H. K. Janjua *et al*., "Efficient Optimization of Actor-Critic Learning for Constrained Resource Orchestration in RAN with Network Slicing," *2023 26th Conference on Innovation in Clouds, Internet and Networks and Workshops (ICIN)*, Paris, France, 2023, pp. 100-104, https://doi.org/10.1109/ICIN56760.2023.10073489

Efficient Optimization of Actor-Critic Learning for Constrained Resource Orchestration in RAN with Network Slicing

Hafiza Kanwal Janjua
Universidad de Valladolid
Valladolid, Spain
0000-0002-6655-6015

Ignacio de Miguel
Universidad de Valladolid
Valladolid, Spain
0000-0002-1084-1159

Ramón J. Durán Barroso
Universidad de Valladolid
Valladolid, Spain
0000-0003-1423-1646

Óscar González de Dios
Telefónica I+D
Madrid, Spain
0000-0002-1898-0807

Juan Carlos Aguado
Universidad de Valladolid
Valladolid, Spain
0000-0002-2495-0313

Noemí Merayo Álvarez
Universidad de Valladolid
Valladolid, Spain
0000-0002-6920-0778

Patricia Fernández
Universidad de Valladolid
Valladolid, Spain
0000-0001-5520-0948

Rubén M. Lorenzo
Universidad de Valladolid
Valladolid, Spain
0000-0001-8729-3085

***Abstract*—Network Slicing (NS) is a key enabler of the 5G network ecosystem due to its potential to provide distinct services over the same physical infrastructure. However, the necessity to optimally orchestrate resources for heterogeneous demands is crucial when dealing with resource constraints and Quality-of-Service (QoS) requirements. We consider a radio access network scenario providing NS over multiple base stations (BS) with limited resources, and we design an efficient resource orchestration technique, based on reinforcement learning, which optimizes resource utilization among different services while satisfying the constraints and complying with Service Level Agreement (SLA) and QoS requirements. The proposed technique makes use of the Trust Region Method to formulate the orchestration objective function and satisfy the constraints and is then optimized via Kronecker Factored Approximate Curvature (K-FAC). Extensive simulations demonstrate that the proposed technique outperforms other Reinforcement Learning (RL) algorithms, reaching 99% of QoS and SLA satisfaction while assuring bandwidth constraints.**

***Index Terms*—Network Slicing, Resource Orchestration, Reinforcement Learning, Constrained Optimization.**

I. Introduction

In 2020, around 190 million 5G consumers were estimated to be associated with wide-ranging services such as autonomous driving, industry 4.0, virtual and augmented reality, and the Internet of Things (IoT) [1]. However, such growing demands of data traffic and heterogeneity require an efficient utilization and orchestration of resources to meet the needs of 5G users in terms of Quality of Service (QoS) and Service Level Agreements (SLA) in a cost-efficient way.

This work is part of the IoTalentum project, which has received funding from the EU H2020 research and innovation programme under the MSCA grant agreement No 953442. It is also supported by Consejería de Educación de la Junta de Castilla y León and the European Regional Development Fund (Grant VA231P20), and the Spanish Ministry of Science of Innovation and the State Research Agency (Grant PID2020-112675RB-C42 funded by MCIN/AEI/10.13039/501100011033).

In this work, we propose an intelligent NS resource orchestration technique to comply with pre-defined QoS and SLA requirements while considering bandwidth resource constraints. This problem is also considered in [2], [3], but those studies ignore two crucial features, the consideration of (1) a distributed Radio Access Network (RAN) scenario consisting of multiple BSs with resource constraints, and (2) the continuous dynamicity and fluctuations of network traffic due to the mobility of users. In contrast with [2], [3], our proposal addresses these two issues. We use the Actor-Critic-based RL method and the K-FAC second-order optimizer for that aim. This makes it possible to reinforce the constraints in orchestration decisions during learning and to manage the huge and dynamic network state fluctuations due to user mobility.

The organization of this paper is as follows. Section II describes the problem statement, including the network model and associated constraints. Then, Section III describes the proposed reinforcement learning technique. Finally, the simulations to assess the performance of the technique, and the results obtained, are described in Section IV.

II. Problem Statement

Let us consider a Cloud-Radio Access Network (C-RAN), composed of a set of BSs, and administrated by a centralized main network orchestrator (MNO). Three basic NS services are offered, which are Enhanced Mobile Broadband (eMBB), Ultra-Low Latency Communication (URLLC), and Voice over Long-Term Evolution (VoLTE). Users of the network may subscribe to these services, and they are assumed to move around the network at a specific speed.

The bandwidth is considered a predetermined limited resource and thus is a crucial impacting factor on the orchestration decisions. Thus, the objective is to dynamically distribute bandwidth in each time slot among the three service slices according to their QoS and SLA requirements. We consider

two constraints during allocation management, which are cumulative and instantaneous constraints. Cumulative constraints (total bandwidth) are updated at each scheduling interval, and it must be ensured that the total aggregated allocation of the resource does not exceed an imposed limit. In contrast, instantaneous constraints (data rate and latency) are associated with every allocation action to ensure they do not exceed the limit during that scheduling interval. Moreover, we consider two Key Performance Indicators (KPIs) based on users' experience which are Service Satisfaction Ratio (SSR) and Service Dissatisfaction Ratio (SDR), which will be formally described in Section IV.

III. Proposed Technique: Efficient Actor-Critic based Kronecker-Factored Optimized Trust Region Method

In order to solve the problem described in Section II, a multi-agent RL technique is used. Each BS in the network has a learning agent to ascertain the resource allocation governed by the MNO. Likewise, all the BSs send their current states (dynamic traffic demands) to the main orchestrator in each time slot and receive the responses in terms of actions (i.e., how to distribute bandwidth) based on the MNO global policy. Thenceforth, the feedback on the impact of these actions, either positive or negative, is submitted back to the MNO in each slot in terms of rewards. The MNO utilizes these rewards to tune the orchestration policy to an optimal state. All the BSs work in an independent cooperative manner, which means that the decision of one BS has an impact on the global policy of the MNO, which in the long run modifies the actions sent by the MNO to the BSs.

In particular, we have designed an actor-critic architecture having trust region-based objective function to deploy efficient resource orchestration decisions dynamically while considering the limited system resources as constraints. Moreover, in order to achieve stable gradient updates when learning the policy and dealing with large-scale distributed and heterogeneous traffic, we have integrated a second-order optimizer to update the value function (critic), which is K-FAC. K-FAC is much faster and more efficient when compared to first-order optimization [4], especially when dealing with a multitude of environmental parameters.

The resource orchestration problem is defined as an optimization problem reflecting the Markov property [5]. To formulate the problem as a Markov problem for an actor-critic-based RL environment, we utilized the following key elements of Constrained Markov Decision Processes (CMDP). Let *S* denote the system state, which provides information on network traffic such as the number of users and traffic demand in the current time slot. The action space *A* is described as a set of orchestration decisions (bandwidth distribution) in the current time slot. Principally, *A* is dependent on the cumulative and instantaneous system constraints (as they must be satisfied in each orchestration decision). Each reward *R* is calculated in each time slot when an action is taken based on the orchestration decisions. If the action satisfies all the requirements and constraints, then a higher positive reward is given; otherwise, it is negative. Also, both cumulative and instantaneous rewards are stored in terms of total system SLA (rate and latency) satisfaction. The reward is thus a real number that depends on the current state, the action taken, and the following state,

$$R : S \times A \times S \mapsto \mathbb{R}. \tag{1}$$

Finally, a cost function is used to model the impact of bandwidth constraints. It becomes challenging to handle all of the requests in each time slot. So, facing dissatisfaction with SLA incurs a cost to the system which is calculated as a cumulative constraint C_c,

$$C_c : S \times A \times S \mapsto \mathbb{R}. \tag{2}$$

A. Objective Function with Cumulative Constraint

Our aim is to get an optimal orchestration policy π, i.e., determining which action to take given a network state, while satisfying the constraints. Such a policy depends on a set of parameters θ, thus it will be denoted by π_θ. The objective is to find the parameters θ which maximize the discounted cumulative reward function $\mathcal{L}_{C_c^{\pi_\theta}}$, defined in equation 3,

$$\mathcal{L}_{C_c^{\pi_\theta}} = \mathbb{E}_{\pi_\theta}\left[\sum_{t=0}^{\infty} \gamma^t R\left(s_t, a_t, s_{t+1}\right)\right] \tag{3}$$

where R is the reward and $\mathbb{E}_{\pi_\theta}[\cdot]$ is the trust prediction generated when using policy π_θ. To find the optimal set of parameters we use the policy gradient method [6], which aims to maximize the objective function $\mathcal{L}_{C_c^{\pi_\theta}}$ by iteratively updating policy parameters $\theta \longleftarrow \theta + \alpha \nabla_\theta \mathcal{L}_{C_c^{\pi_\theta}}$, being α the step size of the update. The policy gradient for objective function can be expressed as in equation 4.

$$\nabla_\theta \mathcal{L}_{C_c^{\pi_\theta}} = \mathbb{E}_\pi\left[\sum_{t=0}^{\infty} \nabla_\theta \pi_\theta\left(a_t \mid s_t\right) A^{\pi_\theta}\left(s_t, a_t\right)\right] \tag{4}$$

where $A^{\pi_\theta}\left(s_t, a_t\right)$ is the advantage function, in short $A(t)$,

$$A(t) = \sum_{i=0}^{k-1} \left(\gamma^i R\left(s_{t+i}, a_{t+i}\right) + \gamma^k V_\phi^\pi\left(s_{t+k}\right)\right) - V_\phi^\pi\left(s_t\right) \tag{5}$$

where V_ϕ^π is value function that approximates the policy π, and thus ϕ are the parameters of the critic.

Furthermore, to satisfy the constraint in each orchestration decision we make use of the trust region surrogate function, $r_p(\theta)$, given in equation 6, where $\pi_{\theta\text{old}}$ is the policy of last time slot.

$$r_p(\theta) = \mathbb{E}_\pi\left[\frac{\pi_\theta\left(a_t \mid s_t\right)}{\pi_{\theta_{\text{old}}}\left(a_t \mid s_t\right)} A(t)\right] \tag{6}$$

Moreover, we use the Kullback-Leibler (KL)-Divergence [7] to formulate the expectation trajectories of divergence of policies $\mathbb{E}_\pi\left[D_{\text{KL}}\left(\pi_{\theta_{\text{old}}}\left(a_t \mid s_t\right), \pi_\theta\left(\cdot \mid s_t\right)\right)\right] \leq \delta$, where $\pi_{\theta\text{old}}$ is the policy of last time slot, δ is step size limitation, and D_{KL} is

TABLE I: RAN Network Model Specifications

Model Specifications	eMBB	VoLTE	URLLC
Initial no.of Users	Poisson [Mean=800]	Poisson [Mean=1200]	Poisson [Mean=400]
Inter-Arrival Time per Subscriber	Truncated Pareto [Exponential Para: 1.2] Mean: 6ms, Max:12.5ms	Uniform [Min: 0, Max: 160 ms]	Exponential mean [180 ms]
Distribution of Packet Size	Truncated Pareto [Exponential Para: 1.2] Mean: 100 bytes, Max: 250 bytes]	Constant [40 bytes]	Constant [0.3 Mbyte]
Speed of Users	Uniform [Min: 1m/s, Max: 5m/s]	Uniform [Min: 1 m/s, Max: 5 m/s]	Uniform [Min: 6 m/s, Max: 10 m/s]
SLA (Rate)	100 Mbps	51 kbps	10 Mbps
SLA (Latency)	10 ms	10 ms	1 ms
QoS (Service Availability)	>99%	>99%	>95%

the KL-Divergence. With this, we can reinforce the constraints into the objective function of the orchestration decision policy.

Finally the main objective function with constrained maximization using $r_p(\theta)$ is formulated as $\mathcal{L}^{\text{CLIP}}(\theta)$ in equation 7.

$$\mathcal{L}^{\text{CLIP}}(\theta) = \mathbb{E}_{\pi_\theta}[\min\{r(\theta), \text{clip}(r_p(\theta), 1-\epsilon, 1+\epsilon)\}A(t)] \quad (7)$$

where clip$(\cdot)$ is a clipping function to set $r_p(\theta)$ between $(1-\epsilon, 1+\epsilon)$. The ϵ parameter is the error bound at step k, which can be discounted by γ at each step to reduce the policy π errors and improve the convergence.

The typical gradient updates are performed by equation 4, which is dependent on θ parameters and which makes the updates unstable. Thus, to have stable policy updates, we employ K-FAC, which normalizes the θ parameter updates for the objective function via the Fisher Information Matrix (FIM), as given by equation 8. FIM is a local quadratic approximation of KL-Divergence which approximates the gradient updates of equation 4 independently of θ.

$$F = \mathbb{E}_{\pi_\theta}[\nabla_\theta \log \pi\,(a_t \mid s_t)\,(\nabla_\theta \log \pi\,(a_t \mid s_t))^\top]. \quad (8)$$

B. Instantaneous Constraint

Since the action initially provided by the policy may not comply with instantaneous constraints (and thus may be unfeasible), it is necessary to implement a projection layer to the actor which will project the infeasible action to the feasible space. For that aim, the optimization problem shown in equation 9 is solved, which provides the closest action a to that initially provided by the policy, but subject to complying but subject to complying with the constraint (ϵ_i) [8].

$$\min_a \frac{1}{2}\|a - \pi_\theta(s)\|^2 \quad (9)$$
$$\text{s.t. } C_i(s,a) \le \epsilon_i$$

C. Actor-Critic Architecture of Proposed Technique

To sum up and clarify the relationship of the previous equations, the architectural detail of the proposed technique is given in Figure 1. The proposed actor-critic architecture is identical for each intelligent agent deployed in each BS. The actor updates the policy and the critic approximates the value function. In each BS the instantaneous constraints are handled via a policy network and we extend the neural network that approximates that policy with an additional softmax safety layer (in this way the value of $C_i(s,a)$, given in equation 9, can be learned simultaneously). On the other hand, the MNO performs the generation of the state and broadcasts the orchestration policy to each BS based on the rewards given by each BS.

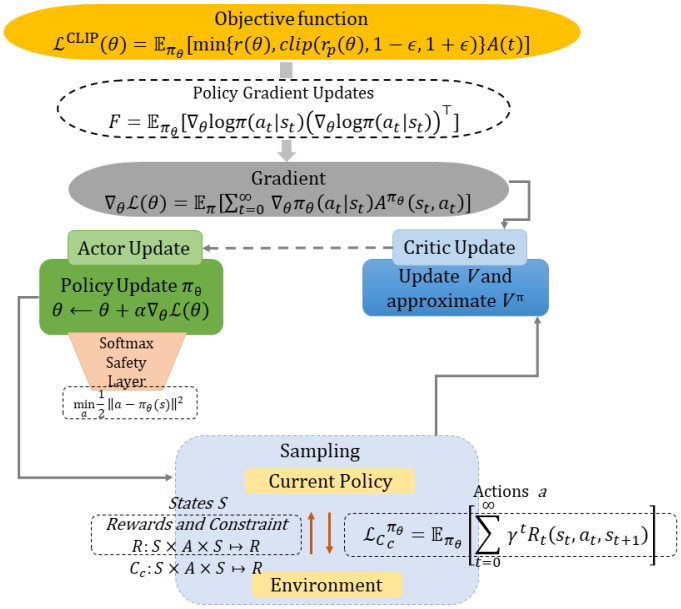
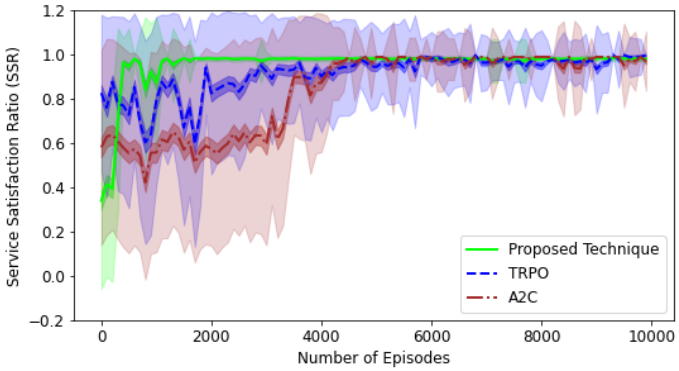


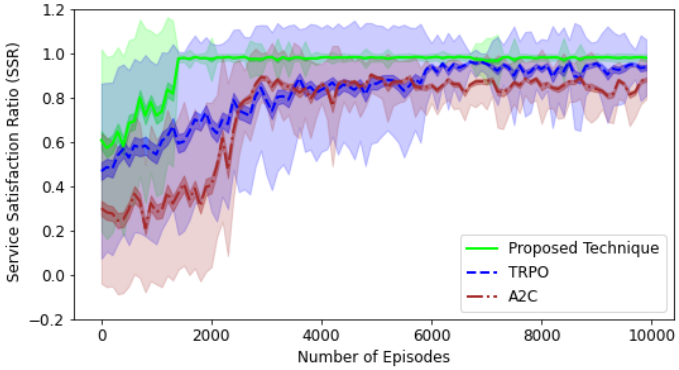
Fig. 1: Actor-Critic architecture of the proposed technique.

IV. Experiments and Simulation Settings

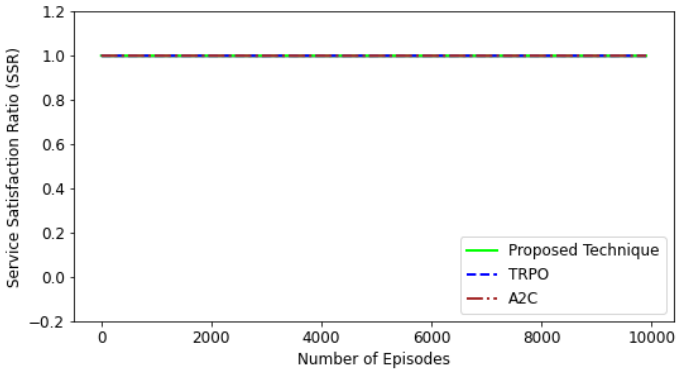
We have implemented the multi-agent RL RAN network model and the proposed technique in Python to analyze its performance. An RL agent is placed in each BS and governed by the MNO. The total bandwidth is set to 1000 Mbps in each BS. The MNO slices the network into three distinct types of user-subscribed services, eMBB, URLLC, and VoLTE, whose SLAs and QoS requirements are obtained from [9] and shown in Table I. The total number of BS is 19, having 2400 moving users where the arrival and leaving of users are set via a Poisson distribution. The dynamics of user mobility is formulated via straight line random bouncing (sLRB) taken from 3GPP [10] mobility model with the specified speeds. Initially, the users are distributed uniformly around the coverage area and then start moving circularly around the BS, eventually bouncing back in a random direction when reaching the edge of the coverage area. The SLA, QoS, speed, and other parameters are given in Table I. In each time slot, set to 1



(a) SSR of eMBB Slice

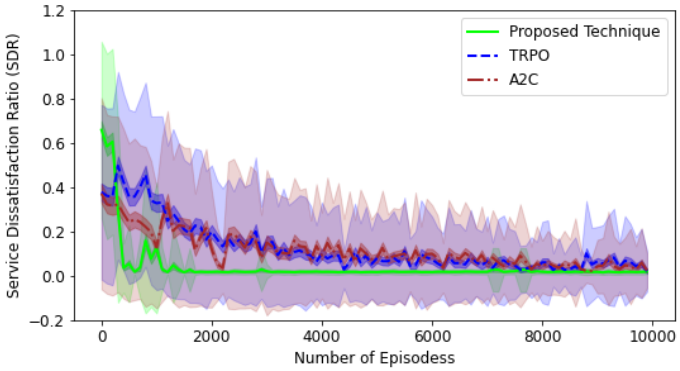


(b) SSR of URLLC Slice

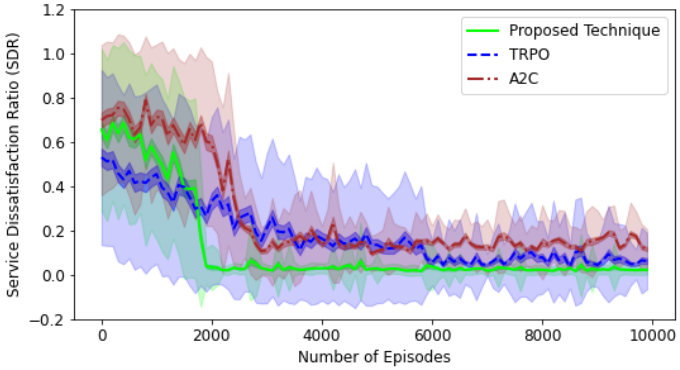


(c) SSR of VoLTE Slice

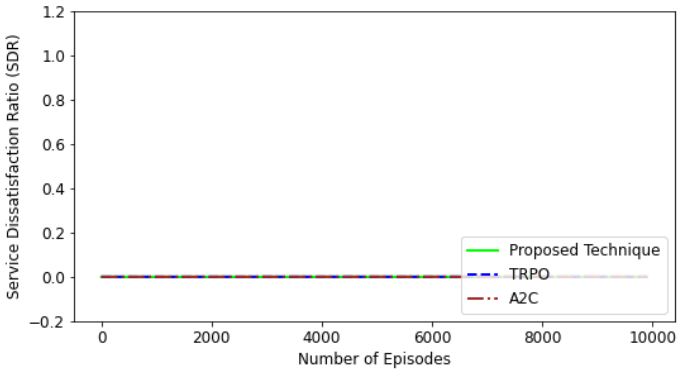
Fig. 2: Illustration of SSR of eMBB, URLLC, and VoLTE



(a) SDR of eMBB Slice



(b) SDR of URLLC Slice



(c) SDR of VoLTE Slice

Fig. 3: Illustration of SDR of eMBB, URLLC, and VoLTE

second (sec), the MNO must re-allocate the bandwidth and adjust it according to the current traffic demands, and update its policy. In addition, each BS also performs the slice band adjustment policy update among each subscriber in a round-robin scheduling manner every 0.5 ms.

To implement the proposed RL technique, first, we set up the network model in the OpenAI gym environment together with the Stable Baselines3 library [11]. Then, we compared the proposed technique with two other RL algorithms, Advantage asynchronous actor-critic network (A2C) [12], and Trust region policy Optimization (TRPO) [7], which are also implemented via OpenAI gym. All the policy networks of the baseline and proposed algorithms comprise two fully connected layers, with 64 and 32 nodes, respectively. The hyper-parameters for the proposed technique are set as follows. The actor learning rate is set to 0.002, the critic learning rate is 0.01, the discount factor γ is 0.99, the clipping ratio ϵ is set to 0.2, trust region radius δ is set to 0.001, and a total number of 10000 episodes are run where each episode is comprised of 500 timesteps. The evaluation of the proposed technique with other RL algorithms is performed via the following key performance indicators (KPIs):

- Service Satisfaction Ratio (SSR): We define SSR as the ratio of traffic packets receiving required resources in terms of bandwidth. SSR is cumulatively dependent on SLA and QoS ranges given in Table I. So, the SSR ranges

from 0 to 1, where 1 is the desired (optimal) ratio. It also states that if SSR is reaching 99% of the desired ratio then the user is experiencing satisfactory service availability.

- Service Dissatisfaction Ratio (SDR): SDR is defined as the cost incurred by the actions of the orchestration policy while satisfying the constraints at each interval. If the agent calculates the total bandwidth allocation and exceeds the constraint, that action (ideally) should not be executed. Hence, the extended layer of the policy network takes the outcome of the unconstrained and constrained actions and computes their difference. Then, actions that are violating the constraints and have a large difference to constrained actions are rejected, thus adapting the served demands to the set of available resources. Due to that, in some intervals, some requests may not be fulfilled according to the SLA and QoS criteria and an increased value in this ratio (optimal SDR ratio is 0) indicates degraded performance at the user level.

A. Results

Simulation results are shown in terms of SSR and SDR in Figures 2 and 3, respectively, when the proposed technique, as well as TRPO and A2C, are used. The figures represent the average of 10 experiments and the associated variance.

As shown in Figure 2, the proposed technique learns a policy that leads to values of SSR that are very close to 1 (which is the optimal value), for the three slices, eMBB (subfigure a), URLLC (subfigure b) and VoLTE (subfigure c). The SSR obtained with the proposed technique is approximately equal (for eMBB and VoLTE) or slightly better (for URLLC) than when TRPO or A2C are used. Moreover, the proposed technique has two main advantages when compared with TRPO and A2C. First of all, the figures show that the proposed technique enables quicker learning of the policy and thus evolves towards the optimal value of SSR in a lower number of episodes. Secondly, when analyzing the variance of the SSR (i.e., the "width" of the shaded area of the plots), it can be seen that, once the policy has been learned, it is much lower for the proposed technique than for the other two methods. Therefore, the proposed technique leads to more stable and consistent results than the other two methods and more quickly.

Figure 3 shows the SDR values for the eMBB, URLLC, and VoLTE slices (subfigures a to c, respectively). In this case, similar conclusions can be obtained. Again, the proposed technique learns more rapidly than the other methods a decision policy with leads to SDR close to 0 (optimal value for this parameter), and the variance is again lower than with the other techniques.

Finally, it is worthy to note that for the VoLTE slice, optimal (or near-optimal) values are obtained from the very beginning, and consistently for all experiments, in contrast to the other slices. This is due to the fact that the SLA requirements for VoLTE in terms of data rate are only 51 kbps, much lower than that of the other slices, so it is easier to comply with them.

V. Conclusions

We have presented an efficient RL-based resource orchestration technique for service slices (eMBB, VoLTE, and URLLC) in distributed RANs. The proposed technique involves an actor-critic architecture with a customized objective function to reinforce the constraints during orchestration decisions using trust region policy optimization. Moreover, the efficient optimization of the proposed technique is achieved by the K-FAC policy update method.

We have analyzed the performance of the proposal in a simulated RAN with a set of BSs and moving users requesting eMBB, VoLTE, and URLLC services. Simulation results have shown that the proposed technique outperforms other baseline RL algorithms (TRPO and A2C). First, the proposed technique learns a more accurate policy that dynamically allocates bandwidth to the different network slices fulfilling their requirements in terms of SLA (rate and latency) and QoS (service availability), thus translating into near-optimal values of SSR and SDR. Second, we have demonstrated that the proposed technique is efficient as it learns a near-optimal policy quicker than the other methods, which is of great importance in evolving (i.e., realistic) environments. Third, it provides more consistent policies, as demonstrated by the low variance when repeating the simulation experiments. However, much future work can be done, like improving spectral efficiency with cumulative rewards.

References

[1] S. E. Elayoubi, S. Ben Jemaa, Z. Altman, and A. Galindo-Serrano, "5G RAN Slicing for verticals: Enablers and challenges," IEEE Commun. Mag., vol. 57, no. 1, pp. 28–34, 2019, doi: 10.1109/MCOM.2018.1701319.

[2] Y. Liu, J. Ding, Z. L. Zhang, and X. Liu, "CLARA: A Constrained Reinforcement Learning Based Resource Allocation Framework for Network Slicing," Proc. - 2021 IEEE Int. Conf. Big Data, Big Data 2021, pp. 1427–1437, 2021, doi: 10.1109/BigData52589.2021.9671840

[3] Y. Xu et al., "Constrained Reinforcement Learning for Resource Allocation in Network Slicing," IEEE Commun. Lett., vol. 25, no. 5, pp. 1554–1558, 2021, doi: 10.1109/LCOMM.2021.3053612.

[4] J. Martens and R. Grosse, "Optimizing neural networks with Kronecker-factored approximate curvature," 32nd Int. Conf. Mach. Learn. ICML 2015, vol. 3, pp. 2398–2407, 2015.

[5] S. H. Lim, H. Xu, and S. Mannor, "Reinforcement learning in robust markov decision processes," Math. Oper. Res., vol. 41, no. 4, pp. 1325–1353, 2016, doi: 10.1287/moor.2016.0779.

[6] J. Song and Y. Wu, "An Empirical Analysis of Proximal Policy Optimization with Kronecker-factored Natural Gradients," pp. 1–8, 2018, [Online]. Available: http://arxiv.org/abs/1801.05566.

[7] J. Schulman, S. Levine, P. Moritz, M. Jordan, and P. Abbeel, "Trust region policy optimization," 32nd Int. Conf. Mach. Learn. ICML 2015, vol. 3, pp. 1889–1897, 2015.

[8] Y. Liu, A. Halev, and X. Liu, "Policy Learning with Constraints in Model-free Reinforcement Learning: A Survey," pp. 4508–4515, 2021, doi: 10.24963/ijcai.2021/614.

[9] NGMN Alliance, "Next generation mobile networks radio access performance evaluation methodology," NGMN Tech. Work. Gr. Steer. Comm., 2008, [Online]. Available: https://www.ngmn.org/wp-content/uploads/NGMN_Radio_Access_Performance_Evaluation_Methodology.pdf

[10] T. Specification and G. Services, "3Gpp Tr 35.909," vol. 0, no. Release 11, pp. 1–28, 2012.

[11] "OpenAI," OpenAI. https://openai.com/ (accessed Oct. 14, 2022).

[12] V. Mnih, "Asynchronous methods for deep reinforcement learning," 33rd Int. Conf. Mach. Learn. ICML 2016, vol. 4, pp. 2850–2869, 2016.