This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LCOMM.2020.3001227, IEEE Communications Letters

The LSTM-based Advantage Actor-Critic Learning for Resource Management in Network Slicing with User Mobility

Rongpeng Li, Chujie Wang, Zhifeng Zhao, Rongbin Guo, and Honggang Zhang

***Abstract*—Network slicing aims to efficiently provision diversified services with distinct requirements over the same physical infrastructure. Therein, in order to efficiently allocate resources across slices, demand-aware inter-slice resource management is of significant importance. In this paper, we consider a scenario that contains several slices in a radio access network with base stations that share the same physical resources (e.g., bandwidth or slots). We primarily leverage advantage actor-critic (A2C), one typical deep reinforcement learning (DRL) algorithm, to solve this problem by considering the varying service demands as the environment *state* and the allocated resources as the environment *action*. However, given that the user mobility toughens the difficulty to perceive the environment, we further incorporate the long short-term memory (LSTM) into A2C, and put forward an LSTM-A2C algorithm to track the user mobility and improve the system utility. We verify the performance of the proposed LSTM-A2C through extensive simulations.**

***Index Terms*—network slicing, deep reinforcement learning, long short-term memory (LSTM), advantage actor critic (A2C), user mobility**

I. Introduction

The emerging fifth-generation (5G) cellular network is envisioned to cater a wide range of services with significantly distinct service requirements like enhanced mobile broadband (eMBB), massive machine-type communications (mMTC), and ultra-reliable and low-latency communications (URLLC) [1]. In order to achieve such a goal, the concept of network slicing has recently been proposed by virtually slicing the physical and computational resources of one network infrastructure to meet the diverse needs of a range of 5G users [2]. In order to provide better-performing and cost-efficient services, inter-slice resource management for radio access network slicing has to track dynamic request patterns of user equipment (UE) and allocate the resources coherently, while guaranteeing an acceptable spectrum efficiency (SE) and satisfying the service level agreements (SLAs) [3]–[5]. The classical dedicated resource allocation fails to address these problems simultaneously [6], [7]. Instead, it becomes incentive to design an intelligent resource management solution and reinforcement learning (RL) emerges as a promising solution

R. Li and H. Zhang are with Zhejiang University; C. Wang was with Zhejiang University and is with Huawei (email: {lirongpeng, 21731125, honggangzhang}@zju.edu.cn). R. Guo and Z. Zhao are with Zhejiang Lab (email: {guorb, zhaozf}@zhejianglab.com).

This work was supported by National Natural Science Foundation (NSF) of China (No. 61701439, 61731002), Zhejiang Key R&D Plan (No. 2019C01002, 2019C03131), the Project sponsored by Zhejiang Lab (2019LC0AB01), Zhejiang Provincial NSF of China (No. LY20F010016).

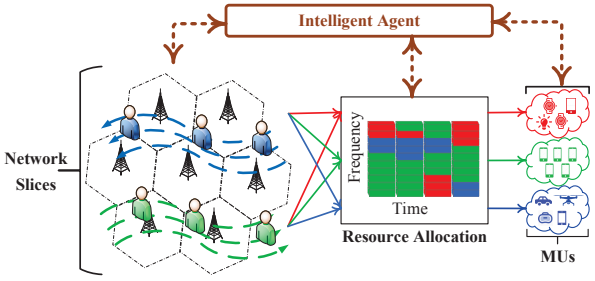

Fig. 1. The access network scenario with multiple base stations and several moving users.

[5], [6], [8]–[11]. However, such efforts have shed little light on the possible impact of user mobility on the perceived demand. In other words, the mobility of users could exacerbate the fluctuation of service requests and make the on-demand resource management for network slicing more challenging. On the other hand, previous works have demonstrated the effectiveness of long short-term memory (LSTM) to forecast the user mobility [12], [13]. In this letter, following the works in [5], [6], [11] to apply RL for inter-slice resource management in network slicing, we incorporate LSTM into the advantage actor-critic (A2C) algorithm and propose an LSTM-A2C algorithm, so as to gain the capability to better track user's mobility and improve the system utility.

The remainder of the paper is organized as follows: Section II formulates the system model. Section III gives the details of LSTM-A2C, while Section IV presents the detailed simulation results. Finally, Section V summarizes the paper.

II. System Model

As depicted in Fig. 1, we consider an access network scenario consisting of multiple base stations like [6]. But different from [6], UEs in this scenario move with different mobility patterns. There exists a list of existing slices $\{1, \cdots, N\}$ sharing the aggregated bandwidth W and having fluctuating demands $\boldsymbol{d} = (d_1, \cdots, d_N)$. We aim to maximize the utility function $f(\cdot)$ which is defined as the weighted sum of SE and SLA satisfaction ratio (SSR) of different services by dynamically adjusting the allocated bandwidth $\boldsymbol{w} = (w_1, \cdots, w_N)$ to each slice. Mathematically, the optimization objective can be formulated as

$$\begin{aligned} \max_{\boldsymbol{w}} \quad & f = \alpha \cdot \mathrm{SE}(\boldsymbol{d}, \boldsymbol{w}) + \boldsymbol{\beta} \cdot \mathrm{SSR}(\boldsymbol{d}, \boldsymbol{w}) \\ s.t. \quad & w_1 + \cdots + w_N = W \\ & w_i = k \cdot \Delta, \ \forall i \in [1, \cdots, N] \end{aligned} \tag{1}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LCOMM.2020.3001227, IEEE Communications Letters

where SE could be obtained from the downlink signal-to-noise ratio (SNR) according to the Shannon capacity while SSR could be computed in terms of the predefined SLAs. α and $\boldsymbol{\beta} = (\beta_1, \cdots, \beta_N)$ denotes the relative importance of SE and SSR, k is an integer and Δ is the minimum allocated bandwidth per slice.

Notably, the traffic demands $\boldsymbol{d}$ at each scheduling period depends not only on the traffic model but also on the dynamic user distribution when users are moving among different BSs. Usually, the user mobility exacerbates the fluctuation of service requests, making the bandwidth allocation problem in (1) more complicated and difficult to yield a direct solution. However, we can map the RAN scenario to the context of Markov decision process (MDP) by taking the number of arrived packets in each slice within a specific time window as the state s and the bandwidth allocated to each slice as the action a, as well as deriving the reward r from SE and SSR. Since the traffic demands are unknown apriori, RL is adopted to tackle this inter-slice resource allocation problem and find the optimal policy for resource management in network slicing.

III. The LSTM-A2C Decision Algorithm

In order to proactively adapt to the dynamic environment and make proper decisions, we incorporate the LSTM algorithm into the A2C algorithm and propose an decision solution, named LSTM-A2C.

A. Basics of A2C and LSTM

Beforehand, we briefly introduce RL. RL aims to optimize the control and decision-making ability under a specific environment through trial-and-error strategy. At each time-step t, the agent receives a state $s_t \in \mathcal{S}$ and selects an action a_t from the set of possible actions $\mathcal{A}$ according to its policy $\pi(a_t|s_t)$. After interacting with the environment, the agent reaches the next state s_{t+1} and receives a reward r_t. The total accumulated return at time-step t is defined as $R_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k}$, where γ is the discount factor between 0 and 1. The goal of the RL agent is to maximize the expected return from each state s_t, which can be estimated by the action-value function $Q(s,a)$ and the state-value function $V(s)$. The action-value function $Q(s,a) = \mathbb{E}[R_t|s_t = s, a_t = a]$ estimates the expected return for selecting action a in state s at time-step t, while the state-value function $V(s) = \mathbb{E}[R_t|s_t = s]$ simply estimates the average expected return from state s. The A2C is an effective way to approximate only the state value function $V(s_t)$ rather than both $Q(s_t, a_t)$ and $V(s_t)$, so as to reduce the number of parameters and simplify the learning process. In particular, the terminology "advantage" refers to $A(s_t, a_t) = Q(s_t, a_t) - V(s_t)$ representing the advantage of performing action a_t at state s_t. Besides, the policy $\pi(a_t|s_t)$ is the actor and the value function $V(s_t)$ is the critic. Typically, $A(s_t, a_t)$ can be estimated by TD error with little variance, since

$$\begin{aligned} A(s_t, a_t) &= Q(s_t, a_t) - V(s_t) = \mathbb{E}[R_t|s_t, a_t] - V(s_t) \\ &\approx r_t + \gamma V(s_{t+1}|s_t, a_t) - V(s_t) = \delta(s_t) \end{aligned} \tag{2}$$

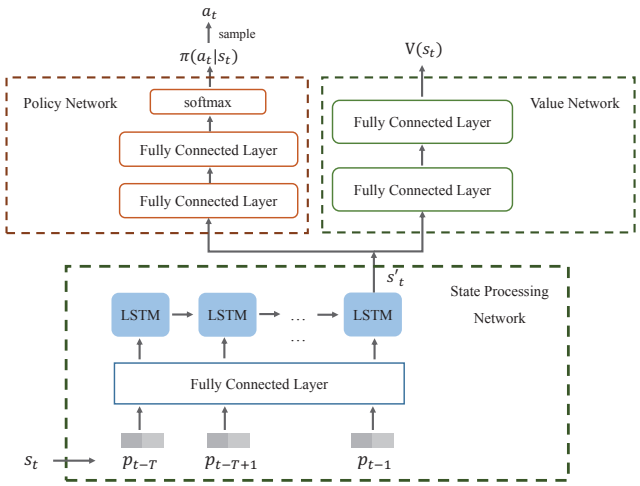

Fig. 2. The architecture of the proposed LSTM-A2C algorithm.

The gradient of the actor is $\nabla_\theta \log \pi(a_t|s_t;\theta)\delta(s_t)$, and the loss function of critic is given as $\mathcal{L}_{\text{Critic}} = \delta(s_t)^2$.

On the other hand, LSTM is an artificial recurrent neural network (RNN) architecture for prediction and classification with astonishing success [12], [14]. Due to the space limitation, we omit the details of LSTM here and interested readers could resort to [12], [14] to find the details.

B. LSTM-A2C

The LSTM-A2C model takes advantage of the superior series processing capability of LSTM to capture the temporal variation regularity of service requests due to user mobility and further applies the powerful learning and decision-making capability of A2C mechanism to optimize its bandwidth allocation policy based on the comprehensive understanding of the dynamic environment. The entire algorithm is shown in Algorithm 1.

In order to capture the temporal correlation of service requests, we define the state $s_t = \{p_{t-T}, p_{t-T+1}, \cdots, p_{t-1}\}$ as a series of observation vectors, where each observation vector p_t is the number of arrived packets in each slice within the t-th scheduling period. The action $a_t = \{w_1, \cdots, w_N\}$ is defined as the bandwidth allocation to each slice. The reward, which is one of the most important parts in an RL problem, needs to be specially designed to guide the agent in finding the optimal policy. Due to the limited total bandwidth, there exist mutual affects among SE and SSRs of different slices. Therefore, the change of any term in the utility function may not influence the final utility in a definite way. In order to take account of the scarcity of spectrum in radio access and improve the system utility in an appropriate and reasonable way, we aim to minimize the loss of SE or even increase it as much as possible while guaranteeing the SSR of each slice rather than blindly sacrificing SSR in exchange for an increase in SE. Therefore, we design the reward function as

$$r_t = g_2(\text{SE}) I_{\text{SSR}}(s_t, a_t) + g_1(q_1, \cdots, q_N)(1 - I_{\text{SSR}}(s_t, a_t)) \tag{3}$$

where $I_{\text{SSR}}(s_t, a_t) = \{0, 1\}$ is an indicator function and q_i is the SSR of the service provided by slice i. When all slices

Algorithm 1 The LSTM-A2C Algorithm

Initialize the actor parameters θ_a and critic parameters θ_c of LSTM-A2C;
Initialize an empty buffer of length T and the time-step $t = T + 1$;
for i = 1 to T **do**
  Randomly choose an action $a_i \in \mathcal{A}$ and performs a_i;
  At the end of the i-th scheduling period, the agent gets the observation p_i;
  Append p_i to the end of the buffer;
end for
repeat
  Concatenate each observation in the buffer to form $s_t = \{p_{t-T}, \cdots, p_{t-1}\}$ at the start of the t-th scheduling period;
  The state processing network takes s_t as input and calculates s'_t;
  The policy network takes s'_t as input and calculates the probability distribution of actions $\pi(a_t|s_t)$;
  The value network takes s'_t as input and calculates the state-value $V(s_t)$;
  The agent samples the action a_t according to $\pi(a_t|s_t)$ and performs a_t;
  The agent receives the reward r_t and gets the new observation o_t;
  Delete the first observation in the buffer and append p_t to the end of the buffer;
  Concatenate each observation in the buffer to form $s_{t+1} = \{p_{t-T+1}, ..., p_t\}$;
  The state processing network takes s_{t+1} as input and calculates s'_{t+1};
  The value network takes s'_{t+1} as input and calculates the state-value $V(s_{t+1})$;
  Calculate $\delta_t(s_t) = r_t + \gamma V(s_{t+1}) - V(s_t)$;
  Update θ_c of the critic network according to (7);
  Update θ_a of the actor network according to (5);
  Update $t \leftarrow t + 1$;
until The predifined maximum number of iterations has been completed.

satisfy the predefined SLA requirements, $I_{\text{SSR}}(s_t, a_t) = 1$; $I_{\text{SSR}}(s_t, a_t) = 0$ otherwise. The first step $g_1(q_1, \cdots, q_N)$ is the reward function depends only on SSRs in the case of $I_{\text{SSR}}(s_t, a_t) = 0$. And the second step $g_2(\text{SE})$ is the reward function correlates only with SE after all SLA requirements have been satisfied.

As shown in Fig. 2, the proposed LSTM-A2C solution is mainly composed of three parts, the state processing network, the policy network, and the value network.

- The state processing network is responsible for capturing temporal correlation of service requests among different scheduling periods. It contains a fully connected (FC) layer and an LSTM layer. The FC layer takes the current state s_t as input and then sends the extracted feature tensor to the LSTM layer. The LSTM network is responsible for capturing the variation regularity of service requests from the past T observation vectors. Therefore, after T recursive updates, the hidden state at the last time-step can be viewed as a completed representation of the environment s', which can be utilized for subsequent action generation and value estimation.
- The policy network is responsible for generating actions based on current states. It contains two FC layers. The first FC layer takes the output of the state processing network s' as input to extract action-related features. The second FC layer contains $|\mathcal{A}|$ neurons and followed by a Softmax function to map the output into the probability of different actions $\pi(a_t|s_t)$. Finally, the action to be performed is sampled according to $\pi(a_t|s_t)$.
- The value network is responsible for estimating state values. It also contains two FC layers. The first FC layer takes the output of the state processing network s' as input to extract value-related features, which is sent to the second FC layer with only one neuron to get the state value $V(s_t)$.

The state processing network and the value network together can be viewed as the critic network with parameter θ_c and the policy network can be seen as the actor network with parameter θ_a. In order to encourage exploration, we add entropy regularization to the loss function of the actor network, which could be further written as

$$\mathcal{L}_{\text{Actor}} = -[\delta_t(s_t;\theta_c)\log\pi(a_t|s_t;\theta_a) + \beta_e H(\pi(a_t|s_t;\theta_a))] \tag{4}$$

where β_e is the weight of the action entropy. The parameter update of the actor network can be represented as

$$\theta_a \leftarrow \theta_a + \frac{\partial \log\pi(s_t|s_t;\theta_a)}{\partial\theta_a}\delta_t(s_t;\theta_c) + \beta_e\frac{\partial H(\pi(a_t|s_t;\theta_a))}{\partial\theta_a}, \tag{5}$$

Consistent with the previous introduction, the loss function of the critic network is formulated as

$$\mathcal{L}_{\text{Critic}} = (r_t + \gamma V(s_{t+1};\theta_c) - V(s_t;\theta_c))^2, \tag{6}$$

and the parameter update of the critic network can be represented as

$$\theta_c \leftarrow \theta_c + \delta_t(s_t;\theta_c)\frac{\partial V(s_t;\theta_c)}{\partial\theta_c}. \tag{7}$$

IV. Simulation Results and Numerical Analysis

A. Simulation Environment Settings

We consider a RAN scenario with three types of services (i.e., VoLTE, eMBB, URLLC) and corresponding slices in a simulation area of 240 m $\times$ 240 m, where there exist 1200 UEs as a default and multiple BSs. For simplicity, we assume that the UEs within the same slice share the moving pattern (e.g., distribution of velocities and moving direction). When a UE reaches the bound of the simulation area, its direction will bounce. The specific configuration of each UE and its moving speed are described in Table I. In addition, each UE generates service traffics as summarized in Table I based on 3GPP TR 36.814 [15] and TS 22.261 [16]. The total bandwidth is 10 MHz, and the bandwidth allocation resolution is 200 kHz. We aim to optimize the bandwidth allocation for the central BS with 40 meters' coverage radius in the simulation area.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LCOMM.2020.3001227, IEEE Communications Letters

TABLE I
A SUMMARY OF KEY SETTINGS FOR TRAFFIC GENERATION PER SLICE

		VoLTE	eMBB	URLLC
Bandwidth		10 MHz		
Scheduling		Round robin per slot (0.5 ms)		
Slice Band Adjustment		1 second (2000 scheduling slots)		
Channel		Rayleigh fading		
UE No	1200 (*Default*)	200	400	600
	2400	400	800	1200
Speed	Fixed (*Default*)	1m/s	4m/s	8m/s
	Varying	Uniform [Min: 1m/s, Max: 5m/s]	Uniform [Min: 1m/s, Max: 5m/s]	Uniform [Min: 6m/s, Max: 10m/s]
Distribution of Inter-Arrival Time per User		Uniform [Min: 0, Max: 160ms]	Truncated Pareto [Exponential Para: 1.2, Mean: 6ms, Max: 12.5 ms]	Exponential [Mean: 180ms]
Distribution of Packet Size		Constant [40 Byte]	Truncated Pareto [Exponential Para: 1.2, Mean: 100 Byte, Max: 250 Byte]	Constant [0.3 MByte]
SLA	Rate	51 kbps	100 Mbps	10 Mbps
	Latency	10 ms	10 ms	1 ms

We expect all three services can achieve satisfactory SSR for subscribers, which is set to be 0.95 in this paper. When this condition is not satisfied, $I_{\text{SSR}}(s_t, a_t) = 0$, and the learning goal of the agent is to improve SSRs under the stimulation of $g_1(\cdot)$ function. Since the URLLC's SLA is relatively more difficult to achieve among the three services, we set a comparatively larger negative reward (i.e., -5) when VoLTE or eMBB does not achieve its predefined SSR requirement, thus making the $g_1(\cdot)$[1] function in (8) to be proportional to $\text{SSR}_{\text{URLLC}}$ and accelerating the learning process.

$$g_1 = \begin{cases} (q_{\text{u}} - 0.7) * 10, & \text{if } q_{\text{v}} \geq 0.95 \text{ and } q_{\text{e}} \geq 0.95, \\ -5, & \text{if } q_{\text{v}} < 0.95 \text{ or } q_{\text{e}} < 0.95, \end{cases} \tag{8}$$

where q_{u}, q_{v}, and q_{e} denote the SSR of URLLC, VoLTE and eMBB, respectively. After the predefined SLA requirements have been guaranteed, the agent is guided to fine-tune its policy to achieve higher SE in (9) by giving an additional reward according to the value and the importance coefficients of SE as

$$g_2 = 4 + \max\{0, \alpha(\text{SE} - 280) * 10\}. \tag{9}$$

B. Simulation Results

We evaluate the performance of LSTM-A2C and compare the results with the classical deep Q-networks (DQN) [5], A2C, GAN-DDQN [6] and hard slicing[2]. In particular, for the LSTM-A2C, the learning rates of actor-network and critic-network are set to be 0.005 and 0.008 respectively. The observation length of LSTM is set to be 10. And the entropy regularization used for encouraging exploration is set to be 0.001. We first set the speed of UEs fixed and the

[1]Notably, $g_1(\cdot)$ and $g_2(\cdot)$ apply for all RL algorithms simultaneously.

[2]As for the hard slicing solution, each slice is allocated with $\frac{1}{3}$ of the whole bandwidth, since there are three types of services in total.

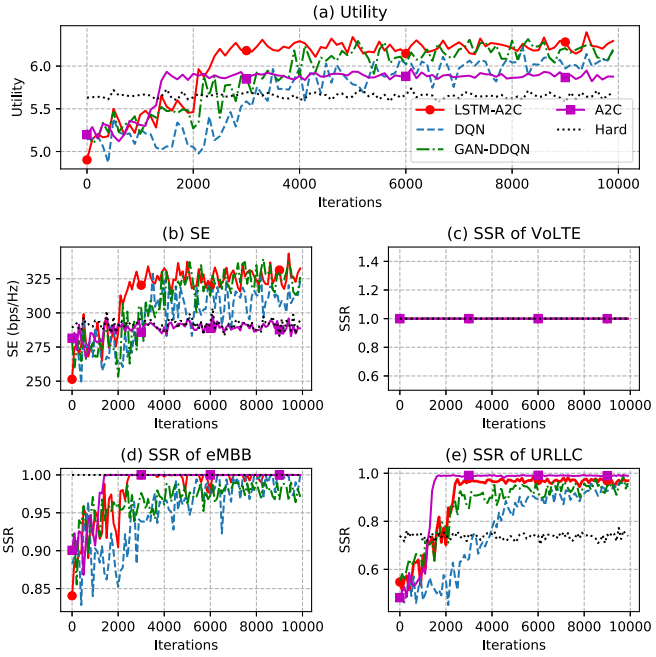

Fig. 3. The comparison of system utility, SE, and SSRs for different methods.

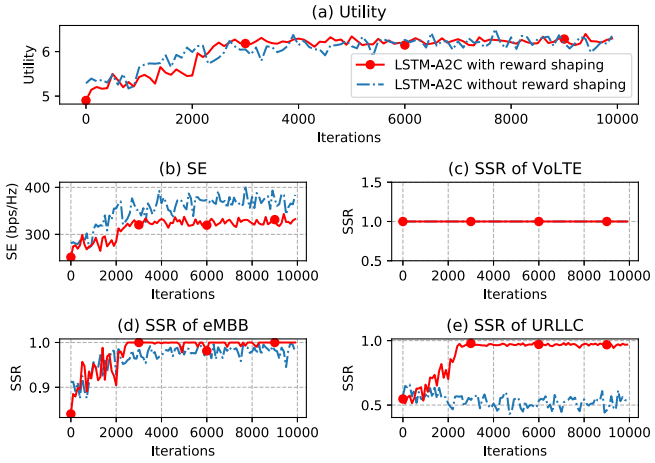

Fig. 4. The impact of reward shaping on performance.

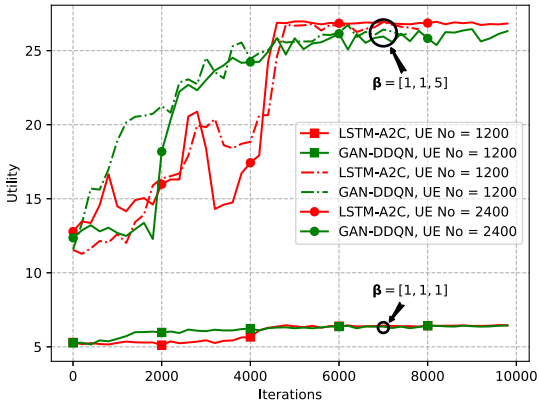

Fig. 5. Extensive comparison between LSTM-A2C and GAN-DDQN.

importance weights in the optimization objective as $\alpha = 0.01$, $\boldsymbol{\beta} = [1, 1, 1]$[3]. Fig. 3(a) depicts the variations of the system utility with respect to the iteration index. It can be observed that all the RL algorithms have apparent performance improvements through learning and ultimately achieve higher system utility than hard slicing. Among RL algorithms, although A2C shows faster convergence, its performance is not as good as the proposed LSTM-A2C algorithm, which can basically maintain the system utility at above 6.2 after around 3000 iterations. In addition, the LSTM-A2C algorithm also exhibits superior performance than the DQN algorithm terms of both convergence rate and obtained utility. Compared with GAN-DDQN, the state-of-the-art method, LSTM-A2C gives slightly superior performance and convergence rate. Fig. 3(b) - Fig. 3(e) further presents the performance comparison on SE and SSR of each service respectively. From the perspective of SSRs, it can be observed that the SLA of all three slices have reached our expectations through learning. For VoLTE slice, it's easier to achieve high SSR due to its low requirements of throughput and latency. For the other slices, despite their high requirements of throughput or latency, both LSTM-A2C and A2C can achieve almost 100% SSR after about 3000 iterations. Compared with LSTM-A2C, the performance of DQN is slightly worse, while A2C shows slightly faster convergence and higher SSR. From the perspective of SE, the proposed LSTM-A2C achieves the highest SE among the three methods, indicating that LSTM-A2C can capture the temporal variation regularity of service requests and adjust the bandwidth allocation flexibly so as to improve SE on the basis of guaranteeing SLAs of different services. However, A2C shows trivial improvement in SE because it tends to allocate bandwidth conservatively to get stable yet inferior SSRs than LSTM-A2C. Compared with GAN-DDQN, LSTM-A2C converges to the same final level as GAN-DDQN but exhibits a more stable convergence curve.

Fig. 4 compares the performance of LSTM-A2C with and without the reward shaping functions in (8) and (9), and demonstrates the reward shaping method yields more stable performance. Moreover, shaping the rewarding guarantees the system utility while avoids sacrificing URLLC's SSR.

Fig. 5 further compares the LSTM-A2C and GAN-DDQN under extensive settings, where the speed of UEs varies according to Table I. It can be observed that contrary to our previous findings, LSTM-A2C converges slightly slower than GAN-DDQN but is capable of leading to encouragingly comparative results, if other settings remain unchanged. Such an observation is consistent with claim [6] that GAN-DDQN performs well under different varying scenarios. If we change $\boldsymbol{\beta}$ as $[1, 1, 5]$, LSTM-A2C gives slightly superior performance and such a performance gain becomes larger along with the increase in the number of UEs. This implies that instead of capturing the variations by approximating the complicated distributions like GAN-DDQN, embedding the variation prediction into RL sounds feasible as well.

[3]Notably, such a setting make SE slightly dominates the system utility.

V. Conclusion and Future Works

In this paper, we have provided an intelligent resource management solution for network slicing. Specifically, we have incorporated the LSTM network into A2C algorithm and proposed an intelligent decision algorithm, LSTM-A2C, so as to better capture the demand variations due to user mobility and make appropriate resource allocation decisions in a dynamic network slicing environment. Experimental results show that the proposed LSTM-A2C can not only guarantee the SLA of different services in cases with large fluctuations in user requests, but also improve the spectrum efficiency. We boldly argue that the in order to tackle the user mobility incurred variations, LSTM-A2C is a promising candidate to make full use of spectrum resources and improving system utility effectively.

However, there still exists many remaining issues to be addressed. For example, (1) How to further design an end-to-end learning algorithm to integrate the inter-slice resource allocation and the intra-slice user scheduling? (2) How to balance the tradeoff between the performance gain and computational complexity for reinforcement learning-based solutions? (3) How to take advantage of inverse reinforcement learning to design a method to yield a generalized reward function?

References

[1] S. E. Elayoubi, *et al.*, "5G RAN slicing for verticals: Enablers and challenges," *IEEE Commun. Mag.*, vol. 57, no. 1, pp. 28–34, Jan. 2019.

[2] X. Zhou, *et al.*, "Network slicing as a service: Enable industries own software-defined cellular networks," *IEEE Commun. Mag.*, vol. 54, no. 7, pp. 146–153, Jul. 2016.

[3] P. Rost, *et al.*, "Implementing radio access network slicing in a mobile network," US Patent US20 190 159 024A1, May, 2019.

[4] J. Zheng and G. de Veciana, "Elastic multi-resource network slicing: Can protection lead to improved performance?" *arXiv:1901.07497 [cs]*, Jan. 2019.

[5] R. Li, *et al.*, "Deep reinforcement learning for resource management in network slicing," *IEEE Access*, vol. 6, pp. 74 429–74 441, Nov. 2018.

[6] Y. Hua, *et al.*, "GAN-powered deep distributional reinforcement learning for resource management in network slicing," *IEEE J. Sel. Area. Comm.*, vol. 38, no. 2, pp. 334–349, Feb. 2020.

[7] R. Li, *et al.*, "Intelligent 5G: When cellular networks meet artificial intelligence," *IEEE Wireless Commun.*, vol. 5, no. 24, pp. 175–183, Oct. 2017.

[8] H. Xiang, *et al.*, "A realization of Fog-RAN slicing via deep reinforcement learning," *IEEE Trans. Wireless Commun.*, 2020.

[9] M. Yan, *et al.*, "Intelligent resource scheduling for 5G radio access network slicing," *IEEE Trans. Veh. Tech.*, vol. 68, no. 8, pp. 7691–7703, Aug. 2019.

[10] V. Sciancalepore, *et al.*, "Rl-nsb: Reinforcement learning-based 5G network slice broker," *IEEE/ACM Trans. Netw.*, vol. 27, no. 4, pp. 1543–1557, Aug. 2019.

[11] C. Qi, *et al.*, "Deep reinforcement learning with discrete normalized advantage functions for resource management in network slicing," *IEEE Commun. Lett.*, vol. 23, no. 6, pp. 1337–1341, Aug. 2019.

[12] Y. Hua, *et al.*, "Deep learning with long short-term memory for time series prediction," *IEEE Commun. Mag.*, vol. 57, no. 6, pp. 114–119, Jun. 2019.

[13] C. Wang, *et al.*, "Exploring trajectory prediction through machine learning methods," *IEEE Access*, vol. 7, pp. 101 441–101 452, Jul. 2019.

[14] K. Greff, *et al.*, "LSTM: A search space odyssey," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 10, pp. 2222–2232, Oct. 2017.

[15] 3GPP, "Evolved universal terrestrial radio access (E-UTRA); Further advancements for E-UTRA physical layer (3GPP TR 36.814)," Jan. 2015. [Online]. Available: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=2493

[16] ——, "Service requirements for the 5G system (3GPP TS 22.261)," Mar. 2017. [Online]. Available: https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3107