamic resource allocation to network slices, the reward at time t is w_{ij} if UE i's request j is satisfied at time t, i.e., $x_{ij}(t) = 1$. An action of the gNodeB at time t corresponds to the assignment of resources to a request at time t. Note that multiple actions can be taken at the same time instance. The states at t are $F(t)$, $P^C(t)$ and $P^T(t)$, namely the available resources at the gNodeB at time t. The transition of the state at time t is driven by blocking resources for requests that are granted at time t and releasing resources after the lifetimes of active services expire at time t. In particular, state transitions are given by (9)-(17).

B. *Baselines*

For comparison, we consider three baseline algorithms: random, first come first serve (FCFS), and myopic algorithms.

1) *Random algorithm*: Available resources are allocated to uniformly randomly selected network slice requests.
2) *FCFS algorithm*: Available resources are allocated to network slice requests based on the arrival times of requests, i.e., at any given time, the oldest network slice request is answered first provided that the available resources are sufficient to grant this request.
3) *Myopic algorithm*: Available resources are allocated to maximize the current utility only by solving the optimization problem (8).

IV. Performance Evaluation

Suppose the gNodeB receives requests possibly from three UEs. Later, we extend this setting to more UEs. For each UE, requests arrive with rate of 0.5 per slot. Here, a slot corresponds to each time block that is 0.23 ms long with 60 kHz subcarrier spacing. CPU usage increases by 2% increments (with total of 50 levels). For each request, weight is a random integer in [1,5], lifetime is assigned randomly in [1,10] slots, and deadline is assigned randomly in [1,20] slots. Transmit power can be at 5 levels and the maximum received SNR is selected randomly in [1.5,3]. The total frequency is 10 MHz and is split into 11 bands. The same scenario over 1000 time slots is run to test different algorithms, namely Q-learning, random, FCFS, and myopic algorithms. For Q-learning, we set discount factor as $\gamma = 0.95$ and learning rate as $\alpha = 0.1$. The simulation code is implemented in Python.

Performance comparison of Q-learning and baseline algorithms is given in Table I that shows the network utility achieved by each algorithm and the ratio of network utility improvement by Q-learning relative to other algorithms. Results show that Q-learning achieves much larger utility (total weights of satisfied requests) than other algorithms. On the other hand, myopic algorithm chooses the current best decision without consideration on future and performs better than FCFS and random resource allocations but falls far behind Q-learning. Note that all algorithms (including Q-learning and myopic algorithms) are run multiple times. The range of achieved utility is measured as $[1731, 1831]$ for Q-learning and $[1359, 1466]$ for myopic algorithm. These results indicate that Q-learning achieves better utility than myopic algorithm for the entire range of results.

Next, we increase the number of UEs that send network slicing requests to the gNodeB and change the arrival rate at UEs. Figure 3 shows how the network utility scales with the number of UEs when Q-learning is used. Note that the increase of network utility with the increasing number of UEs declines as the demand for the fixed resources grows.

TABLE I: Performance comparison of Q-learning and baseline algorithms.

Algorithm	Network utility	Ratio of improvement
Q-learning	1807	–
Myopic	1456	24.11%
FCFS	1416	27.61%
Random	1334	35.46%

TABLE II: The impact of weight on the number of served requests.

Weight of UEs	number of served requests
[1,1,1]	[173, 139, 169]
[1,3,5]	[81, 117, 278]
[5,3,1]	[274, 129, 68]

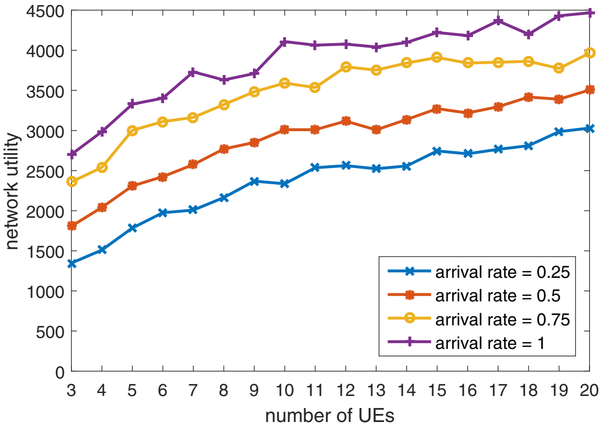

Fig. 3: Network utility vs. number of UEs.

In optimization problem (18), weight w_{ij} assigns priority to request j of UE i. Next, we evaluate the effect of weights on optimization results. For that purpose, we fix the weight of all requests of one UE. The first case is [1,1,1], i.e., all requests have a weight 1. The second case is [1,3,5], i.e., all UE 1's requests have weight 1, all UE 2's requests have weight 3, and all UE 3's requests have weight 5. The third case is [5,3,1]. i.e., all UE 1's requests have weight 5, all UE 2's requests have weight 3, and all UE 3's requests have weight 1. We show the number of served requests for each UE in Table II. Results indicate that if a UE's weight is increased and it is larger than others, the number of served requests for this UE increases relative other UEs. On the other hand, the actual values of weights do not play as big a role as the order of weights in terms of the effect on the network utility, i.e., weights [1,2,3] achieve the same result as weights [1,3,5].

Next, we consider the 5G-radar spectrum coexistence scenario, where the radar signals possibly occupy multiple frequency blocks over time. We consider two types of arrival patterns for radar signals. The first pattern corresponds to independent identically distributed (i.i.d.) arrivals and radar signal appears at any time (slot) with probability p_I. The second pattern corresponds to session (bursty) arrivals and the lifetime of sessions is selected uniformly randomly from [10,50] slots. Then, the arrival of sessions is adjusted to obtain p_I as the probability of incumbent occupancy. We assume that the radar signal is reliably detected. In [22], it was shown that the potential error in detecting radar signals becomes low when deep learning is applied.

In Figure 4, we show the network utility achieved by Q-learning when we vary p_I for both (i.i.d. and session) arrival types of incumbent user. Q-learning adapts to the spectrum occupancy pattern of the incumbent user successfully and is effective in utilizing network resources that are left from the incumbent user's spectrum occupancy. As expected, the network utility achieved by Q-learning drops as p_I increases but this drop is sub-linear with p_I.

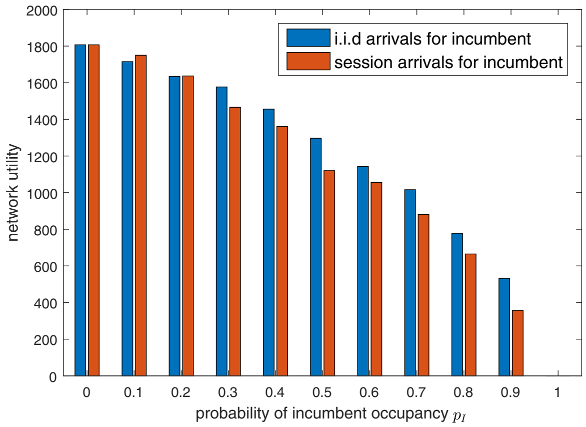

Fig. 4: Network utility when 5G shares the spectrum with an incumbent user.

V. Conclusion

In this paper, we studied the problem of dynamic resource allocation for network slicing in 5G RAN and presented the RL solution that solves the underlying optimization problem over a time horizon. We considered both communication (frequency-time blocks and transmit powers) and computational resources (CPU usage) that are assigned to the incoming network slicing requests for downlink communications from the gNodeB to UEs. Each request comes with latency, rate, and CPU usage requirements. We constructed a Q-learning solution where actions are assigning resources to network slice requests and rewards are measured by the network utility, namely the weighted sum of satisfied network slicing requests (where weights account for priorities of these requests). The underlying states that transition over time correspond to the available resources that become temporarily blocked for either granted network slices or due to the spectrum use of incumbent users. As sessions of network slices are completed, their assigned resources are made available for use by other network slices. We considered random, FCFS and myopic resource allocation algorithms as baselines and showed that Q-learning outperforms them significantly in terms of the network utility. We showed that Q-learning scales well with the increasing number of UEs. We also showed that Q-learning can effectively allocate resources to network slices when some resources become unavailable randomly due to sharing the spectrum opportunistically with incumbent users.

References

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

[2] J. Ordonez-Lucena, P. Ameigeiras, D. Lopez, J. J. Ramos-Munoz, J. Lorca, and J. Folgueira, "Network slicing for 5G with SDN/NFV: Concepts, architectures, and challenges," *IEEE Communications Magazine*, vol. 55, no. 5, pp. 80–87, May 2017.

[3] P. Rost, C. Mannweiler, D. S. Michalopoulos, C. Sartori, V. Sciancalepore, N. Sastry, O. Holland, S. Tayade, B. Han, D. Bega, D. Aziz, and H. Bakker, "Network slicing to enable scalability and flexibility in 5G mobile networks," *IEEE Communications Magazine*, vol. 55, no. 5, pp. 72–79, May 2017.

[4] A. Kaloxylos, "A survey and an analysis of network slicing in 5G networks," *IEEE Communications Standards Magazine*, vol. 2, no. 1, April 2018.

[5] S. D'Oro, F. Restuccia, A. Talamonti, and T. Melodia, "The slice is served: Enforcing radio access network slicing in virtualized 5G systems," *IEEE INFOCOM*, May 2019.

[6] A. Nakao, and P. Du, "Toward in-network deep machine learning for identifying mobile applications and enabling application specific network slicing," *IEICE Transactions on Communications*, July 2018.

[7] A. Thantharate, R. Paropkari, V. Walunj, C. Beard, "DeepSlice: A deep learning approach towards an efficient and reliable network slicing in 5G networks," *IEEE 10th Annual Ubiquitous Computing, Electronics and Mobile Communication Conference (UEMCON)*, Oct. 2019.

[8] R. Li, Z. Zhao, Q. Sun, I. Chih-Lin, C. Yang, X. Chen, M. Zhao, and H. Zhang, "Deep reinforcement learning for resource management in network slicing," *IEEE Access*, vol. 6, pp. 74429–74441, Nov. 2018.

[9] J. Koo, M. R. Rahman, V. B. Mendiratta, and A. Walid, "Deep reinforcement learning for network slicing with heterogeneous resource requirements and time varying traffic dynamics," *IEEE International Conference on Network and Service Management (CNSM)*, Oct. 2019.

[10] H. Wang, Y. Wu, G. Mina, J. Xu, and P. Tang, "Data-driven dynamic resource scheduling for network slicing: A deep reinforcement learning approach," *Information Sciences*, Sept. 2019.

[11] Q. Liu and T. Han, "When network slicing meets deep reinforcement learning," *International Conference on Emerging Networking Experiments and Technologies*, Dec. 2019.

[12] Z. Xu, Y. Wang, J. Tang, J. Wang, and M. C. Gursoy, "A deep reinforcement learning based framework for power-efficient resource allocation in cloud RANs," *IEEE International Conference on Communications (ICC)*, May 2017.

[13] J. A. Ayala-Romero, A. Garcia-Saavedra, M. Gramaglia, X. Costa-Perez, A. Banchs, and J. J. Alcaraz, "vrAIn: A deep learning approach tailoring computing and radio resources in virtualized RANs," *International Conference on Mobile Computing and Networking*, Oct. 2019.

[14] N. C. Luong, D. T. Hoang, S. Gong, D. Niyato, P. Wang, Y. C. Liang YC, and D. I. Kim, "Applications of deep reinforcement learning in communications and networking: A survey," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 4, pp. 3133–3174, May 2019.

[15] H. Ye, G. Y. Li, and B. H. F. Juang, "Deep reinforcement learning based resource allocation for V2V communications," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 4. pp. 3163–3173, Feb. 2019.

[16] N. Abu Zainab, T. Erpek, K. Davaslioglu, Y. E. Sagduyu, Y. Shi, S. Mackey, M. Patel, F. Panettieri, M. Qureshi, V. Isler, and A. Yener, "QoS and jamming-aware wireless networking using deep reinforcement learning," *IEEE Military Communications Conference (MILCOM)*, Nov. 2019.

[17] "Citizens Broadband Radio Service", Code of Federal Regulations. Title 47, Part 96, June 2015.

[18] 3GPP TS 38.306: "NR; User Equipment (UE) radio access capabilities".

[19] 3GPP TS 38.211: "NR; Physical channels and modulation".

[20] D. Bega, M. Gramaglia, M. Fiore, A. Banchs, and X. Costa-Perez, "DeepCog: Cognitive network management in sliced 5G networks with deep learning," *IEEE INFOCOM*, May 2019.

[21] R. Caromi, M. Souryal, and W. B. Yang, "Detection of incumbent radar in the 3.5 GHz CBRS band," *IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, Nov. 2018.

[22] W. M. Lees, A. Wunderlich, P. J. Jeavons, P. D. Hale, and M. R. Souryal, ”Deep learning classification of 3.5 GHz band spectrograms with applications to spectrum sensing," *IEEE Transactions on Cognitive Communications and Networking*, vol. 5, no. 2, pp. 224–236, June 2019.