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

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

Abstract—Non-orthogonal multiple access (NOMA) exploits the potential of the power domain to enhance the connectivity for the 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 article develops an intelligent resource allocation scheme for uplink NOMA-IoT communications. To maximise the average performance of sum rates, this work designs an efficient optimisation 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 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 have lower computational complexity with acceptable accuracy 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 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 the power domain to enable more connectivity than the traditional orthogonal multiple access (OMA). More specifically, NOMA supports multiple users at 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, the sum-rate is widely used as a significant performance indicator for wireless networks by numerous research works [7], [8]. It shows the significance of the sum-rate maximization based objective functions. Regarding 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 users are simultaneously 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]–[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 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 [20] 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 optimize 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], a 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 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 these designers use the 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 play a 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 mechanisms, 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 the 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 article, 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 a 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 is strongly dependent on the main problems “balancing of network resources, IoT users and, the dynamic network behaviour”.

B. Contributions and Organization

This article 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 article 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.

II. SYSTEM MODEL

In this article, 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 $N_c$ 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 $\Phi_b = \{b_1, \ldots, b_{N_b}\}$ and $\Phi_s = \{s_1, \ldots, s_{N_s}\}$, 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 $\Phi^{i,j}_u = \{u_1, \ldots, u_{N^{i,j}_u}\}$, where $N^{i,j}_u$ is the number of the intra-set users and $\sum_{i=1}^{N_b} \sum_{j=1}^{N_s} N^{i,j}_u = N_u$. 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 article, each sub-channel represents one NOMA cluster and $N^{i,j}_u \geq 2$ [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. A detailed explanation is present in 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 long-term communication. For an arbitrary user $u_k$, we define its clustering variable at time $t$ as follows:

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

(1)

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

B. Signal Model

In a NOMA cluster $s_j$, one BS $b_i$ first receives the superposed messages from the active users in $\Phi^{i,j}_u$ and then applies SIC to sequentially decode each user’s signal [46]. Without loss of generality, we assume the order of channel gains is $g^{i,j}_1 \leq g^{i,j}_2 \leq \cdots \leq g^{i,j}_{N^{i,j}_u}$, where $g^{i,j}_k$ is the channel gain for
the $k$-th user in $\Phi_{u_k}^{i,j}$ [47]. Therefore, the decoding order in this article 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 $u_{i,j}^{k-1} \in \Phi_u$ is given by

$$
\gamma_{k}^{i,j}(t) = \frac{c_{k}^{i,j}(t)p_{k}^{i,j}(t)g_{k}^{i,j}(t)}{\sum_{k'=1}^{k-1} c_{k'}^{i,j}(t)p_{k'}^{i,j}(t)g_{k'}^{i,j}(t) + I_{\text{inter}}(t) + \sigma^2(t)} ,
$$

where

$$
I_{\text{inter}}(t) = \sum_{i' \in \Phi_u \setminus \Phi_k} \sum_{k' \in \Phi_{i',j}^{k-1}} c_{k'}^{i',j}(t)p_{k'}^{i',j}(t)g_{k'}^{i',j}(t)
$$

and $p_{k}^{i,j}(t)$ is the transmit power of the user $u_{i,j}^{k}$ and the set of transmit power is given by $P_k$ ($p_{k}^{i,j}(t) \in P_k, \forall i, j, k$) [49]. The power of thermal noise obeys $\sigma^2(t) = k \beta T_r$, where $T_r$ is the temperature of resistors $k_\beta$ is Boltzmann’s constant, $B$ is the considered bandwidth. In this article we use $T_r = 300$ K therefore, $\sigma^2(t) \approx 4.14 \times 10^{12}$ BW. The $I_{\text{inter}}(t)$ 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 $u_{i,j}^{k}$ is based on the SIC process of its previous user $u_{i,j}^{k+1}$. If the data rate of successfully completing the SIC process is $R_{k+1}^{th}$, when the decoding rate of user $u_{i,j}^{k+1}$ obeys

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

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

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

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

### 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 $[2U_sN_tN_a, U_sN_tN_a \leq N_s]$. Under this condition, the average long-term sum rate can be maximized by optimizing clustering parameters $C = \{C_1, \ldots, C_T \}$ and transmit
power $P = \{P_1, \ldots, P_T\}$. Therefore, the objective function is given by
\[
\max_{C, P} \frac{B}{N_s} \sum_{i=1}^{T} \sum_{j=1}^{N_s} \sum_{l=1}^{N_c} \sum_{k=1}^{N^i_j} \log_2 \left( 1 + \gamma^{i,j}_k (t) \right), \quad (6a)
\]
subject to:
\[
g^{i,j}_k (t) \leq \cdots \cdots \leq g^{i,j}_0, \quad \forall i, j, t, \quad (6b)
\]
\[
\sum_{k=1}^{N_u} c^{i,j}_k (t) p^{i,j}_k (t) \leq P_s, \quad \forall i, j, t, \quad (6c)
\]
\[
\gamma^{i,j}_k (t) \geq 2^{R^{i,j}_k N_s / B} - 1, \quad \forall k, t, \quad (6d)
\]
\[
2 \leq \sum_{i=1}^{N_u} \sum_{j=1}^{N_u} \sum_{k=1}^{N^i_j} c^{i,j}_k (t) \leq N_u, \quad \forall t, \quad (6e)
\]
\[
\sum_{k=1}^{N^i_j} \sum_{j=1}^{N^i_j} c^{i,j}_k (t) \leq U_s, \quad \forall i, j, t, \quad (6f)
\]
\[
\sum_{k=1}^{N_u} N_u = N, \quad \forall \sum_{i=1}^{N_u} j=1 \sum_{k=1}^{N^i_j} c^{i,j}_k (t) \in \{1, 0\}, \quad \forall k, t, \quad (6g)
\]
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 a 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 [44] and [50].

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 $\pi$ 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 $s' \rightarrow$ 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 trial $T$ 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 $\pi'$ is learned by estimation of state-value optimization function $Q^\pi_t (s, a)$, $\forall s, a$, but it requires more memory for complex state space. DRL uses a neural network to simplify the Q-table by reducing memory requirements to handle more complex types of problems. Furthermore, the design complexity for the SARSA algorithm is less because we only need to design Q-table. However, for the DRL algorithm, the design is more complex due to the deep neural network (DNN) and additional hyperparameters. 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 the SARSA algorithm fails to provide an optimal policy for any type of network traffic during threshold trial $T_e$ then the 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.

![Fig. 2. Overview of the proposed framework for 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.](image-url)
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 as huge as compared to heavy traffic. Therefore, we use the SARSA learning algorithm to find optimal long term policy. The traditional Q-learning is not suitable for the 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 as huge as compared to heavy traffic that requires 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 \(\gamma\), sum reward, and the number of iterations are significant hyperparameters. The details for the flow of the information update is shown in Fig. 3. The 5-tuple \((S, A, P, R, S')\) SARSA-learning elements are mentioned below:

1) \(S\), 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) \(A\), is an action space that 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 \(A\) means agent performs swap operation of user between sub-channels at BS. In this model, agents have a total of 8 swap operations between BSs and sub-channels.

\[
A = \left\{ \begin{array}{ccc}
-1 & 0 & 0 \\
0 & -1 & 0 \\
-1 & 0 & 1 \\
0 & 1 & 0 \\
0 & 0 & -1 \\
0 & 0 & 1 \\
1 & 0 & -1 \\
1 & 0 & 0 \\
\end{array} \right\}.
\] (7)

3) \(P\), is an expected probability \(P_{s ightarrow s'} = Pr(s'|s,a)\) 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) \(R\), is a finite set of rewards, where the reward obtained after state \(s\) transition to next state \(s'\) by taking action \(a\). The reward function is denoted by \(r_{s ightarrow s'}^a\), showing that in the result of all associations the agent will receive a reward according to the conditions mentioned in reward function.

5) Multi-constrained reward function, the short-term reward in the 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 \(S'\). The reward function...
can be expressed as follows:

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

(8)

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

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

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

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

\[
\pi'(s) = \arg \max_a Q'(s, a), \quad \forall s \in S',
\]

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

\[
V_{\pi'}(s_t) = \max_a \left[ r(s_t, a_t) + \gamma \sum_{s'} P_{s \rightarrow s'} V_{\pi'}(s') \right].
\]

(10)

For Q-table value updating that contains state and corresponding action values of an agent. Bellman equation is defined as:

\[
Q(s, a) \leftarrow (1 - \alpha)Q(s, a) + \alpha [r' + \gamma Q(s', a')],
\]

where \(\gamma \in (0, 1)\) is a discount factor, which is a balancing factor between historical and future Q-table values. The larger \(\gamma\) 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 \(\alpha\) contributes to fast learning but due to minimal experience, it may result in non-convergence). Similarly, if the value of \(\alpha\) 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 \(P_{\pi}(A = a|S = s)\) 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 [51], [52]. Due to dynamic behaviour of IoT users the 3D state \(S\) and action space, \(A\) influence learning process more as \(S\) and \(A\) are main parts of Q-table \(Q(s, a)\). The convergence of the reward functions \(r\) and reinforcement learning hyperparameters guide the algorithm towards optimal policy \(V\). In other words, the choice of a reward function and the values for \(\{\epsilon, \alpha \text{ and } \lambda\}\) are used by reinforcement learning agent/s to avoid the random walk. The random walk in search space causes infinite exploration of the search space resulting in no convergence. Therefore, we are able to propose the following conclusion.

**Remark 1**: The selection of suitable rewards \(r_{s-a}\) 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 \(\epsilon \rightarrow \text{greedy}\) 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 \(S_t = \{s_1, s_2, \ldots, s_N\}\) possible states, an agent will converge towards the optimal states and actions pairing with the highest probability \(P_{s-t}\). 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 the 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:

- Line \#(1 – 6): presents the 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 \#2 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 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 \#(3 – 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
Algorithm 1 SARSAR-Learning Based NOMA-IoT Uplink Resource Optimization

1: Inputs for SARSAR:
   1) Episodes $N_e$
   2) Explorations per trials $T_{e}$
   3) Learning rate $\alpha$
2: Initialization for SARSAR:
   1) Network parameters $(N_b, b_t, N_s, s_j, N_u, u_k, P_b)$
   2) Q-Table $Q(s, a)$
3: Define number of clusters-k
4: Define range of users per cluster
5: load $s = N_0_{N_s} \times N_0_{N_s}$ and $a = [-1, 0, +1]$
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 $r(s_{t}, s_{t+1}, a_{t})$
12:   $r = 0$, if $R_{s_{t+1}} \geq R_{s_{t}}$
13:   and $\sum_{t=1}^{T_e} (u_k^t) = \sum_{t=1}^{T_e} (u_k^t+1)$
14:   $r = -10$, otherwise.
15: update $R_{s_{t+1}}$
16: update $Q(s, a) \leftarrow (1 - a)Q(s, a) + \alpha [r + \gamma Q(s', a')]$
17: Update $\pi \rightarrow$ greediness
18: $s \leftarrow s', a \leftarrow a'$
end for
19: return optimised $(c, p)$ (6) under constraints (6a),(6b),(6c),(6d) and (6e)
18: end for
19: Return Q-Table $Q(s, a)$

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 (S, A, P, R, S', A') values (11) is updated on-line. To perform online updates using S, A, P, R, S', A' instead of S, A, P, R as (traditional Q-learning) the online learning mechanism becomes more fast converging. In other words, the agent finds optimal long-term on-line 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 $S$ 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 DNN $Q(s, a; \theta)$, to generate its Q-table with the help of $\theta$ by approximating the Q-values $Q(s, a)$ [38]. Therefore, DRL agents only need to memorize the $\theta$ 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, $\theta$ 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 a given environment.

1) $S$, 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 an instantaneous and average reward from previous iterations.

2) $R$, is a reward of the system that is denoted by $R = \{r_i, r_t\}$, where $r_i$ is an instantaneous rewards similar to SARSAR algorithm and $r_t = \sum_{i=0}^{500} r_i/t$ denotes long-term average rewards for the time slot $t$.

3) $A$, is a multi-dimensional matrix representing actions as $A = \{a_1, a_2, \cdots, a_N\}$. For the DRL algorithm, the action mechanism is based on two main parts as; allocation strategies described as switching strategy $a_s$ and association strategy $a_a$ for the optimization process, where $a_s$ is a switching mechanism similar to SARSAR and used for the DRL channel switching process. The second strategy $a_a$ is a result of selected switching strategy $a_s$, $a_i$ 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 $\theta$ based on the previous experience.

$$loss(\theta) = 1/N_e \sum_{i=1}^{N_e} [y_{i}^{DRL} - Q(s, a; \theta)]^2,$$ (12)

where

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

and $y_{i}^{DRL}$ is the target Q-values from target DNN. For the improved training, in general the update frequency of the target network $\theta'$ 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 $\theta$ is provided in (16), which is based on the outcome of new experience. The updating function for $\theta$ is defined in (18), namely DRL Bellman equation.

$$\theta \leftarrow \theta - [y_{i}^{DRL} - Q(s, a; \theta)] \nabla Q(s, a; \theta),$$ (14)

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

$$q_{x'}(s, a) = r(s, a) + \gamma \sum_{s' \in S} p_{s'x}(a) \max_{a' \in A} q_{x'}(s', a'),$$ (16)

The DRL agent uses gradient decent method as in (12) to reduce the prediction error by minimizing the loss function. The updating of $\theta$ is provided in (16), which is based on the outcome of new experience. The updating function for $\theta$ is defined in (18), namely DRL Bellman equation.
where the function $q_{\pi^*}(s, a)$ shows Q-values and the long-term reward calculations for DRL based on the discount factor $\gamma$ and below mentioned optimal DRL policy $\pi^*$.

$$
\pi^*(s) = \arg \max_{a \in A} [ q_{\pi^*}(s, a) ], \; \forall \; s \in S,
$$

(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 state set after taking appropriate action $a$.

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

(18)

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

$$
s(t) = \{ a_1 (r_1, r_1), a_2 (r_1, r_1), ..., a_t (r_1, r_1) \}.
$$

(19)

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

(20)

where in (19), $s(t)$ represents state of the DRL agent and equation (20) shows the activation mechanism for each neuron layer $I_j$ based on weights $W_j$ for $j$-th depth of neurons with bias term $\psi_j$. 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 weight matrix $W_j$ to obtain the final output as a set of different Q-values for all actions. For the DRL framework, the size of output actions is 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 $\varepsilon$. When the size $\varepsilon$ is full, the first experience as the 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 distribution of the samples is independent and identical. Therefore, to get more general output, the $W_j$ update process is performed based on randomly sampled tuples $(s_t, a_t, r_t, s')$ 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.

**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:

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

1. **Inputs for DRL:**
   1) Episodes $N_e$
   2) Explorations per trials $T_e$
   3) Learning rate $\alpha$

2. **Initialization for DRL:**
   1) Network parameters $(N_b, b_1, N_s, s_j, N_u, u_k, P_b)$
   2) Memory, hidden size, State size, action size and mini-batch

3. **train DRL to find a good policy $\theta$**
4. **for iteration = 1:$N_e$ do**
5. **for iteration = 1:$T_e$ do**
6. $s_t, a_t$
7. compute $r(s_t, s_{t+2}, a_t)$
8. do
9. if $R_{s_{t+1}} \geq R_{s_t}$ and $\sum_{i=1}^{T_e} (u_k) = \sum_{i=1}^{T_e} (u_k)$
10. $r = -10$, otherwise.
11. update $\theta$ using $q_{\pi^*}(s, a) = r(s, a) + \gamma \max_{a' \in A} q_{\pi^*}(s', a')$, update using $y_{\text{DRL}} = r + \gamma \max_{a' \in A} Q(s', a'; \theta')$.
12. $s \leftarrow s', a \leftarrow a'$
13. end if
14. end for
15. end for
16. Return $Q(W_a)$

1) **DRL:**
   - Line #((1–2)): In this stage, the parameter initialization is performed, which is similar to SARSA. However, instead of state action pairing, the weight matrix is initialized for DRL to find optimal policy $\pi$.
   - Line #3 Pre-training: In this stage, initial actions are selected using uniform random distribution as an initial state space in a continuous environment. In this way, initial weights are also calculated to start the optimization process.
   - Line #((4–17)): Whole process for DRL is similar to SARSA from line #((5–11)) with DRL bellman equations (12) to (19).
   - Loss Calculation: The equation (12) is to calculate the loss $\theta$ 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 $\theta$ 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, $\gamma$ discount factor has a 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 $\pi$ is calculated using (17) to maximize the Q-values by the 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 ($\rho$). 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 $\delta_j$ 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 $\lambda$ before activation. In the next steps, current states, actions, and rewards are added to mini-batch for experience replay (for self-training). In $\#(10 - 14)$, 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 a replay mechanism with the help of mini-batch processing.

- **Neural Networks:** this article 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 $\theta$ and $\theta'$ shows the weights of these two networks. For the training of the DNN network, $\theta$ weights are predicted based on the current state and action. On the other hand, $\theta'$ weights are based on the previous episodes and these weights are fixed during the calculation of $\theta$ 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 $\theta$ and $\theta'$ to check the deviation of the predicted DNN weights.

- **Output:** Finally, the output of this algorithm is the optimal policy for all clusters where the 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 $\delta_j$ and $\lambda_i$ 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 $L$. $f(x) = w_{\Lambda+1}\rho(w_{\Lambda}\rho(w_{\Lambda-1}\ldots \rho(w_2\rho(w_1\rho + \psi_1) + \psi_2) \ldots \psi_{\Lambda-1}) + \psi_\Lambda.$

In this definition, $f(x)$ is a function to show the construction of neural network with weights for each layer $w_\lambda \in \mathbb{R}^{\delta_{\lambda-1}}$ and $\rho$ 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 $\rho$ and the input of the neural network. In the neural network $\Psi$ 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 a 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. Note close to 0 to explore the environment in search of different solutions most cases and some times it processes a random action. For DRL, unbalanced random actions cause huge error propagation so that this ϵ - greedy is suitable to be applied for achieving efficient learning in dynamic state space. Note that the boundary for the policy selection is 0 ≤ ϵ ≤ 1. For ϵ close to 0 the policy becomes a greedy policy, and for ϵ close to 1 the agent explores more.

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 layers, ψ > 0. For any hidden layer Λ, κ ∈ N, \( \{ \delta_j \}_{j=0}^{\Lambda+1} \leq N. \)

\[
F(\Lambda, \{ \delta_j \}_{j=0}^{\Lambda+1}, \kappa, \Psi) = \left( f : \max_{\lambda \in [\Lambda+1]} \left\| W^{\lambda} \right\|_\infty \leq 1, \right. \\
\sum_{\lambda=1}^{\Lambda+1} \left\| W^{\lambda} \right\|_2 \leq \kappa, \max_{j \in [\Lambda+1]} \left\| f_j \right\|_\infty \leq \Psi \right)
\]

where \( W^\lambda \) is used to represent \( W^\lambda, \psi^\lambda \). The function \( f \) is from Definition 5 and \( f_j \) is the \( j-\text{th} \) 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 article 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 \( N_u \) channels from users. The computation complexity for SARSA-learning is \( O(N_b N_u N_s) \) operations with \( N_s N_u \times \times (2 \times N_b \times N_u + 1) \) memory requirement for Q-table to simulate brain of the learning agent/s. The complexity of DRL is \( O(N_s T_u) \) with smaller Q-table \( O(Q(W_j)) \) 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 \( O(N_u N_b N_s) \).

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, the network load in our case represents the load of each resource block instead of the total number of users in the network. Therefore, max network load = 10 with two RB’s for each BS means \( 10 \times 4 = 40 \) 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 episode. Similarly, \( \lambda_{\{SARSA/DRL\}} = \{0.5, \gamma_{\{SARSA/DRL\}} = 0.6, \alpha_{\{SARSA/DRL\}} = \{0.75, 0.1\} \), and \( \epsilon \) - greedy exploration values are 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 \( [1, 1.5, 2] \times 10^{-5} \) [47]. 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. 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 scenarios with high traffic densities. Furthermore, to show the performance of DRL with medium traffic (M), we can see that compared to the SARSA algorithm DRL is handling medium traffic better by fast converging. 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 \( \theta \) weights as \( R_{DRL} = \frac{1}{T} \sum_{t=1}^{T} (f_t(\theta_t) - f_t(\theta_0)) \), where \( \theta' = \arg\min_{\theta \in \mathbf{R}} \sum_{t=1}^{T} f_t(\theta) \) and \( \kappa \) is feasible set for all \( t - 1 \) steps.

Definition 7: The bounded gradients of the function \( f_t^{DRL} \) is \( \| f_t^{DRL} \|_2 \leq G^{DRL} \| f_t^{DRL} \|_\infty \leq G^{DRL} \), \forall \theta \in R_{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 \( p, q \in \{1, \ldots, T\} \) with the bias terms \( \beta_1, \beta_2 \in [0, 1] \) satisfying the \( \frac{\beta_2}{\sqrt{\beta_2}} < 1 \) 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 [53]:

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

The results obtained from the primary analysis of sum rate and traffic densities are shown in Fig. 5(b) in long-term settings. It shows that the proposed model performs close to the benchmark scheme and better than OMA. Additionally, we compare the proposed NOMA with dynamic power and traditional NOMA scheme with the fixed power as NOMA-fix(P). It is clearly visible that the traditional method of allocation as NOMA-fix(P) is not as efficient as the proposed allocation strategy. 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, due to the fast convergence with heavy traffic as shown in Fig. 5(a) the performance of the DRL is better than SARSA. Interestingly it also shows that with the increase of the traffic density, the sum rate improves. 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...
Fig. 7. Overview of the proposed framework for the sum-rate maximization problem. Sub-figure (a) DRL loss vs the 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.

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 the 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 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 the loss function, the rewards improves as well. Therefore, 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 the impact of learning rates $\alpha$ 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, the learning rate can negatively influence the learning process. Therefore, with improper tuning the learning process becomes unbalanced and the agent can be searching for the solution for an infinite amount of time. Lastly, the clustering time increases but not significant when the max load is increased from 3 to 10.
V. CONCLUSION

This article 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 a 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 have indicated that the proposed technique performed close to the benchmark scheme in all the scenarios. The second major finding is that this framework provides a 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 $\Phi_u^{i,j}$ is NP-complete. Step 2: this step is to prove the relationship of $u_k^{i,j}$ and the problem in (A.1) is similar to our objective function. The problem (6a) in this article 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:

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

subject to:

$$2 \leq c_{k}^{i,j}(t), \quad \forall i, k = 1, \quad (A.2)$$

$$\sum_{k=1}^{N_u} c_{k}^{i,j}(t)p_{k}^{i,j}(t) \leq P_b, \quad \forall i, \forall j. \quad (A.3)$$

The above-mentioned problem has been proved to be NP-hard in [54] 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 [44].

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, Y. Liu, Z. Qin, and A. Nallanathan, “Reinforcement learning for user clustering in NOMA-enabled uplink IoT,” in *Proc. IEEE Int. Conf. Commun. Workshops (ICC Workshops)*, Jun. 2020, pp. 1–6.

[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. Topics Signal Process.*, vol. 13, no. 3, pp. 553–566, Jun. 2019.

[3] S. M. 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. 712–742, 2nd Quart., 2017.

[4] S. K. Sharma and X. Wang, “Toward massive machine type communications in ultra-dense cellular IoT networks: Current issues and machine learning-assisted solutions,” *IEEE Commun. Surveys Tuts.*, vol. 22, no. 1, pp. 426–471, 1st Quart., 2020.

[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, Apr. 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, May 2020.

[7] M. Zeng, X. Li, G. Li, W. Hao, and O. A. Dobre, “Sum rate maximization for IRS-assisted uplink NOMA,” 2020, arXiv:2004.10791. [Online]. Available: http://arxiv.org/abs/2004.10791

[8] D. Tse and P. Viswanath, *Fundamentals of Wireless Communication*. Cambridge, U.K.: Cambridge Univ. 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, Oct. 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, Oct. 2018.

[11] M. Shipon 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.*, vol. 7, no. 6, pp. 5042–5063, Jun. 2020.

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

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

[15] F. Hussain, S. Ali Hassan, R. Hussain, and E. Hossain, “Machine learning for resource management in cellular and IoT networks: Potentials, current solutions, and open challenges,” 2019, arXiv:1907.08965. [Online]. Available: http://arxiv.org/abs/1907.08965

[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, Mar. 2016.

[17] Z. Ding, Z. Yang, P. Fan, and H. Vincent Poor, “On the performance of non-orthogonal multiple access in 5G systems with randomly deployed users,” 2014, arXiv:1406.1516. [Online]. Available: http://arxiv.org/abs/1406.1516

[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, Jan. 2016.

[19] H. Zhang et al., “Energy efficient dynamic resource optimization in NOMA system,” *IEEE Trans. Wireless Commun.*, vol. 17, no. 9, pp. 5671–5683, Sep. 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, Nov. 2016.
C. Jiang, H. Zhang, Y. Ren, Z. Han, K.-C. Chen, and L. Hanzo, Y. Xu, D. Cai, F. Fang, Z. Ding, C. Shen, and G. Zhu, “Outage S. Bi, R. Zhang, Z. Ding, and S. Cui, “Wireless communications M. Basit Shahab, R. Abbas, M. Shirvanimoghaddam, and S. J. Johnson, “Grant-free non-orthogonal multiple access for IoT: A survey,” 2019. 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, Sep. 2018. S. Hochreiter and J. Schmidhuber, “Long short-term memory,” Neural Comput., vol. 9, no. 8, pp. 1735–1780, 1997. 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 Proc. IEEE Global Commun. Conf. (GLOBECOM), Dec. 2018, pp. 1–6. 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, Apr. 2017. S. Bi, R. Zhang, Z. Ding, and S. Cui, “Wireless communications in the era of big data,” 2015, arXiv:1508.06369. [Online]. Available: http://arxiv.org/abs/1508.06369 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. J. Cui, Y. Liu, 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, Nov. 2018. Y. Liu, S. Bi, Z. Shi, and L. Hanzo, “When machine learning meets big data: A wireless communication perspective,” 2019, arXiv:1901.08329. [Online]. Available: http://arxiv.org/abs/1901.08329 F. Fang, Z. Ding, Y. Wu, 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, Feb. 2020. T. B. F. de Oliveira, A. L. C. Bazzan, B. C. da Silva, and R. Gruniztki, “Comparing multi-armed bandit algorithms and Q-learning for multiagent action selection: A case study in route choice,” in Proc. Int. Joint Conf. Neural Netw. (IJCNN), Jul. 2018, pp. 1–8.
Waled Ahsan

Wenqiang Yi

Waled Ahsan (Graduate Student Member, IEEE) received the B.S. degree in telecommunication and networking from the COMSATS Institute of Technology, Pakistan, in 2016, and the M.S. degree in computer science from COMSATS University Islamabad, Pakistan, in 2018. He is currently pursuing the Ph.D. degree with Communication Systems Research Group, School of Electronic Engineering and Computer Science, Queen Mary University of London. His research interests include 5G and beyond wireless networks, the Internet of Things, NOMA, 5G/6G networks, V2X communications, evolutionary computation, meta-heuristic algorithms, swarm intelligence, and deep reinforcement learning.

Wenqiang Yi (Member, IEEE) received the Ph.D. degree in electrical engineering from the Queen Mary University of London, U.K., in 2020. He has been a Post-Doctoral Researcher with the Communication Systems Research Group, School of Electronic Engineering and Computer Science, Queen Mary University of London, since 2020. His research interests include NOMA, millimeter-wave communications, stochastic geometry, and reinforcement learning.

He served as a TPC member for many IEEE conferences, such as GLOBECOM. He received the Exemplary Reviewer of the IEEE COMMUNICATIONS LETTERS in 2019 and the IEEE TRANSACTIONS ON COMMUNICATIONS in 2019 and 2020. He serves as the Secretary for the Signal Processing Group on Next Generation Multiple Access (NGMA) by the Signal Processing and Computing for Communications Technical Committee.
Zhijin Qin (Senior Member, IEEE) received the B.S. degree from the Beijing University of Posts and Telecommunications in 2012 and the Ph.D. degree in electronic engineering from the Queen Mary University of London (QMUL) in 2016. She was a Post-Doctoral Research Associate with Imperial College London from 2016 to 2017, and then a Lecturer with Lancaster University from 2017 to 2018. Since 2018, she has been a Lecturer with the School of Electronic Engineering and Computer Science, QMUL. Her research interests include deep learning enabled semantic communications, end-to-end communications, deep compressive sensing, and intelligent resource allocation in various applications, such as UAV communications and reconfigurable intelligent surface.

She served as the Track-Chair for IEEE VTC fall 2019 and IEEE GLOBECOM 2020. She is serving as an Area Editor of the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS (JSAC) Series on Communications and Networks, an Associate Editor of the IEEE TRANSACTIONS ON COMMUNICATIONS, the IEEE COMMUNICATIONS LETTERS, and the IEEE TRANSACTIONS ON COGNITIVE COMMUNICATIONS AND NETWORKING.

Yuanwei Liu (Senior Member, IEEE) (http://www.eecs.qmul.ac.uk/~yuanwei) received the B.S. and M.S. degrees from the Beijing University of Posts and Telecommunications in 2011 and 2014, respectively, and the Ph.D. degree in electrical engineering from the Queen Mary University of London, U.K., in 2016. From 2016 to 2017, he was with the Department of Informatics, King’s College London, where he was a Post-Doctoral Research Fellow. He has been a Lecturer (Assistant Professor) with the School of Electronic Engineering and Computer Science, Queen Mary University of London, since 2017. His research interests include non-orthogonal multiple access, 5G/6G networks, machine learning, and stochastic geometry.

He served as a TPC member for many IEEE conferences, such as GLOBECOM and ICC. He received the IEEE ComSoc Outstanding Young Researcher Award for EMEA in 2020. He served as the Publicity Co-Chair for VTC 2019-Fall. He is the leading contributor for “Best Readings for Non-Orthogonal Multiple Access (NOMA)” and the primary contributor for “Best Readings for Reconfigurable Intelligent Surfaces (RIS)”. He serves as the Chair for Special Interest Group (SIG) in Signal Processing and Computing for Communications (SPCC) Technical Committee on the topic of signal processing Techniques for next generation multiple access (NGMA), the Vice Chair for SIG Wireless Communications Technical Committee (WTC) on the topic of Reconfigurable Intelligent Surfaces for Smart Radio Environments (RISE), and the Tutorials and Invited Presentations Officer for Reconfigurable Intelligent Surfaces Emerging Technology Initiative. He is currently an Editor on the Editorial Board of the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, the IEEE TRANSACTIONS ON COMMUNICATIONS, and the IEEE TRANSACTIONS ON COMMUNICATIONS LETTERS.

He serves as the Leading Guest Editor for the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS (JSAC) special issue on Next Generation Multiple Access, a Guest Editor for the IEEE JOURNAL OF SELECTED TOPICS IN SIGNAL PROCESSING (JSTSP) special issue on Signal Processing Advances for Non-Orthogonal Multiple Access in Next Generation Wireless Networks. He received the Exemplary Reviewer Certificate of the IEEE WIRELESS COMMUNICATIONS LETTERS in 2015, the IEEE TRANSACTIONS ON COMMUNICATIONS in 2016 and 2017, and the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS in 2017 and 2018.

Arumugam Nallanathan (Fellow, IEEE) was an Assistant Professor with the Department of Electrical and Computer Engineering, National University of Singapore, from August 2000 to December 2007. He was with the Department of Informatics, King’s College London, from December 2007 to August 2017, where he was a Professor of wireless communications from April 2013 to August 2017 and has also been a Visiting Professor since September 2017. He has been a Professor of wireless communications and the Head of the Communication Systems Research (CSR) Group, School of Electronic Engineering and Computer Science, Queen Mary University of London, since September 2017. He published nearly 500 technical articles in scientific journals and international conferences. His research interests include artificial intelligence for wireless systems, beyond 5G wireless networks, the Internet of Things (IoT), and molecular communications. He was a co-recipient of the Best Paper Awards presented at the IEEE International Conference on Communications 2016 (ICC’2016), the IEEE Global Communications Conference 2017 (GLOBECOM’2017), and the IEEE Vehicular Technology Conference 2018 (VTC’2018). He has been selected as a Web of Science Highly Cited Researcher in 2016 and AI 2000 Internet of Things Most Influential Scholar in 2020. He is an IEEE Distinguished Lecturer.

He received the IEEE Communications Society SPCE Outstanding Service Award 2012 and the IEEE Communications Society RCC Outstanding Service Award 2014. He served as the Chair for the Signal Processing and Communication Electronics Technical Committee of the IEEE Communications Society and a technical program chair and a member of Technical Program Committees in numerous IEEE conferences. He is an Editor-at-Large of the IEEE TRANSACTIONS ON COMMUNICATIONS and a Senior Editor for the IEEE WIRELESS COMMUNICATIONS LETTERS. He was an Editor for the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS from 2006 to 2011, the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY from 2006 to 2017, and the IEEE SIGNAL PROCESSING LETTERS.