Resource Allocation in Uplink NOMA-IoT Networks: A Reinforcement-Learning Approach

Waleed Ahsan, *Student Member, IEEE,* Wenqiang Yi, *Student Member, IEEE,* Zhijin Qin, *Member, IEEE,* Yuanwei Liu, *Senior Member, IEEE,* and Arumugam Nallanathan, *Fellow, IEEE*

Abstract

Non-orthogonal multiple access (NOMA) exploits the potential of power domain to enhance the connectivity for Internet of Things (IoT). Due to time-varying communication channels, dynamic user clustering is a promising method to increase the throughput of NOMA-IoT networks. This paper develops an intelligent resource allocation scheme for uplink NOMA-IoT communications. To maximise the average performance of sum rates, this work designs an efficient optimization approach based on two reinforcement learning algorithms, namely deep reinforcement learning (DRL) and SARSA-learning. For light traffic, SARSA-learning is used to explore the safest resource allocation policy with low cost. For heavy traffic, DRL is used to handle traffic-introduced huge variables. With the aid of the considered approach, this work addresses two main problems of the fair resource allocation in NOMA techniques: 1) allocating users dynamically and 2) balancing resource blocks and network traffic. We analytically demonstrate that the rate of convergence is inversely proportional to network sizes. Numerical results show that: 1) compared with the optimal benchmark scheme, the proposed DRL and SARSA-learning algorithms achieve high accuracy with low complexity and 2) NOMA-enabled IoT networks outperform the conventional orthogonal multiple access based IoT networks in terms of system throughput.

Index Terms

Deep reinforcement learning, internet of things, non-orthogonal multiple access, power allocation, SARSA learning, user clustering

I. Introduction

Internet of things (IoT) enable millions of devices to communicate simultaneously. It is predicted that the number of IoT devices will rapidly increase in the next decades [2]. Owing to a

W. Ahsan, W. Yi, Z. Qin, Y. Liu, and A. Nallanathan are with Queen Mary University of London, London, UK (email:{w.ahsan, w.yi, z.qin, yuanwei.liu, a.nallanathan}@qmul.ac.uk).

Part of this work was submitted in IEEE International Conference on Communications (ICC) Workshops, June, Ireland, 2020 [1].

large number of time-varying communication channels, the dynamic network access with massive connectivity becomes a key requirement for future IoT networks. Recently, non-orthogonal multiple access (NOMA) is evolved as a promising approach to solve this problem [3], [4]. The key benefit of using NOMA is that NOMA exploits power domain to enable more connectivity than the traditional orthogonal multiple access (OMA). More specifically, NOMA supports multiple users in the same time/frequency resource block (RB) by employing superposition coding at transmitters and successive interference cancellation (SIC) techniques at receivers [5]. Various model-based schemes have been proposed to improve different metrics of NOMA-IoT networks, such as coverage performance, energy efficiency, system throughput (sum-rates), etc. Additionally, on the importance of sum-rates, the recent work in wireless networks based on the state of the art reflective intelligent surfaces (RIS) considered sum-rate maximization objective function [6]. The sum-rate is an important parameter to depict the average performance of wireless networks in detail for each user. Due to this sum-rate is widely used as significant performance indicator in wireless networks by the research community [7] and [8]. It shows the significance of the sum-rate maximization based objective functions. Regrading the system design, the uncertainty and dynamic mechanisms of wireless communication environments are difficult to be depicted by an accurate model. The dynamic mechanism involves spectral availability, channel access methods (e.g., OMA, NOMA, hybrid systems, etc.), and dynamic traffic arrival. Especially in practical NOMA systems by allowing resource share among more than one users the process is more dynamic, when the number of users are joining and leaving the network in short term and long term basis. Numerous model-based techniques target to solve dynamic behaviour of wireless networks but failed to provide long-term performance outcomes [9], [10], [11], [12] and [13]. Moreover, due to the absence of learning abilities, to provide long term network stability the computational complexity of traditional schemes becomes ultra-high. This is due to the fact that, by default traditional approaches cannot extract knowledge from any given problem (e.g, given distributions) online. Fortunately, the online learning properties of recently developed machine learning (ML) methods are extremely suitable to handle such type of dynamic problems [14].

A. Related Works and Motivations

1) Studies on NOMA-IoT Networks: Due to the aforementioned benefits, academia has proposed numerous studies on the optimization of resource allocation in NOMA-enabled IoT

networks. For single-cell scenarios, the authors in [10] proposed a two-stage NOMA-based model to optimize the computation offloading mechanism for IoT networks [15]. In the first stage, a large number of IoT devices are clustered into several NOMA groups depending on their channel conditions. In the second stage, different power levels are allocated to users to enhance the network performance. The comparison between uplink NOMA-IoT and OMA-IoT is presented in [16], which considered the optimal selection of targeted data rates for each user. Regarding downlink transmission, the similar topic was studied in [17] and [18]. Different from others, in [19] using 2D matching theory authors performed dynamic resource allocations considering energy efficiency for downlink NOMA. Similarly, in [12] for the massive Machine Type Communications (mMTC) usage scenario, also known as massive Internet of Things (mIoT) dynamic resource management is performed with Sparse Code Multiple Access (SCMA) domain using conventional mathematical tools. The authors in [20] proposed a general power allocation scheme for uplink and downlink NOMA to guarantee the quality of service (QoS). In [21], NOMA scheduling schemes in terms of power allocation and resource management were optimized to realize the massive connectivity in IoT networks. For multi-cell scenarios, the impact of NOMA on large scale multi-cell IoT networks was investigated in [22]. To characterize the communication distances, the authors in [23] analysed the performance of large scale NOMA communications via stochastic geometry. It is worth noting that NOMA-IoT channels are time-varying in the real world. Therefore, the study in [24] considered a practical framework with dynamic channel state information for evaluating the performance of massive connectivity. The authors in [25], [26], and [27] discussed the advantages of various NOMA-IoT applications. Interestingly, the proposed schemes introduced artificial intelligence (AI) methods to solve some practical challenges of NOMA-IoT systems. For both uplink and downlink scenarios, AI-based multi-constrained functions can be utilized to optimise multiple parameters simultaneously.

2) Studies on ML-based NOMA Systems: Due to the dynamic nature of NOMA-IoT communications, traditional methods may not be suitable for such type of networks [13]. Note that ML-based methods are capable to handle the complex requirement of future wireless networks via learning. In [28], one typical deep learning method, namely long short-term memory (LSTM) [29], was applied for the maximization of user rates by minimizing the received signal-to-noise-ratio (SINR). In [30], successive approximation based algorithm was proposed to minimize outage probabilities through optimizing power allocation strategies. For next-generation ultra-dense networks, ML-aided user clustering schemes were discussed in [31] for obtaining

efficient network management and performance gains. Because using clustering schemes, the entire network can be divided into several small groups, which helps to ease the resource management [32]. Regarding AI-based cluster techniques, in [33] and [34], resources were assigned to the most suitable user to ensure the best QoS for unmanned aerial vehicle (UAV) networks and millimetre wave networks, respectively. It is worth noting that the optimization of clustering is an NP-hard problem. Therefore, for such type of problems the authors in [28], [31], and [35] recommended to use AI instead of conventional mathematical models. Currently, realistic datasets are not available for most of the machine learning algorithms, to overcome this designers use synthetic dataset for simulations. The data set is generated for a certain environment so it is difficult to depict general property and online scenarios of wireless networks. Therefore, algorithms like reinforcement learning plays very important role where data is collected online (during simulation) to learn the given search space for the simulation requirements. There are various Q-learning algorithm variants used for NOMA systems.Due to inefficient learning mechanism other methods like traditional Q-learning and Multi-arm bandits (MABs) are heavily influenced by regret (negative reward) [36] [37]. On the other hand two most powerful methods, deep reinforcement learning (DRL) and SARSA learning created by google deep mind [38] and by the authors in [39]. Both DRL deep mind and SARSA learning algorithms are efficient learners. Due to unique learning behaviour DRL and SARSA tend to receive more rewards. The main advantage of deep mind and online SARSA learning is to handle dynamic control as in [40]. With the development of such type of RL techniques, the challenges for NOMA systems, which are difficult to be solved via traditional optimization methods, have been reinvestigated via RL-based approaches [41]–[43].

3) Motivations: Combining multi-user relationship and resource allocation increases the complexity of NOMA-IoT systems, which also introduces new problems for optimizing power allocation and scheduling schemes. Unlike traditional methods [21], where only one BS is considered for small scale network with no inter-cell interference and dynamic user connectivity. The design of schedulers should be in tandem with the large scale dynamic resource allocations and user decoding strategies. Therefore, due to the high complexity of the problem under multi-cell multi-user cases, AI can be a feasible option for the dynamic resource allocation [44]. For large-scale NOMA-IoT networks, an intelligent reinforcement learning (RL) algorithm becomes a promising approach to find the optimal long-term resource allocation strategy. This algorithm should jointly optimize multiple criteria under dynamic network states. In this paper, our main

goal is to address the following research questions:

- **Q1**: In NOMA-IoT networks, how to maximize the long-term sum rates of users for a given network traffic density?
- **Q2**: How does the inter-cell interference affect the long-term sum rates?
- **Q3**: What is the correlation between traffic density, system bandwidth, and the number of clusters in NOMA-IoT networks?

From above as it is known that model-free methods are suitable to address multi-constrained long-term problem online. Therefore, in long-term there is strong correlation of mentioned research questions with general problems of "intermittent connectivity of IoT users (continuously joining and leaving the network), balanced resource allocations (optimal allocations policy for dynamic network settings) and network traffic (as the (Min-Max) number of users competing for the resource blocks)" in wireless networks. Similarly, research Q1 for capacity maximization, research Q2 for network scalability and, research Q3 for long-term network performance are strongly dependent on the main problems "balancing of network resources, IoT users and, the dynamic network behaviour".

B. Contributions and Organization

This paper considers uplink NOMA-IoT networks, where multiple IoT users are allowed to share the same RB based on NOMA techniques. With the aid of RL methods, we propose a multi-constrained clustering solution to optimize the resource allocation among IoT users, base stations (BSs), and sub-channels, according to the received power levels of IoT users. Appropriate bandwidth selection for the entire system with different traffic densities is also taken into consideration for enhancing the generality. Our work provides several noteworthy contributions:

- We design a 3D association model free framework for connecting IoT users, BSs, and sub-channels. Based on this framework, we formulate a sum-rate maximization problem with multiple constraints. These constraints consider long-term variables in the proposed NOMA-IoT networks, such as the number of users, channel gains, and transmit power levels. To characterize the dynamic nature (online), at each time slot, these variables are changeable.
- We propose two RL techniques, namely SARSA-learning with $\epsilon - greedy$ and DRL, to solve this long-term optimization problem. SARSA-learning is used for light traffic scenarios to avoid high complexity and memory requirements. Heavy traffic scenarios with

a huge number of variables are studied by DRL, where three different neuron activation mechanisms, namely TanH, Sigmoid, and ReLU, are compared to evaluate the impact of neuron activation on the convergence of the proposed DRL algorithm.

- We design novel 3D state and action spaces to minimise the number of Q-tables for both SARSA and DRL frameworks. The considered action space represent switching between RBs, which is the most efficient strategy for our networks. Based on this adequate Q-table design, DRL is able to converge faster.
- We show that: 1) according to the time-varying environment, resources can be assigned dynamically to IoT users based on our proposed framework; 2) for the proposed model, the learning rate $\alpha = 0.75$ provides the best convergence and data rates; 3) for SARSA and DRL the sum-rate is proportional to the number of users; 4) DRL with the ReLU activation mechanism is more efficient than TanH and Sigmoid; and 5) IoT networks with NOMA provide better system throughput than those with OMA.

The rest of the paper is organised as follows: In Section II, the system model for the proposed NOMA-IoT networks is presented. In Section III, SARSA-learning and DRL-based resource allocation is investigated. The corresponding algorithms are also presented. Finally, numerical results and conclusions are drawn in Section IV and Section V, respectively.

TABLE I
TABLE OF NOTATIONS

Symbol	Definition
N_b, b_i	Number of BSs, symbol of BSs
N_s, s_j	Number of sub-channels (NOMA clusters), symbol of sub-channels (NOMA clusters)
N_u, u_k	Number of users, symbol of users
Φ_b	Set of BSs
Φ_s	Set of sub-channels (NOMA clusters)
$\Phi_u^{i,j}$,$u_k^{i,j}$	Set of users connected to BS b_i via sub-channel s_j, user k in the set $\Phi_u^{i,j}$
$c_k^{i,j}(t)$	Clustering variable for user u_k connecting to BS b_i via sub-channel s_j at time t
$p_k^{i,j}(t)$	Transmit power for user $u_k^{i,j}$ at time t
$g_k^{i,j}(t)$	Channel gain for user $u_k^{i,j}$ at time t
$\sigma(t)$	Additive white Gaussian noise at time t
$I_{inter}(t)$	Inter-cell interference at time t
$\gamma_k^{i,j}(t)$	Instantaneous SINR for user $u_k^{i,j}$ at time t
$R_k^{i,j}(t)$	Instantaneous data rate for user $u_k^{i,j}$ at time t
R_k^{th}	Rate requirement for the SIC process of user $u_k^{i,j}$
U_s, P_s	Maximal load of each sub-channel, Maximal power for each sub-channel
T	Duration of the considered long-term communication
$\mathbf{C}$, $\mathbf{P}$	Matrix for clustering parameters, matrix for transmit power
$\theta_\mathbf{t}$	Vector for DRL gradients
β_1, β_2	Moment estimation decay rate

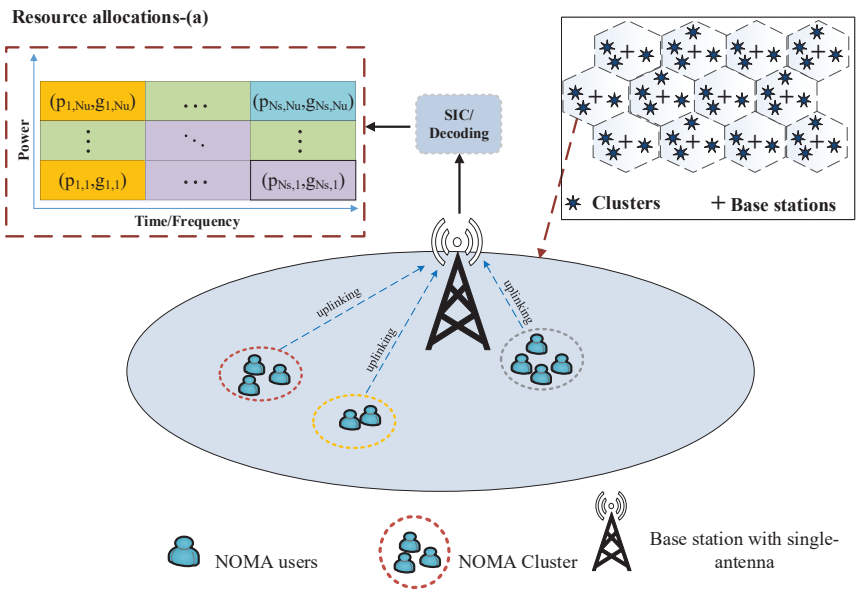


Fig. 1. Illustrating uplink NOMA resource allocation by using the optimization algorithm to efficiently cluster users for resource blocks at the base-station side. Resource allocations-(a) presents different resource blocks in yellow, green, and blue with power on (x-axis) and time/frequency on (y-axis) assigned to IoT users. The powers and gains of N_u users are denoted with p and g.

II. System Model

In this paper, we consider an uplink IoT network with NOMA techniques as shown in Fig. 1, where N_b BSs communicate with $N_u(t)$ IoT users via N_s orthogonal sub-channels. we assume $N_u(t)$ dynamic in each time-slot in our model, however for simplicity we omit (t) for further sections. Additionally, channel gains are also dynamic for each user at each time-slot, even for the same user. The BSs and sub-channels are indexed by sets $\Phi_b = \{b_1, ..., b_{N_b}\}$ and $\Phi_s = \{s_1, ..., s_{N_s}\}$, respectively. Regarding users, the set for users severed by one BS $b_i \in \Phi_b$ $(i \in [1, N_b])$ through a sub-channel $s_j \in \Phi_s$ $(j \in [1, N_s])$ is defined as $\Phi_u^{i,j} = \{u_1, ..., u_{N_u^{i,j}}\}$, where $N_u^{i,j}$ is the number of the intra-set users and $\sum_{i=1}^{N_b} \sum_{j=1}^{N_s} N_u^{i,j} = N_u$. BSs and users are assumed to be equipped with a single antenna. For each BS, the entire bandwidth B is equally divided into N_s sub-channels and hence each sub-channel has $\frac{B}{N_s}$ bandwidth. In a time slot, we assume a part of users are active and the rest users keep silence. To share knowledge, we consider fiber link with ideal back-haul for inter BS connectivity. The defined notations in this system model are listed in TABLE I.

A. NOMA Clusters

Based on the principles of NOMA, more than two users can be served in the same resource block (time/frequency), which forms a NOMA cluster. In this paper, each sub-channel represents one NOMA cluster and $N_u^{i,j} \geq 2$ [45]. To simplify the analysis, we assume BSs contain perfect CSI of all users. That CSI is our state space showing signalling and the channel conditions of IoT users connected to sub-channel via base-station. Detailed explanation is present in the section III-b and section III-c. Based on such CSI, BSs are capable to dynamically optimize the sub-channel allocation for active users in a long-term communication. For an arbitrary user u_k, we define its clustering variable at time t as follows:

$$c_k^{i,j}(t) = \begin{cases} 1, & \text{user } u_k \text{ connects to BS } b_i \text{ via sub-channel } s_j \\ 0, & \text{otherwise} \end{cases}. \tag{1}$$

It is worth noting that $c_k^{i,j}(t)$ also implies the activity status of users. If user k is inactive, we obtain that $c_k^{i,j}(t) \equiv 0, \forall i, j$. The set of clustering parameters is defined as $\mathbf{C}_t$ and $c_k^{i,j}(t) \in \mathbf{C}_t, \forall i, j, k$.

B. Signal Model

In a NOMA cluster s_j, one BS b_i first receives the superposed messages from the active users in $\Phi_u^{i,j}$ and then applies SIC to sequentially decode each user's signal [46]. Without loss of generality, we assume the order of channel gains is $g_1^{i,j} \leq g_2^{i,j}, ..., \leq g_{u_{N_u^{i,j}}}^{i,j}$, where $g_k^{i,j}$ is the channel gain for the k-th user in $\Phi_u^{i,j}$ [47]. Therefore, the decoding order in this paper is the reverse of the channel gain order [48]. In a time slot t, the instantaneous signal-to-interference-plus-noise ratio (SINR) for the intra-cluster user $u_k^{i,j} \in \Phi_u^{i,j}$ is given by

$$\gamma_k^{i,j}(t) = \frac{c_k^{i,j}(t) p_k^{i,j}(t) g_k^{i,j}(t)}{\sum\limits_{k'=1}^{k-1} c_{k'}^{i,j}(t) p_{k'}^{i,j}(t) g_{k'}^{i,j}(t) + I_{inter}(t) + \sigma^2(t)}, \tag{2}$$

where

$$I_{inter}(t) = \sum_{i' \in \Phi_b \backslash b_i} \sum_{k' \in \Phi_u^{i',j}} c_{k'}^{i',j}(t) p_{k'}^{i',j}(t) g_{k'}^{i',j}(t) \tag{3}$$

and $p_k^{i,j}(t)$ is the transmit power of the user $u_k^{i,j}(t)$ and the set of transmit power is given by $\mathbf{P}_t$ $(p_k^{i,j}(t) \in \mathbf{P}_t, \forall i, j, k)$ [49]. The power of thermal noise obeys $\sigma^2(t) = k_b T_r B$, where T_r is temperature of resistors k_b is Boltsmann's constant, B is the considered bandwidth. In this

paper we use $T_t = 300$ K therefore, $\sigma^2(t) \approx 4.14 \times 10^{12}\, BW$. The $I_{inter}(t)$ represents the inter-cell interference, which is generated by the active users served by other BSs using the same sub-channel s_j.

In uplink NOMA, the decoding of user $u_k^{i,j}$ is based on the SIC process of its previous user $u_{k+1}^{i,j}$. If the data rate of successfully completing the SIC process is R_{k+1}^{th}, when the decoding rate of user $u_{k+1}^{i,j}$ obeys

$$R_{k+1}^{i,j}(t) = \frac{B}{N_s} \log_2 \left(1 + \gamma_{k+1}^{i,j}(t)\right) \geq R_{k+1}^{th}, \tag{4}$$

the data rate of user $u_k^{i,j}$ is given by

$$R_k^{i,j}(t) = \frac{B}{N_s} \log_2 \left(1 + \gamma_k^{i,j}(t)\right). \tag{5}$$

Otherwise, if $R_{k+1}^{i,j}(t) < R_{k+1}^{th}$, the decoding of all rest users $u_k^{i,j}, ..., u_1^{i,j}$ fails, namely $R_k^{i,j}(t) = ... = R_1^{i,j}(t) \equiv 0$.

C. Problem Formulation

For a long-term communication with period T, the number of active users is different across each time slot. Given the maximal load of each sub-channel U_s, we assume the number of active users are uniformly distributed in the range $[2, U_s N_b N_s]$ and $U_s N_b N_s \leq N_u$. Under this condition, the average long-term sum rate can be maximized by optimizing clustering parameters $\mathbf{C} = \{\mathbf{C}_1, ..., \mathbf{C}_T\}$ and transmit power $\mathbf{P} = \{\mathbf{P}_1, ..., \mathbf{P}_T\}$. Therefore, the objective function is given by

$$\max_{\mathbf{C},\mathbf{P}} \quad \frac{B}{N_s} \mathbb{E}\left[\sum_{t=1}^{T} \sum_{i=1}^{N_b} \sum_{j=1}^{N_s} \sum_{k=1}^{N_u^{i,j}} \log_2\left(1 + \gamma_k^{i,j}(t)\right)\right], \tag{6a}$$

$$\text{s.t}: \quad g_1^{i,j} \leq, ..., \leq g_{N_u^{i,j}}^{i,j}, \ \forall i, j, t, \tag{6b}$$

$$\sum_{k=1}^{N_u^{i,j}} c_k^{i,j}(t) p_k^{i,j}(t) \leq P_s, \forall i, j, t, \tag{6c}$$

$$\gamma_k^{i,j}(t) \geq 2^{R_k^{th} N_s / B} - 1, \ \forall k, t, \tag{6d}$$

$$2 \leq \sum_{i=1}^{N_b} \sum_{j=1}^{N_s} \sum_{k=1}^{N_u^{i,j}} c_k^{i,j}(t) \leq N_u, \ \forall t, \tag{6e}$$

$$\sum_{k=1}^{N_u^{i,j}} c_k^{i,j}(t) \leq U_s,\ \forall i,j,t \tag{6f}$$

$$\sum_{i=1}^{N_b}\sum_{j=1}^{N_s} c_k^{i,j}(t) \in \{1,0\}\ \forall k,t, \tag{6g}$$

where (6b) is the ordered channel gains based on the perfect CSI. (6c) is to impose the power constraint of each sub-channel. (6d) ensures all clustered IoT users can be successfully decoded for maximizing the connectivity.(6e) and (6f) limits the number of clustered users for the entire system and each sub-channel, respectively. (6g) indicates that each user belongs to only one cluster. Problem (6a) is an NP-hard problem, even only fixed number of users per cluster is considered instead of dynamic range, especially, in case of (6c) and (6f). The proof process is provided in Appendix A. The proof of (6a) follows the idea in [50] and [51].

III. Intelligent Resource Allocation

A. Markov Decision Process Model for Uplink NOMA

In this section, we formulate user clustering and optimal resource allocation for uplink NOMA as a Markov decision process (MDP) problem. Problem transformations are shown in Fig. 2(a) and Fig. 2(b). A general MDP problem contains single or multiple agents, environment, states, actions, rewards, and policies. The process starts with the interaction of an agent with a given environment. In each interaction, the agent processes an action followed by a policy π with previous state s. After processing action according to these conditions and observed state agent/s receives a reward r in the form of feedback to change its state s^t to next state s^{t+1}. A reward can be positive (reward) or negative (penalty). It helps the agent/s to find an optimal set of actions to maximize the cumulative reward for all interactions. Q-table acts as the brain of an agent. The main function of Q-table is to store/memorize states s and corresponding actions a that the agent can take according to all the states as $Q_\pi^T(s,a)$ during trail T for the basic RL algorithms. SARSA and DRL are two promising RL methods to solve this MDP problem. SARSA learns the safest path, the policy π' is learned by estimation of state-value optimization function $Q'(s,a) = Q_\pi(s,a), \forall s,a$, but it requires more memory for complex state space. DRL uses neural network to simplify the Q-table by reducing memory requirements to handle more complex types of problems. Therefore, this work implements SARSA learning for light traffic. To further reduce the impact of state-space complexity DRL is used for heavy traffic scenarios.

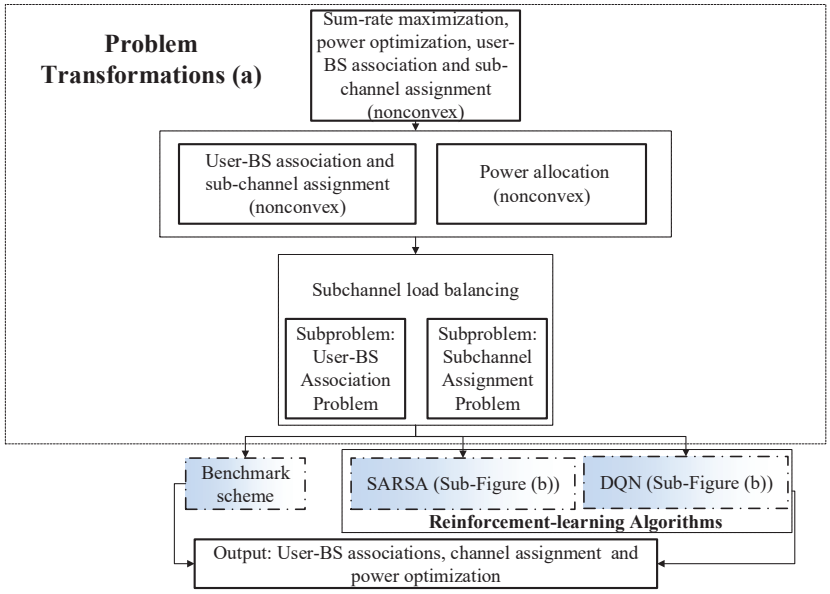
Additionally, in any case when SARSA algorithm fails to provide optimal policy for any type of network traffic during threshold trial T_e then final allocation is done using DRL.

Finally to summarize, this model follows model free on policy SARSA-learning algorithm instead of value iteration and off-policy methods for light traffic and DRL for complex networks. The major advantage of proposed algorithms is to avoid huge memory requirements (DRL) and learn the safest allocation policy (SARSA) for the different traffic conditions.

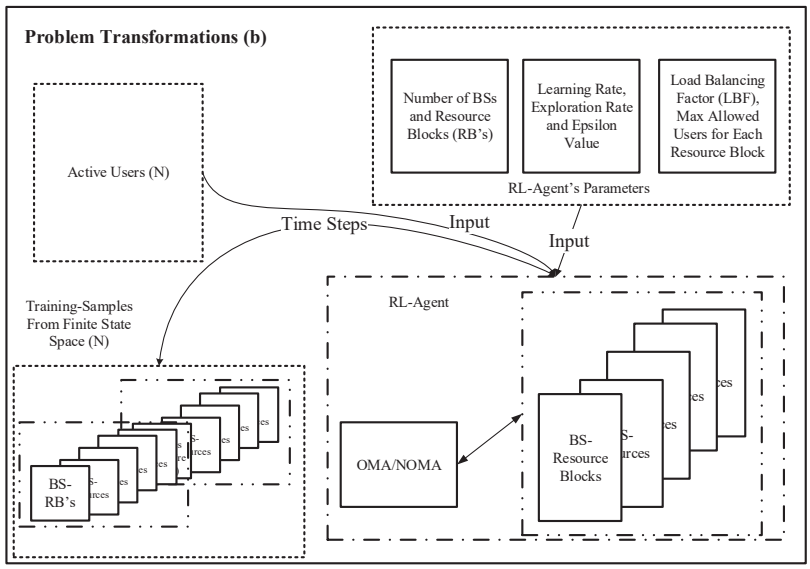
B. SARSA-Learning Based Optimization For Light Traffic ((2-3,2-4)-Ue's)

As the name suggests, for this type of traffic scenarios there is less number of users joining and leaving the network. In other words the state space is not huge as compare to heavy traffic. Therefore, we use SARSA learning algorithm to find optimal long term policy. The traditional Q-learning is not suitable for long term because it uses tuple of 3 (S_t, A_t, R_t) for policy learning which doesn't know the knowledge of next step that is not suitable for our case. Secondly, the state space is not huge as compare to heavy traffic that require more complex control. To efficiently utilise system resources we use SARSA learning for light traffic and DRL for heavy traffic where the state space is huge with dynamic users. For SARSA learning, discount factor γ, sum reward, and the number of iterations are significant hyper parameters. The details for the flow of the information update is shown in Fig. 3. The 5-tuple ($\mathbf{S}$, $\mathbf{A}$, $\mathbf{P}$, $\mathbf{R}$, $\mathbf{S}'$, $\mathbf{A}'$) SARSA-learning elements are mentioned below:

1) $\mathbf{S}$, is a state space consists of finite set having dimensions $N_b \times N_s$ containing $N_u^{N_b \times N_s}$ total number of states. Each state represents one sub-set of 3D associations among users, BSs, and sub-channels.
2) $\mathbf{A}$, is an action space consists of a finite set of actions to move the agent in a specific environment. Actions in this model are $[-1, 0, +1]$. The '-1' is to reduce any one of the state elements from state matrix. Similarly, '+1' shows an increment in any of the state matrix elements. The last action '0' represents no change in the current state of the agent (BSs). It means that actions are swap operations between sub-channels and all BSs. For example, when an agent takes an action from (7), the first action in $\mathbf{A}$ means agent performs swap operation of user between sub-channels at BS. In this model, agents have total 8 swap operations between BSs and sub-channels.



(a)



(b)

Fig. 2. Overview for the proposed framework to the sum-rate maximization problem. Sub-figure (a) is an optimization problem breakdown to show where RL algorithms are applied and Sub-figure (b) shows problem transformations for the users and BSs as system states and the brain of reinforcement learning agents, respectively.

$$\mathbf{A} = \left\{ \begin{pmatrix} -1 & 0 \\ 1 & 0 \end{pmatrix}, \begin{pmatrix} 1 & 0 \\ -1 & 0 \end{pmatrix}, \begin{pmatrix} 0 & -1 \\ 0 & 1 \end{pmatrix}, \begin{pmatrix} 0 & 1 \\ 0 & -1 \end{pmatrix}, \begin{pmatrix} 0 & 0 \\ 1 & -1 \end{pmatrix}, \begin{pmatrix} 0 & 0 \\ -1 & 1 \end{pmatrix}, \begin{pmatrix} 1 & -1 \\ 0 & 0 \end{pmatrix}, \begin{pmatrix} -1 & 1 \\ 0 & 0 \end{pmatrix} \right\}. \tag{7}$$

3) $\mathbf{P}$, is an expected probability $P^{a}_{s \to s'} = Pr(s'|s,a)$ to change current state s into next state s' by taking action a. The total number of actions for an agent are $(2 \times N_b \times N_s + 1)$ with '8' swap operations. These operations include '+1','-1', and '0' actions, the agent selects suitable actions according to corresponding state to obtain an optimal state and action pair.

4) $\mathbf{R}$, is a finite set of rewards, where the reward obtained after state s transition to next state s' by taking action a. Reward function is denoted by $r^{a}_{s \to s'}$, showing that in result of all associations the agent will receive reward according to the conditions mentioned in reward function.

5) Multi-constrained reward function, the short-term reward in proposed model depends on two conditions:1) sum-rate and 2) the state of the system means the total number of users associated to BSs and sub-channels, which is defined as $\mathbf{S}'$. The reward function can be expressed as follows:

$$r(s_t, s_{t+2}, a_t) = \begin{cases} r = 0, & if\ R_{s_{t+1}} \geq R_{s_t} \text{ and } \sum_{t=1}^{T_e}(u_k^{st}) = \sum_{t=1}^{T_e}(u_k^{s_{t+1}}) \\ r = -10, & \text{otherwise.} \end{cases} \tag{8}$$

6) $\mathbf{S}'$, is a next state of an agent based on the previous state, action, and reward pairs of an agent.

7) $\mathbf{A}'$, is a next possible action can be taken by an agent from state $\mathbf{S}'$.

Definition 1. *The parameters of 3D state matrix* $\mathbf{S}$ *defined as* $Z = \{1, 2, \cdots, N_u^{N_b \times N_s}\}$ *total number of states with* $N_b \times N_s$ *dimensions* $\sum_1^Z \mathbf{S}_{N_s}^{N_b}$. *For all types of network traffic minimum for* Z *is defined as* $\sum_1^N Z_{(ij)} \geq 2$, *the maximum for light traffic is* $\sum_1^N Z_{(ij)} \leq \{3, 4\}$ *and for heavy network traffic the maximum load is* $\sum_1^N Z_{(ij)} \leq 10$.

Furthermore, the optimal policy of the aforementioned parameters can be discovered by an agent using following function:

$$\pi'(s) = \arg\max_{a} Q'(s,a), \forall s \in \mathbf{S}', \tag{9}$$

where $\pi'(s)$ represents the optimal policy. This function provides the optimal policy value for each state s from the finite sate set after taking appropriate action a. For a better understanding, the optimal policy can be defined:

$$V_{\pi'}(s_t) = \max_{a} \left[r(s_t, a_t) + \gamma \sum_{s'} P_{s \to s'} V_{\pi'}(s') \right]. \tag{10}$$

For Q-table value updating that contains state and corresponding action values of an agent. Bellmen equation is utilised to perform optimization processes. According to Bellman equation statement, there is only one optimal solution strategy for each environment setting. Bellman's equation is defined as:

$$Q(s,a) \leftarrow (1-a)Q(s,a) + \alpha \left[r' + \gamma Q(s',a') \right], \tag{11}$$

where $\gamma \in (0,1)$ is a discount factor, which is a balancing factor between historical and future Q-table values. The larger γ is the more weight for the future value and vice versa. $\alpha \in (0,1)$ indicates learning rate, it works like a step function (i.e., larger α contributes to fast learning but due to minimal experience, it may result in non-convergence. Similarly, if the value of α is too small then it will increase the time complexity of the system by leading it to a slow learning process).

Definition 2. *For Q-learning we define* $Q_{t=0}(s,a) = -100$ *to learn greedy policy* $\mathbb{P}_{\pi}(\mathbf{A} = a | \mathbf{S} = s)$ *for all state and action pairs.*

One main limitation of reinforcement learning algorithms is slow convergence due to $Q(s,a)$ requirement. Additionally, it is challenging with 3D state space and dynamic systems [52], [53]. Due to dynamic behaviour of IoT users the 3D state $\mathbf{S}$ and action space $\mathbf{A}$ influence learning process more as $\mathbf{S}$ and $\mathbf{A}$ are main parts of Q-table $Q(s,a)$. The convergence of the reward functions r and reinforcement learning hyper parameters guide the algorithm towards optimal policy V. In other words the choice of reward function and the values for $\{\epsilon, \alpha$ *and,* $\lambda\}$ are used by reinforcement learning agent/s to avoid the random walk. The random walk in search space

cause infinite exploration of the search space resulting no convergence. Therefore, we are able to propose the following conclusion.

Remark 1. *The selection of suitable rewards $r^{a}_{s\to s'}$ according to system dynamics $Q(s,a)$ is critical for effective convergence to find optimal $V_{\pi'}$. Consequently, following* (10) *altering the reward function does not change the output of RL algorithms but the convergence towards policy $V_{\pi'}$ is highly influenced.*

It is known that the proposed protocols are capable to handle multi-constrained optimization problems for different network traffic scenarios. We used $\epsilon - greedy$ SARSA-learning and DRL algorithms to explore and exploit search space to find dynamic outcomes, so the proposed protocols are capable to successfully obtain the optimal clustering solution. The Q-table in our model contains solutions for all subsets (user associations) in the search space. Therefore, in each episode N_e, only a specific subset of users will be active.

Remark 2. *In reinforcement learning to find the best associations s_t from the set $S_t = \{s_1, s_2, \cdots, s_N\}$ possible states, an agent will converge towards the optimal states and actions pairing with the*

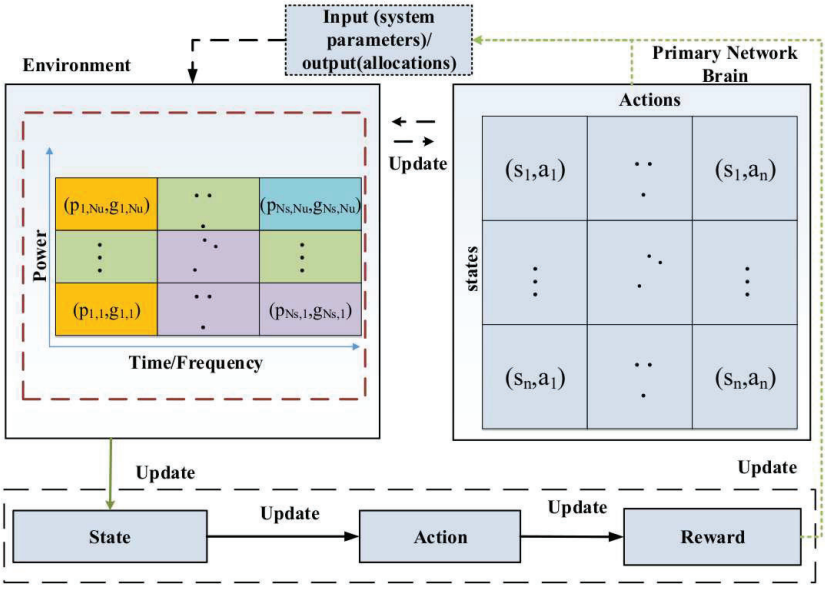


Fig. 3. An illustration of the communication environment for proposed algorithm, where RL technique (SARSA) is invoked to optimize NOMA-IoT uplink 3D associations and resource allocation. The agents in this case are the BSs. The process of associations and resource allocation based on users activities is the state for our system.

highest probability $\mathbf{P}_{\pi'}$*. In this way, by the increase in probabilities, the number of visits per state-action pair and rewards increase as well.*

Since an agent has limited successful visits, the achieved rewards will be as described in **Remark 1** and **Remark 2**. As a result, the agent successfully finds the optimal policy for the given system by processing best actions.

1) SARSA-algorithm: Based on the above discussions, we design **Algorithm 1** for step by step significant optimization stages of the SARSA algorithm for light traffic networks. The details of the mentioned algorithm are as follows:

Algorithm 1 SARSA-Learning Based NOMA-IoT Uplink Resource Optimization

1: Inputs for SARSA:
   1) Episodes N_e
   2) Explorations per trials T_e
   3) Learning rate α
2: Initialization for SARSA:
   1) Network parameters $(N_b, b_i, N_s, s_j, N_u, u_k, P_b)$
   2) Q-Table $Q(s,a)$
3: Define number of clusters-k
4: Define range of users per cluster
5: load $\mathbf{s} = N_u^{N_b \times N_s}$ and $\mathbf{a} = [-1, 0, +1]$
6: Random user association to any BSs and Cluster
7: **for** iteration = 1:N_e **do**
8: $s_t = rand()$
9: **for** iteration = 1:T_e **do**
10: s_t, a_t
11: compute $r(s_t, s_{t+1}, a_t) = \begin{cases} r = 0, & if\ R_{s_{t+1}} \geq R_{s_t} \text{ and } sum(u_k^{s_t}) == sum(u_k^{s_{t+1}}) \\ r = -10, & \text{otherwise.} \end{cases}$
12: update R_{sum}
13: update $Q(s,a) \leftarrow (1-a)Q(s,a) + \alpha\left[r' + \gamma Q(s',a')\right].$
14: Update π towards greediness
15: $s \leftarrow s', a \leftarrow a'$
16: **end for**
17: return optimised **(c,p)** (6) under constraints (6a),(6b),(6c),(6d) and (6e)
18: **end for**
19: Return Q-Table $Q(s,a)$

- Line $\#(1-7)$: presents initialization of the SARSA algorithm, in which the system is initialized by initial sets of users, BSs, and sub-channels as an initial state S_t. After this we define the maximum number of clusters and the maximum number of users for each cluster. In line$\#4$ the brain of an agent is initialized with -100 having dimensions $[s \times a]$ as Q-table. The purpose of initialization with -100 is to show that the brain of an agent needs training. Therefore, after training, the Q-table will contain values approaching to zero for the best case and vice versa. Secondly, it also shows that the proposed algorithm is targeted to solve the maximization problem, maximum Q-value means better solutions.

Line $\#(5-6)$ shows SARSA-learning parameter definition and initial random association among IoT users, BSs, and sub-channels.

- Line $\#(7-17)$: shows key training steps based on Q-table updates via bellman equations. From line$\#1$, an agent performs actions according to a given state of the environment, that is 3D associations and cluster allocation. In line$\#8$ agent picks new associations for different active users in one episode, then for all trials agent is trying to get optimal associations with optimal sum rate, if the associations are successful then the agent gets a reward (0) and if it fails then negative (-10) is given as a punishment. In line$\#13$, based on the 3D designed 5-tuple ($\mathbf{S}$, $\mathbf{A}$, $\mathbf{P}$, $\mathbf{R}$, $\mathbf{S}'$, $\mathbf{A}'$) values (11) is updated on-line. To perform online updates using $\mathbf{S}$, $\mathbf{A}$, $\mathbf{P}$, $\mathbf{R}$, $\mathbf{S}'$, $\mathbf{A}'$ instead of $\mathbf{S}$, $\mathbf{A}$, $\mathbf{P}$, $\mathbf{R}$ as (traditional Q-learning) the online learning mechanism becomes more fast converging. In other words, the agent finds optimal long-term online allocation policy more efficiently. Similarly, these updates are calculated for maximum episode $N_e = 500$ and all the trials $T_e = 500$ to maximize the overall long-term average reward of the system.

Definition 3. *In 3D state matrix* $\mathbf{S}$ *from the set* $S_t = \{s_1, s_2, \cdots, s_N\}$ *possible states, is defined as CSI of the proposed network that is known to both of the reinforcement learning agents. Therefore, the reinforcement learning agent contains perfect knowledge of the CSI for the whole network.*

C. Deep Reinforcement Learning For Heavy Traffic ((2-10)Ue's)

In general, both on-line and off-line Q-learning methods require high memory space to build a state of the systems. However, practical systems are high in dimension and complex. Due to this reason Q-learning is not suitable for a large action space, this is a major drawback of conventional Q-learning methods. To overcome this, DRL method adopts a deep neural network (DNN) $Q(s, a; \theta)$, to generate its Q-table with the help of θ by approximating the Q-values $Q(s, a)$ [38]. Therefore, DRL agents only need to memorize the θ weights instead of reserving huge memory space for all possible states and action pairs. This is the main advantage to use DNN. More specifically, in conventional Q-leaning algorithms, the optimization of $Q(s, a)$ is equal to the optimization of $Q(s, a; \theta)$ in DRL with low memory requirements. Similarly, θ updates are based on history states, actions, and reward values. More specifically, these values

are based on DRL agent interactions with the environment to learns the relationship among the different actions and states by continuously observing given environment.

1) $\mathbf{S}$, is a unique state space used as an input of DNN. Each state is a combination of multiple sub-sets of 3D associations among users, BSs, and sub-channels. It also consists of current rewards of the system as instantaneous and average reward from previous iterations.
2) $\mathbf{R}$, is a reward of the system that is denoted by $\mathbf{R} = \{r_i, r_l\}$, where r_i is an instantaneous rewards similar to SARSA algorithm and $r_l = \sum_{t=1}^{500} r_i^t/t$ denotes long-term average rewards for the time slot t.
3) $\mathbf{A}$, is a multi-dimensional matrix representing actions as $\mathbf{A} = \{\mathbf{a_1}, \mathbf{a_2}, \cdots, \mathbf{a_8}\}$. For the DRL algorithm, the action mechanism is based on two main parts as; allocation strategies described as switching strategy $\mathbf{a_s}$ and association strategy $\mathbf{a_i}$ for the optimization process, where $\mathbf{a_s}$ is a switching mechanism similar to SARSA and used for the DRL channel switching process. The second strategy $\mathbf{a_i}$ is a result of selected switching strategy $\mathbf{a_s}$, $\mathbf{a_i}$ denotes an index of the 3D associations among users, BSs, and sub-channels. Finally, the DRL agent uses loss function mentioned in (12) to calculate θ based on the previous experience.

$$loss(\theta) = 1/N_e \sum_{t=1}^{N_e} [y_t^{DRL} - \boldsymbol{Q}(s_t, a_t; \theta)]^2, \tag{12}$$

where

$$y_t^{DRL} = r + \gamma \max_{a' \in \boldsymbol{A}} \boldsymbol{Q}(s', a'; \theta') \tag{13}$$

and y_t^{DRL} is the target Q-values from target DNN. For the improved training, in general the update frequency of the target network θ' is performed in slow manner. Due to this reason the target network remains fixed for the target network update threshold T_e'.

The DRL agent uses gradient decent method as in (12) to reduce the prediction error by minimizing the loss function. The updating of θ is provided in (16), which is based on the outcome of new experience. The updating function for θ is defined in (18), namely DRL Bellman

equation.

$$\theta \leftarrow \theta - [y' - \boldsymbol{Q}(s,a;\theta)]\nabla \boldsymbol{Q}(s,a;\theta). \tag{14}$$

$$q_\pi(s,a) = r(s,a) + \gamma \sum_{s' \in \boldsymbol{S}} \sum_{a' \in \boldsymbol{A}} p_{ss'}(a) q_\pi(s',a'), \tag{15}$$

$$q_{\pi^*}(s,a) = r(s,a) + \gamma \sum_{s' \in \boldsymbol{S}} p_{ss'}(a) \max_{a' \in \boldsymbol{A}} q_{\pi^*}(s',a'), \tag{16}$$

where the function $q_{\pi^*}(s,a)$ shows Q-values and the long-term reward calculations for DRL based on the discount factor γ and below mentioned optimal DRL policy π^*.

$$\pi^*(s) = \arg\max_{a \in \boldsymbol{A}} \left[q_{\pi^*}(s,a) \right], \ \forall \ s \in \boldsymbol{S}, \tag{17}$$

where $\pi^*(s)$ represents the optimal policy for the DRL algorithm. This function provides the optimal policy value for each state s from finite sate set after taking appropriate action a.

$$\boldsymbol{Q}(s,a) \leftarrow (1-\alpha)\boldsymbol{Q}(s,a) + \alpha \left[r(s,a) + \gamma \max_{a' \in \boldsymbol{A}} \boldsymbol{Q}(s',a') \right], \tag{18}$$

where $\boldsymbol{Q}(s,a)$ is showing Q-value update according to DRL Bellman equation.

$$s(t) = \{a_1(r_i^1, r_l^1), a_2(r_i^2, r_l^2) \ldots, a_t(r_i^t, r_l^t)\}. \tag{19}$$

$$\rho(x) : \sum_{j=0}^{d_j} W_j \times I_j(s(t)) + \psi_j, \tag{20}$$

where in (19), $s(t)$ represents state of the DRL agent and equation (20) shows the activation mechanism for each neuron layer I based on weights W_j for $j-th$ depth of neurons with bias term ψ_j. In this model, the input of the DRL algorithm is the instantaneous network observation as s_t. This state is sent to the different neural network neurons with specific network W_j to obtain the final output as a set of different Q-values for all actions. For the DRL framework the size of output actions are similar to the SARSA. We use the replay memory as an experience for the DRL agent to store the tuple $(s_t, a_t, r_t, s^{'})$ for all the time steps T_e in an experience dataset E with size ε. When the size ε is full, the first experience as an oldest tuple will be removed to free some space for the new experience update. The reason for this updated experience is to reflect the sequential exploration of the DRL framework. However, the samples distribution is independent and identical. Therefore, to get more general output, the W_j update process is performed on the basis of randomly sampled tuple $(s_t, a_t, r_t, s^{'})$ instead of the current tuple. This

is because output is highly influenced by the correlated set of tuples $(s_t, a_t, r_t, s^{'})$ and variance of the updates.

Definition 4. *DRL design in this work is defined with two main elements, the first element is target Q-network based on $\theta^{'}$. The second main element of DRL is state transition mechanism $(s_t, a_t, r_t, s^{'})_{t\in[n]}$. This mechanism is used to construct mini-batch for experience reply from dataset E to train DNN.*

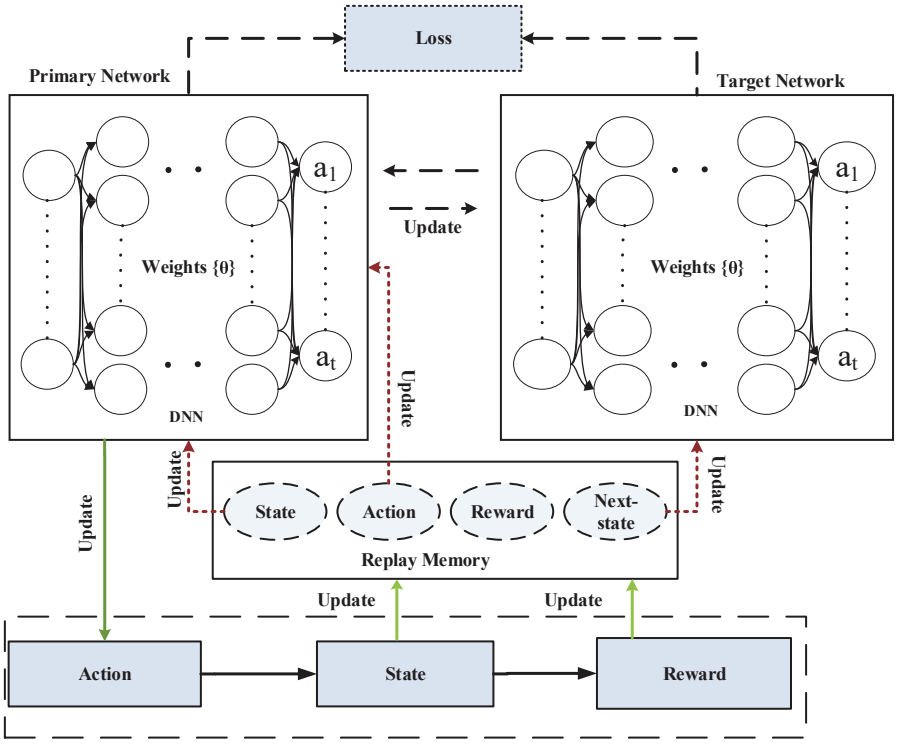


Fig. 4. DRL structure: it shows the flow of information between target and training networks to minimize the loss function using states, actions, rewards, and replay memory.

Remark 3. *The convergence rate/speed of the proposed algorithm varies according to the initial 3D association (states) that is randomly selected. In this model, the state space means allocation strategies that include subsets of all possible associations of active users $u_k = 2 \leq N_u$ for each sub-channel at the episode N_e.*

Based on the above discussions, we design **Algorithm 2** for step by step significant optimization stages of the DRL algorithm for heavy traffic. The details of the mentioned algorithm are as follows:

1) DRL:

- Line $\#(1-2)$: In this stage, the parameter initialization is performed, which is a similar initialization step like SARSA. However, instead of state action pairing, the weight matrix is initialized for DRL to find optimal policy π.
- Line $\#(3)$ Pre-training: In this stage, initial actions are selected using uniform random distribution as an initial state space in continuous environment. In this way initial weights are also calculated to start the optimization process.
- Line $\#(4-18)$: Whole process for DRL is similar to SARSA from line $\#(5-11)$ with DRL bellmen equations (12) to (19).
- Loss Calculation: The equation (12) is to calculate the loss θ that is the mean squared error (MSE) indicating the difference of the target and predicted networks. To optimise these values between the target network and prediction network we use Adam optimiser. The Adam optimizer is used for the loss minimization to further improve the optimal predicted Q-values for each episode. Therefore, the DRL framework converges faster even in huge state space. In (13), we calculate the target Q-values based on the tuple (S_t, A_t, R_t) from mini-batch and the mini-batch is updated after 100 iterations.
- DRL Updates: The updating function for the prediction of DRL θ and long-term reward calculation is shown in (14) to (16), where DRL agents obtain rewards and prediction loss after every transition from s_t to next state s_{t+1} to find the greedy policy. Additionally, γ discount factor has significant impact in search of the greedy-policy because based on discount factor as we mentioned in the previous paragraphs, the agent selects immediate or previous Q-values. The policy π is calculated using (17) to maximize the Q-values by greedy search. The calculation for DRL Bellman equation is performed using states in (18).

- Sparse Activations: The $\rho(x)$ is an activation function for DNNs sparse activations using ReLU (ρ). The sparse activations help agents to efficiently converge by avoiding useless neuron activations. The outcome of sparsity is shown in the results section, comparing sparse ReLU, Sigmoid, and TanH. In (20), the activations are performed for the δ_j density of neurons with $j-th$ index, for each neuron we use state of the system as an input that is multiplied with weight W_j of $j-th$ density and adding bias value as λ before activation. In next steps, current states, actions, and rewards are added to mini-batch for experience

replay (for self-training). In $\#(13-15)$, the agent receives next states from mini-batch that is learned in previous sections based on pre-training. Before that, the learning process of the agent is based on pre-training but when mini-batch is full, the agent will learn the optimal policy by experiencing replay mechanism with the help of mini-batch processing.

- Neural Networks: this paper uses the DRL that is built with two DNNs as shown in Fig. 4: 1) a training network $Q(s', a'; \theta)$ that learns the policy and 2) a target network $Q(s', a'; \theta^*)$ to compute target Q-values for every update, where θ and θ^* shows the weights of these two networks. For the training of the DNN network, θ weights are predicted based on the current state and action. On the other hand, θ^* weights are based on the previous episodes and these weights are fixed during the calculation of θ for training purpose. Additionally, We utilize MSE loss function (12) to evaluate the accuracy of the training for the target network. Therefore, the proposed loss function is based on θ and θ^* to check the deviation of the predicted DNN weights.
- Output: Finally, the output of this algorithm is the optimal policy for all clusters where overall long-term sum rate is maximum.

Definition 5. *We use ReLU activation function $\rho(x) = max(x, 0)$ (x is the input neuron) for DRL performance evaluations. A ReLU network of density δ_j and λ_i hidden layers with each layer width $\{\delta_j\}_{i=0}^{\lambda+1} \subseteq \mathbb{N}$ can be represented as $f : \mathbb{R}^{\delta_0} \rightarrow \mathbb{R}^{\delta\Lambda+1}$ for any positive number L.*

$$f(x) = w_{\Lambda+1}\rho(w_\Lambda\rho(w_{\Lambda-1}...\rho(w_2\rho(w_1\rho + \psi_1) + \psi_2)...\psi_{\Lambda-1}) + \psi_\Lambda.$$

In this definition, $f(x)$ is a function to show the construction of neural network with weights for each layer $w_\lambda \in \mathbb{R}_{\delta_\lambda}^{\delta_{\lambda-1}}$ and ρ is the activation for each neuron. The mesh structure of the neural network remains fixed in this model to learn two main neural network parameters $(w_\lambda, \psi_\lambda)_{[\lambda\in\Lambda+1]}$ in addition with the activation function ρ and the input of the neural network. In the neural network Ψ bias terms are added with the input of the DNN as $\Psi_{\Lambda+1}$ as a shift value. To optimise our dynamic objective function, the greedy search agent is used. With the help of greedy search, the DRL agent receives higher rewards.

Remark 4. *To avoid useless visits, greedy policy $\pi^{'}$ provides a balanced exploitation, because $\epsilon - greedy$ exploits in the most cases and some times it processes a random action to explore the environment in search of different solutions $Q_{N_e}(s, a) = \mathbb{E}[\sum_{t=1}^{N_e} \gamma_t r_t]$.*

For DRL, unbalanced random actions cause huge error propagation so that this $\epsilon - greedy$ is suitable to be applied for achieving efficient learning in a dynamic state space. Note that the boundary for the policy selection is $0 \leq \epsilon \leq 1$. For ϵ close to 0 the policy becomes greedy policy, and for ϵ close to 1 the agent explores more.

Algorithm 2 Deep Q-Learning Based NOMA-IoT Uplink Resource Optimization

Inputs for DRL:
1) Episodes N_e
2) Explorations per trials T_e
3) Learning rate α

Initialization for DRL:
1) Network parameters $(N_b, b_i, N_s, s_j, N_u, u_k, P_b)$
2) memory, hidden size, State size, action size and mini-batch

train DRL to find a good policy $Q(W_a)$
for iteration = 1:N_e **do**
  for iteration = 1:T_e **do**
    s_t, a_t
    compute $r(s_t, s_{t+1}, a_t) = \begin{cases} r = 0, & if\ R_{s_{t+1}} \geq R_{s_t} \text{ and } sum(u_k^{s_t}) == sum(u_k^{s_{t+1}}) \\ r = -10, & \text{otherwise.} \end{cases}$
    update $Q(W_a)$ using $q_{\pi^*}(s,a) = r(s,a) + \gamma \sum_{s' \in \boldsymbol{S}} p_{ss'}(a) \max_{a' \in \boldsymbol{A}} q_{\pi^*}(s', a')$.
    $loss(\theta) = [y_t^{DRL} - \boldsymbol{Q_t}(s', a'; \theta)]^2$ update using $y_t^{DRL} = r + \gamma \max_{a' \in \boldsymbol{A}} \boldsymbol{Q}(s', a'; \theta)$.
    $s \leftarrow s', a \leftarrow a'$
    update mini-batch (Experience)
    if $T_e > State - size$ **then**
      get $s \leftarrow s', a \leftarrow a'$ from mini-batch
    end if
  end for
end for
Return $Q(W_a)$

Definition 6. *(Sparsity for ReLU DNN): The sparsity of the ReLU network is a weight based sparsity denoted by κ, sparse ReLU networks are bounded by Ψ for Λ_i layers, $\Psi > 0$. For any hidden layer $\Lambda_i, \kappa \in \mathbb{N}, \{\delta_j\}_{i=0}^{\Lambda+1} \subseteq \mathbb{N}$.*

$$\mathsf{F}(\Lambda, \{\delta_j\}_{i=o}^{\Lambda+1}, \kappa, \Psi) = \left(f : \max_{\lambda \in [\Lambda+1]} \left\| {W_\lambda}' \right\|_\infty \leq 1, \sum_{\lambda=1}^{\Lambda+1} \left\| {W_\lambda}' \right\|_0 \leq \kappa, \max_{j \in [\delta_{\Lambda+1}]} \|f_j\|_\infty \leq \Psi \right),$$

where W_λ' is used to represent W_λ, ψ_λ. The function f is from **Definition 5** and f_j is the $j-th$ element of f.

D. Complexity

The complexity of the proposed model is based on the number of BSs N_b, total number of sub-channels N_s and the number of users communicating N_u. In proposed scheme, simulated experiments are based on different examples. This paper considers $N_b = 2$, $N_s = 2$ and $N_u = 3, 4$, for light traffic and 10 for heavy traffic. These examples are association decisions for the user N_u and the sub-channel N_s at BS N_b that receives signals for $N_u^{N_b}$ channels from users. The computation

complexity for SARSA-learning is $\mathcal{O}(N_b N_u^{N_s})$ operations with $N_u^{N_b \times N_s} \times (2 \times N_b \times N_s + 1)$ memory requirement for Q-table to simulate brain of the learning agent/s. The complexity of DRL is $\mathcal{O}(N_e T_e)$ with smaller Q-table $\mathcal{Q}(W_j)$ and DRL uses 1D experience replay containing states vector (19) instead of huge memory requirements like traditional Q-learning. The benchmark scheme considered in this work is a memory-less method, which shows the maximum achievable rate by exhaustively searching all possible combinations of 3D associations. Consequently, it requires more number of operations. Due to this reason, the computation complexity increases in exponential manner as $\mathcal{O}(N_u^{N_b \times N_s})$.

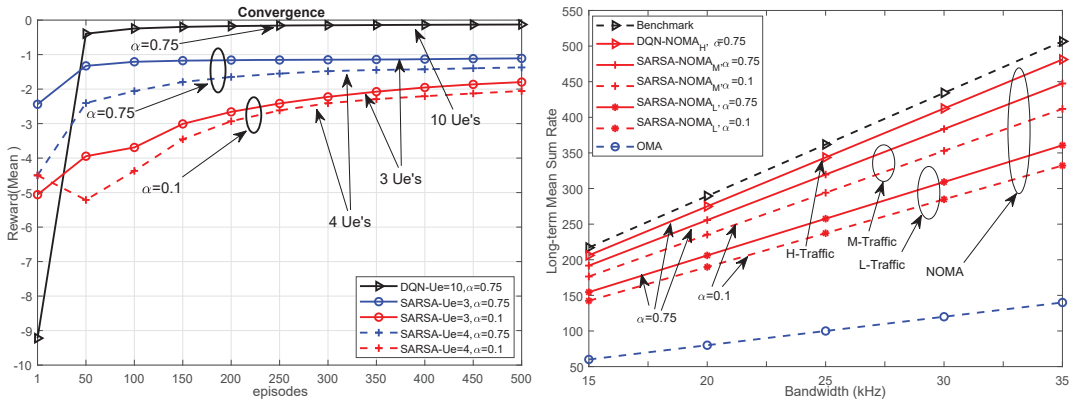
TABLE II
NETWORK PARAMETERS

Load balancing factor k values per resource block	2-3,2-4,2-10
Total number of trials	500
Total number of time steps	500
Bandwidth	$[15-35]kHz$
Gain	$[1, 1.5, 2] * 10^{-5}$ [54]
γ_δ	0.6
α	0.75, 0.1
ϵ	0.1
λ	0.5
Optimisers	SARSA-DRL (Adam)
Deep neural network activations	Sigmoid, TanH, ReLU

IV. NUMERICAL RESULTS

In this section, simulation results are provided for the performance evaluation of the proposed multi-constrained algorithms. The proposed multi-constrained algorithms are tested under different network settings to solve: 3D associations among user, BSs, and sub-channels as well as sum rate optimization with different network traffic. For simulations, we have considered two different traffic density threshold values to analyse the impact of network load with various power levels on the sum rate and 3D associations. Additionally, network load in our case represents load of each resource block instead of total number of users in the network. Therefore, max network load=10 with two RB's for each BSs means $10 * 4 = 40$ users in the network. To show the significance of available channel bandwidth, we start with a minimum channel bandwidth of 60(kHz) and then increase it to 120(kHz) under different network traffic conditions. The hardware and software system used for experimentation is Intel core i7-7700 CPU with 3.60 GHz frequency having 16 GB of RAM (Random Access Memory) and 64-bit operating system (windows-10). All the experiments are simulated using Matlab version-R2019a and Python 3.6. From Table.II for both the algorithms we used 500 episodes with 500 iterations for each episodes. Similarly, $\lambda_{[SARSA/DRL]} = 0.5, \gamma_{[SARSA/DRL]} = 0.6, \alpha_{[SARSA/DRL]} = \{0.75, 0.1\}$, and $\epsilon - greedy$

exploration are values for the significant hyper-parameters of proposed algorithms. We used Load balancing factor k values per resource block to show the maximum and minimum user connectivity for each resource block. The values of channel gain for each user is defined as $[1, 1.5, 2]*10^{-5}$ [54]. For the DRL, additional parameters are trained, such as loss MSE, activation functions, batch-size$= 500$, optimisers, experience memory $E = 500$, pre-training length $= 500$, the number and size of hidden units. We use ReLU, Sigmoid, and TanH as activation functions with two hidden layers having density $\delta = 500$ units. Adam optimizer is utilized for the optimal convergence of DRL.



(a) (b)

Fig. 5. Overview for the proposed framework to the sum-rate maximization problem. Sub-figure (a) is convergence for proposed algorithms: DRL for heavy traffic (max scheduling up to 10 users), SARSA for medium and low traffic range (support 2, 3, 4, and upto 10 users scheduling) and the comparison for two different learning rates ($\alpha = 0.75, \alpha = 0.1$). (b) shows long-term comparison between the channel bandwidth, average sum rate, and power levels for the proposed SARSA, DRL and benchmark scheme.

A. Convergence vs Sum Rate vs Traffic Density

Fig. 5(a) shows the inter-correlations among the four measures of convergence. It is apparent from this figure that if traffic density increases then convergence is slower and vice versa. DRL has better convergence for heavy traffic with the maximum allocation capacity/load, which makes DRL more suitable for the scenarios with high traffic densities. Secondly, another interesting insight is that the performance of SARSA $\alpha = 0$ is better than $\alpha = 0.1$ with $\epsilon - greedy$. The convergence of Adam depends on DRL θ weights as $R_{DRL} = \sum_{t=1}^{T} \big(f_t(\theta_t) - f_t(\theta^{'})\big)$. where $\theta^{'} = \arg\min_{\theta\in\kappa} \sum_{t=1}^{T} f_t(\theta)$ and κ is feasible set for all $t-1$ steps.

Definition 7. *The bounded gradients of the function* f_t^{DRL} *is* $\|\delta f_t(\theta)\|_2 \leq G^{DRL}\|\delta f_t(\theta)\|_\infty \leq G_\infty^{DRL}, \forall\theta \in R_d^{DRL}$. *Secondly, the distance generated by the Adam optimiser is bounded as:*

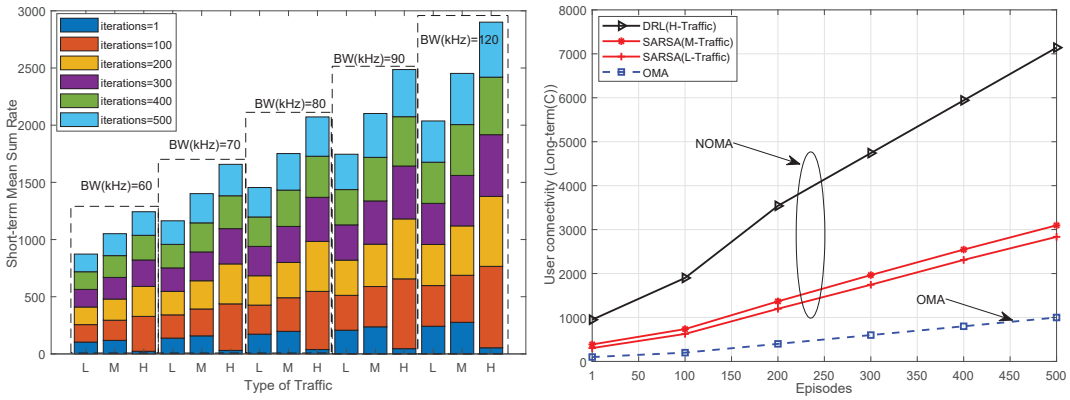
$\|\theta_p - \theta_q\|_2 \leq D, \|\theta_p - \theta_q\|_\infty \leq D_\infty$ for any $p, q \in \{1, \cdots, T\}$ with the bias terms $\beta_1, \beta_2 \in [0, 1]$ satisfying the $\frac{\beta_1^2}{\sqrt{\beta_2}} < 1$ condition. Let the learning rate of the Adam optimiser be $\alpha_t = \alpha/\sqrt{t}$ and bias term $\beta_1^t = \beta_1 \lambda^{t-1}, \lambda \in [0, 1]$ for each step t, for all $T \geq 1$ Adam obtains the following condition [55]:

$$R_{DRL}(T) \leq D^2/2\alpha(1-\beta_1) \sum_{i=0}^{d} \sqrt{T v'_{T,i}} + \frac{\alpha(1+\beta_1) G_\infty^{DRL}}{(1-\beta_1)\sqrt{1-\beta_2}(1-\gamma)^2} \sum_{i=0}^{d} \|g_{1:T,i}\|_2 + \sum_{i=0}^{d} \frac{D_\infty^2 G_\infty^{DRL} \sqrt{1-\beta_2}}{2\alpha\ (1-\beta_1)(1-\lambda)^2} \tag{21}$$

The results obtained from the primary analysis of sum rate and traffic densities are shown in Fig. 5(b) in long-term settings, it is clearly visible that the proposed model performs close to the benchmark scheme and better than OMA.Fig. 6(a) shows short-term performance analysis between sum rates, bandwidth, and the number of iterations. This figure illustrates the performance of DRL and SARSA according to different bandwidths, where the performance of the DRL is better than SARSA. Interestingly it also shows that as the traffic density increases the sum rate also increases. Therefore sum rates are proportional to the number of users/traffic density in this case. Furthermore, from Fig. 5(b) even with light traffic conditions the sum rate of NOMA systems is higher than OMA. Lastly, Fig. 6(b) shows the relationships among long-term users connectivity during the simulation time. Where it is clearly visible that NOMA is more efficient for user connectivity by serving more users than OMA. From this figure we can see that the connectivity is improving as reinforcement learning agents, specifically DRL agent learning the dynamic environment. The number of served users are significantly increasing after 150 episodes of learning. As we can see the total number of served users are more than 3000 for DRL NOMA and more than 1000 for SARSA NOMA within 200 episodes.

B. DQN Loss vs Rewards

In Fig. 7(a), the loss (MSE) for the DRL algorithm is shown, comparing three well-known activation functions (ReLU, TanH, and Sigmoid). As it can be seen that ReLU performs better than both Sigmoid and TanH activation functions. Sigmoid and TanH perform relatively better only in initial steps due to less experience of the DRL agent. Therefore, when DRL agent gains some experience after the process of exploration and exploitation of the given environment, the outcome of the DRL algorithm is changed accordingly. The loss (y-axis) for all the activation



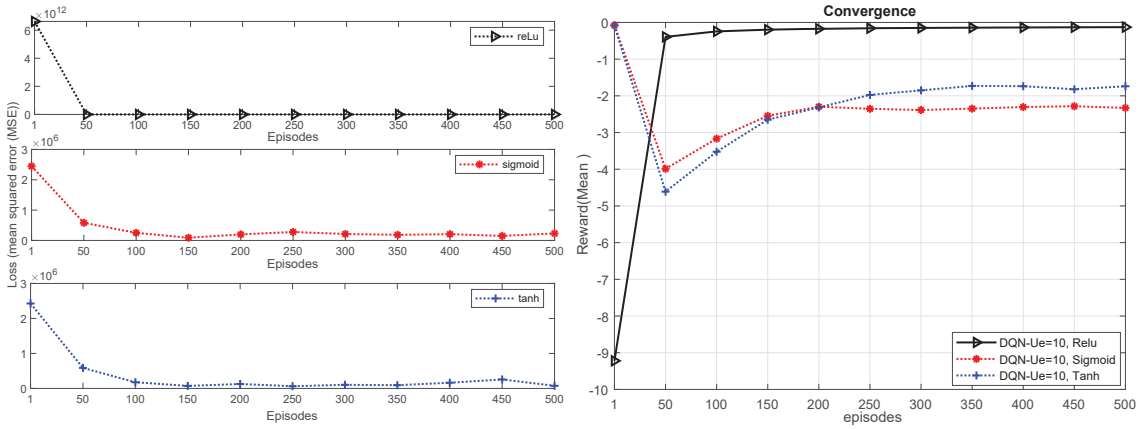
(a) (b)

Fig. 6. Overview for the proposed framework to the sum-rate maximization problem. Sub-figure (a) is short-term comparison between the channel bandwidth, average sum rate, and different network traffic loads for the proposed DRL and SARSA. Where (L) denotes light traffic,(M) denotes medium, and (H) is for heavy traffic. (b) shows long-term comparison between time episodes and clustering parameter c, showing sum of connected users in long-term for the proposed SARSA, DRL and OMA scheme.

functions is decreasing according to the number of episodes (x-axis). Furthermore, this figure also shows that the performance (loss) of the DRL algorithm is efficient when ReLU activation is used. Fig. 7(b) provides the summary statistics of achieved average rewards for the three different activation functions of the DRL algorithm. From the data in Fig. 7(b), it is apparent that the DRL algorithm with ReLU outperforms Sigmoid and TanH activation functions. After combining Fig. 7(a) and Fig. 7(b), another interesting outcome is that by the improvement of loss function, the rewards improves as well. Therefore, the loss and reward are proportional to each other. Lastly, DRL with the Sigmoid activations is the second best until 200 episodes and in all the remaining cases, where the episode is greater than 200 the performance of TanH is better than Sigmoid.

C. Clustering Time

The average clustering time in second is compared for DRL and SARSA algorithms with different types of traffic and impact of learning rates α in Fig. 8. The learning rate is the significant hyper-parameter of RL algorithms, which shows how long the agent spends to explore and exploit the given environments. From the figure, it can be seen that there is no large effect of learning rates on clustering time (y-axis) for all the scenarios with current hyper-parameters but if it is not tuned with other hyper-parameters, learning rate can negatively influence the learning process. Therefore, with improper tuning the learning process becomes unbalanced and



(a) (b)

Fig. 7. Overview for the proposed framework to the sum-rate maximization problem. Sub-figure (a) DRL loss vs number of episodes: A comparison between DRL loss and training episodes for different activation functions (ReLU, TanH, Sigmoid). Sub-figure (b) shows Rewards vs activation functions: A comparison between achieved rewards and episodes for different activation functions (ReLU, TanH, Sigmoid) of DRL algorithm.

the agent can be searching for the solution for an infinite amount of time. Lastly, the clustering time increases but not significant when max load is increased from 3 to 10.

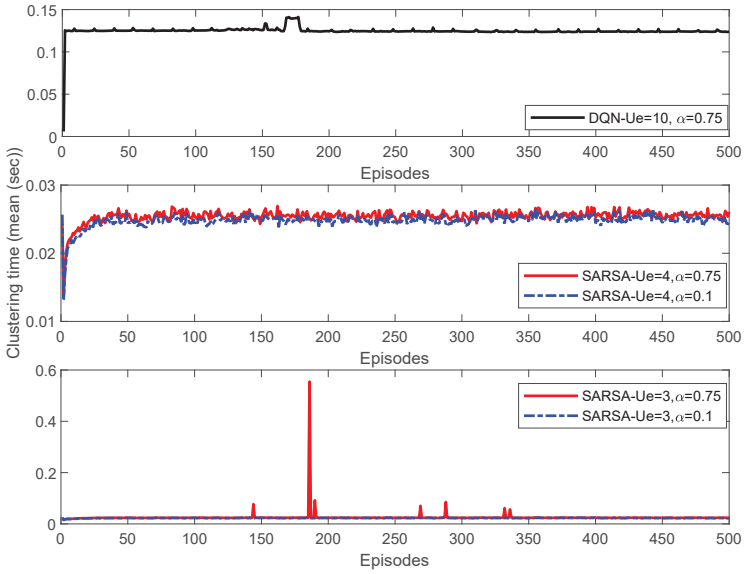


Fig. 8. Clustering time (mean (sec)) vs Traffic Densities for DRL and SARSA: A comparison between different traffic densities and learning rates of the proposed algorithms.

V. Conclusion

This paper has proposed resource allocation for IoT users in the uplink transmission of NOMA systems. Two algorithms DRL and SARSA in the present study have been designed to determine

the effect of three different traffic densities on the sum rate of IoT users. In order to improve the overall sum rate under different number of IoT users, we have formulated a multi-dimensional optimization problem using intelligent clustering based on RL algorithms with several interesting outcomes. Firstly, the simulation results of this study has indicated that the proposed technique performed close to the benchmark scheme in all the scenarios. The second major finding is that this frame work provides long-term guaranteed average rate with long-term reliability and stability. Thirdly, it has proved that DRL is efficient for complex scenarios. Additionally, we have proved that the sparse activations improve the performance of the DNNs when compared to the traditional mechanisms. Therefore, DRL with sparse activations is suitable for heavy traffic and SARSA is suitable for light traffic conditions. Furthermore, in general, both the algorithms (DRL and SARSA) have obtained better sum rates than OMA systems. Lastly, further research will explore performance improvements under the different scale of the networks.

APPENDIX A
PROOF OF PROBLEM (6a)

With the aid of the theory of computation complexity, we are able to use the following two steps to prove that the problem (6a) is an NP-hard problem. Step 1: the association problem for every subset of $\mathbf{\Phi_u^{i,j}}$ is NP-complete. Step 2: this step is to prove the relationship of $u_k^{i,j}$ and the problem in (**??**) is similar to our objective function. The problem (6a) in this paper is NP-hard, following proof can be divided into two cases, namely $N_u = 1$ (static clustering/association) and $N_u \geq 1$ (dynamic clustering/associations).

1) For the case $N_u = 1$ (static clustering/association), the problem (6a) is similar to the conventional OMA systems so that the resource management problem can be expressed as follows:

$$\max_{\mathbf{C},\mathbf{P}} \ \mathbb{E}[R_{sum}(t)], \tag{A.1}$$

$$\text{s.t}: \quad 2 \leq c_k^{i,j}(t), \forall i, k = 1, \tag{A.2}$$

$$\sum_{k=1}^{N_u^{i,j}} c_k^{i,j}(t) p_k^{i,j}(t) \leq P_b, \forall i, \forall j. \tag{A.3}$$

The above-mentioned problem has been proved to be NP-hard in [56] for OMA systems.

2) For the case $N_u > 1$ (dynamic clustering/associations), even with known power allocations we show that the problem (6a) is NP-hard since the optimal power selection for multiple

users is NP-hard. Additionally, it is known that 3D associations are NP-hard problems [50]. Under the condition that $N_u > 1$, for any $u_k^{i,j}$, there are more than one combinations in the set $\Phi_u^{i,j}$ even for the 3D association problem in OMA systems. Moreover, the combinations in NOMA is larger than those in OMA.

As a result, the decision problem of the constructed instance is NP-complete and the main instance is NP-hard.

REFERENCES

[1] W. Ahsan, W. Yi, Z. Qin, Y. Liu, and A. Nallanathan, "Reinforcement learning for user clustering in NOMA-enabled uplink IoT," in *IEEE Proc. of International Commun. Conf. (ICC Wkshps)*, Jun. 2020.

[2] D. Zhai, R. Zhang, L. Cai, and F. R. Yu, "Delay minimization for massive internet of things with non-orthogonal multiple access," *IEEE J. Sel. Areas Commun.*, vol. 13, no. 3, pp. 553–566, 2019.

[3] S. R. Islam, N. Avazov, O. A. Dobre, and K.-S. Kwak, "Power-domain non-orthogonal multiple access (NOMA) in 5G systems: Potentials and challenges," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 2, pp. 721–742, 2017.

[4] S. K. Sharma and X. Wang, "Towards massive machine type communications in ultra-dense cellular iot networks: Current issues and machine learning-assisted solutions," *IEEE Commun. Surveys Tuts.*, pp. 1–1, 2019.

[5] D. Wan, M. Wen, F. Ji, H. Yu, and F. Chen, "Non-orthogonal multiple access for cooperative communications: Challenges, opportunities, and trends," *IEEE Wireless Commun.*, vol. 25, no. 2, pp. 109–117, 2018.

[6] H. Guo, Y.-C. Liang, J. Chen, and E. G. Larsson, "Weighted sum-rate maximization for reconfigurable intelligent surface aided wireless networks," *IEEE Trans. Wireless Commun.*, vol. 19, no. 5, pp. 3064–3076, 2020.

[7] M. Zeng, X. Li, G. Li, W. Hao, and O. Dobre, "Sum rate maximization for irs-assisted uplink noma," *arXiv preprint arXiv:2004.10791*, 2020.

[8] D. Tse and P. Viswanath, *Fundamentals of wireless communication*. Cambridge university press, 2005.

[9] Z. Ding, X. Lei, G. K. Karagiannidis, R. Schober, J. Yuan, and V. K. Bhargava, "A survey on non-orthogonal multiple access for 5G networks: Research challenges and future trends," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 10, pp. 2181–2195, 2017.

[10] X. Shao, C. Yang, D. Chen, N. Zhao, and F. R. Yu, "Dynamic IoT device clustering and energy management with hybrid NOMA systems," *IEEE Trans. Ind. Informat.*, vol. 14, no. 10, pp. 4622–4630, 2018.

[11] M. S. Ali, H. Tabassum, and E. Hossain, "Dynamic user clustering and power allocation for uplink and downlink non-orthogonal multiple access (noma) systems," *IEEE access*, vol. 4, pp. 6325–6343, 2016.

[12] L. Miuccio, D. Panno, and S. Riolo, "Joint control of random access and dynamic uplink resource dimensioning for massive mtc in 5g nr based on scma," *IEEE Internet Things J.*, 2020.

[13] A. E. Mostafa, Y. Zhou, and V. W. Wong, "Connection density maximization of narrowband iot systems with NOMA," *IEEE Trans. Wireless Commun.*, vol. 18, no. 10, pp. 4708–4722, 2019.

[14] D. Wang, D. Chen, B. Song, N. Guizani, X. Yu, and X. Du, "From IoT to 5G I-IoT: The next generation IoT-Based intelligent algorithms and 5G technologies," *IEEE Commun. Mag.*, vol. 56, no. 10, pp. 114–120, OCTOBER 2018.

[15] F. Hussain, S. A. Hassan, R. Hussain, and E. Hossain, "Machine learning for resource management in cellular and IoT networks: Potentials, current solutions, and open challenges," *arXiv preprint arXiv:1907.08965*, 2019.

[16] N. Zhang, J. Wang, G. Kang, and Y. Liu, "Uplink nonorthogonal multiple access in 5G systems," *IEEE Commun. Lett.*, vol. 20, no. 3, pp. 458–461, 2016.

[17] Z. Ding, Z. Yang, P. Fan, and H. V. Poor, "On the performance of non-orthogonal multiple access in 5G systems with randomly deployed users," *arXiv preprint arXiv:1406.1516*, 2014.

[18] M. F. Hanif, Z. Ding, T. Ratnarajah, and G. K. Karagiannidis, "A minorization-maximization method for optimizing sum rate in the downlink of non-orthogonal multiple access systems." *IEEE Trans. Signal Process.*, vol. 64, no. 1, pp. 76–88, 2016.

[19] H. Zhang, B. Wang, C. Jiang, K. Long, A. Nallanathan, V. C. Leung, and H. V. Poor, "Energy efficient dynamic resource optimization in NOMA system," *IEEE Trans. Wireless Commun.*, vol. 17, no. 9, pp. 5671–5683, 2018.

[20] Z. Yang, Z. Ding, P. Fan, and N. Al-Dhahir, "A general power allocation scheme to guarantee quality of service in downlink and uplink NOMA systems," *IEEE Trans. Wireless Commun.*, vol. 15, no. 11, pp. 7244–7257, 2016.

[21] D. Zhai, R. Zhang, L. Cai, B. Li, and Y. Jiang, "Energy-efficient user scheduling and power allocation for NOMA-based wireless networks with massive IoT devices," *IEEE Internet Things J.*, vol. 5, no. 3, pp. 1857–1868, 2018.

[22] Y. Liu, Z. Qin, M. Elkashlan, Y. Gao, and L. Hanzo, "Enhancing the physical layer security of non-orthogonal multiple access in large-scale networks." *IEEE Trans.Wireless Commun.*, vol. 16, no. 3, pp. 1656–1672, 2017.

[23] W. Yi, Y. Liu, A. Nallanathan, and M. Elkashlan, "Clustered millimeter-wave networks with non-orthogonal multiple access," *IEEE Trans. Commun.*, vol. 67, no. 6, pp. 4350–4364, Jun. 2019.

[24] M. S. Ali, E. Hossain, and D. I. Kim, "Coordinated multipoint transmission in downlink multi-cell NOMA systems: Models and spectral efficiency performance," *IEEE Wireless Commun.*, vol. 25, no. 2, pp. 24–31, 2018.

[25] L. P. Qian, A. Feng, Y. Huang, Y. Wu, B. Ji, and Z. Shi, "Optimal SIC ordering and computation resource allocation in MEC-aware NOMA NB-IoT networks," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 2806–2816, April 2019.

[26] M. B. Shahab, R. Abbas, M. Shirvanimoghaddam, and S. J. Johnson, "Grant-free non-orthogonal multiple access for IoT: A survey," *arXiv preprint arXiv:1910.06529*, 2019.

[27] L. Dai, B. Wang, Z. Ding, Z. Wang, S. Chen, and L. Hanzo, "A survey of non-orthogonal multiple access for 5G," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 3, pp. 2294–2323, thirdquarter 2018.

[28] G. Gui, H. Huang, Y. Song, and H. Sari, "Deep learning for an effective nonorthogonal multiple access scheme," *IEEE Trans. Veh. Technol.*, vol. 67, no. 9, pp. 8440–8450, 2018.

[29] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[30] Y. Xu, D. Cai, F. Fang, Z. Ding, C. Shen, and G. Zhu, "Outage analysis and power allocation for HARQ-CC enabled NOMA downlink transmission," in *IEEE Glob. Commun. Conf. (GLOBECOM)*, Dec 2018, pp. 1–6.

[31] C. Jiang, H. Zhang, Y. Ren, Z. Han, K.-C. Chen, and L. Hanzo, "Machine learning paradigms for next-generation wireless networks," *IEEE Wireless Commun.*, vol. 24, no. 2, pp. 98–105, 2017.

[32] S. Bi, R. Zhang, Z. Ding, and S. Cui, "Wireless communications in the era of big data," *arXiv preprint arXiv:1508.06369*, 2015.

[33] M. Y. Arafat and S. Moh, "Localization and clustering based on swarm intelligence in UAV networks for emergency communications," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 8958–8976, Oct 2019.

[34] J. Cui, Z. Ding, P. Fan, and N. Al-Dhahir, "Unsupervised machine learning-based user clustering in millimeter-wave-NOMA systems," *IEEE Trans. Wireless Commun.*, vol. 17, no. 11, pp. 7425–7440, 2018.

[35] Y. Liu, S. Bi, Z. Shi, and L. Hanzo, "When machine learning meets big data: A wireless communication perspective," *arXiv preprint arXiv:1901.08329*, 2019.

[36] F. Li, D. Yu, H. Yang, J. Yu, H. Karl, and X. Cheng, "Multi-armed-bandit-based spectrum scheduling algorithms in wireless networks: A survey," *IEEE Wireless Commun.*, vol. 27, no. 1, pp. 24–30, 2020.

[37] T. B. de Oliveira, A. L. Bazzan, B. C. da Silva, and R. Grunitzki, "Comparing multi-armed bandit algorithms and q-learning

for multiagent action selection: a case study in route choice," in *2018 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2018, pp. 1–8.

[38] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton *et al.*, "Mastering the game of go without human knowledge," *Nature*, vol. 550, no. 7676, p. 354, 2017.

[39] G. A. Rummery and M. Niranjan, *On-line Q-learning using connectionist systems*. University of Cambridge, Department of Engineering Cambridge, England, 1994, vol. 37.

[40] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," *arXiv preprint arXiv:1509.02971*, 2015.

[41] L. Xiao, Y. Li, C. Dai, H. Dai, and H. V. Poor, "Reinforcement learning-based NOMA power allocation in the presence of smart jamming," *IEEE Trans. Veh. Technol.*, vol. 67, no. 4, pp. 3377–3389, 2017.

[42] Y. Liu, Z. Qin, Y. Cai, Y. Gao, G. Y. Li, and A. Nallanathan, "UAV communications based on non-orthogonal multiple access," *IEEE Wireless Commun.*, vol. 26, no. 1, pp. 52–57, 2019.

[43] H. Yang, P. Du, W.-D. Zhong, C. Chen, A. Alphones, and S. Zhang, "Reinforcement learning based intelligent resource allocation for integrated VLCP systems," *IEEE Wireless Commun. Lett.*, vol. 8, no. 4, pp. 1204–1207, 2019.

[44] J. Cui, Y. Liu, Z. Ding, P. Fan, and A. Nallanathan, "Optimal user scheduling and power allocation for millimeter wave NOMA systems," *IEEE Trans. Wireless Commun.*, vol. 17, no. 3, pp. 1502–1517, 2017.

[45] A. Kiani and N. Ansari, "Edge computing aware NOMA for 5G networks," *IEEE Internet Things J.*, vol. 5, no. 2, pp. 1299–1306, 2018.

[46] Y. Liu, Z. Ding, M. Elkashlan, and H. V. Poor, "Cooperative non-orthogonal multiple access with simultaneous wireless information and power transfer," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 4, pp. 938–953, 2016.

[47] Qinqing Zhang and S. A. Kassam, "Finite-state Markov model for rayleigh fading channels," *IEEE Trans. Commun.*, vol. 47, no. 11, pp. 1688–1692, Nov 1999.

[48] F. Fang, Z. Ding, W. Liang, and H. Zhang, "Optimal energy efficient power allocation with user fairness for uplink MC-NOMA systems," *IEEE Wireless Commun. Lett.*, pp. 1–1, 2019.

[49] X. Liu, Z. Qin, Y. Gao, and J. A. McCann, "Resource allocation in wireless powered IoT networks," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4935–4945, June 2019.

[50] J. Cui, Y. Liu, Z. Ding, P. Fan, and A. Nallanathan, "Optimal user scheduling and power allocation for millimeter wave NOMA systems," *IEEE Trans. Wireless Commun.*, vol. 17, no. 3, pp. 1502–1517, 2018.

[51] J. Cui, Y. Liu, and A. Nallanathan, "Multi-agent reinforcement learning-based resource allocation for UAV networks," *IEEE Trans. Wireless Commun.*, vol. 19, no. 2, pp. 729–743, Feb 2020.

[52] C. J. Watkins and P. Dayan, "Q-learning," *Machine learning*, vol. 8, no. 3-4, pp. 279–292, 1992.

[53] F. S. Melo, "Convergence of Q-learning: A simple proof," *Institute Of Systems and Robotics, Tech. Rep*, pp. 1–4, 2001.

[54] Q. Zhang and S. A. Kassam, "Finite-state Markov model for rayleigh fading channels," *IEEE Trans. Commun.*, vol. 47, no. 11, pp. 1688–1692, 1999.

[55] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[56] O. Naparstek and K. Cohen, "Deep multi-user reinforcement learning for distributed dynamic spectrum access," *IEEE Trans. Wireless Commun.*, vol. 18, no. 1, pp. 310–323, 2018.