Cooperative Deep Reinforcement Learning based Grant-Free NOMA Optimization for mURLLC

Yan Liu*†, Yansha Deng‡, Maged Elkashlan†, and Arumugam Nallanathan†
*Key Laboratory of Ministry of Education in Broadband Wireless Communication and Sensor Network Technology
Nanjing University of Posts and Telecommunications, China
†School of Electronic Engineering and Computer Science, Queen Mary University of London, London, UK
‡Department of Engineering, King's College London, UK

***Abstract*—Grant-free non-orthogonal multiple access (GF-NOMA) is a potential technique to support massive Ultra-Reliable and Low-Latency Communication (mURLLC) service. However, the dynamic resource configuration in GF-NOMA systems is challenging due to the random traffics and collisions, which are unknown at the base station (BS). Meanwhile, joint consideration of the latency and reliability requirements makes the resource configuration of GF-NOMA more complex. To address this problem, we develop a general learning framework for signature-based GF-NOMA in mURLLC service taking into account the multiple access signature collision, the user (UE) detection, as well as the data decoding procedures for the K-repetition GF-NOMA. The goal of our learning framework is to maximize the long-term average number of successfully served UEs under the latency constraint. We propose a Cooperative Multi-Agent Deep Neural Network based Q-learning (CMA-DQN) approach to optimize the configuration of both the repetition values and the contention-transmission unit (CTU) numbers. Our results show the superior performance of CMA-DQN over the LE-URC in heavy traffic and demonstrate its capability in dynamically configuring in long term for mURLLC service.**

***Index Terms*—mURLLC, NOMA, grant-free, deep reinforcement learning, resource configuration.**

I. Introduction

As a new and dominating service class in 6th Generation (6G) networks, massive Ultra-Reliable and Low Latency Communications (mURLLC) integrates URLLC with massive access to support massive short-packet data communications in time-sensitive wireless networks with high reliability and low access latency [1]. This requires a reliability-latency-scalability trade-off and mandates a principled and scalable framework accounting for the delay, reliability, and decision-making under uncertainty [2]. Concretely speaking, the Third Generation Partnership Project (3GPP) standard [3] has defined a general URLLC requirement: $1-10^{-5}$ reliability within 1ms user plane latency for 32 bytes. It is also anticipated that the device density may grow to hundred(s) of devices per cubic meter in the 6G networks.

Current cellular network can hardly fulfill the joint massive connectivity, ultra reliability, and low latency requirements in mURLLC service. To achieve low latency, *grant-free (GF) access* has been proposed [4] as an alternative for traditional grant-based (GB) access. Different from GB access, GF access allows a User Equipment (UE) to transmit its data to the Base Station (BS) in an arrive-and-go manner, without sending a scheduling request and obtaining a resource grant from the network [4]. To achieve high reliability, the *K-repetition* GF scheme has been proposed, where a pre-defined number (K) of consecutive replicas of the same packet are transmitted [4]. To achieve massive connectivity, *non-orthogonal multiple access (NOMA)* has been proposed to synergize with GF in order to deal with the multiple access (MA) physical resource collision in contention-based GF access on orthogonal multi-access (OMA) physical resources, when two or more UEs transmit data using the same MA physical resource [5]. Here, we focus on the signature-based GF-NOMA, where the NOMA technique allows multiple UEs to transmit over the same MA physical resource by employing user-specific signature patterns (e.g, codebook, pilot sequence, interleaver/mapping pattern, demodulation reference signal, power, etc.) [6]. However, when two or more UEs transmit data using the same MA physical resource and the same MA signature, the MA signature collision occurs, and the BS cannot differentiate among different UEs and therefore cannot decode the data [6].

The main challenges of the dynamic resource configuration optimization of GF-NOMA include: 1) the set of active UEs and their respective channel conditions are unknown to the BS, which prohibits the pre-configuration and the pre-assignment of resources, including pilots/preambles, power, codebooks, repetition values, and etc; 2) simultaneously satisfy the reliability and latency requirements under random traffics, the optimal parameter configurations vary over different time slots, which is hard to be described by a tractable mathematical model; 3) the MA signature collision detection and the blind UE activity detection, as well as the data decoding, need to be considered, which largely impacts the resource configuration in each time slot; 4) a general optimization framework for GF-NOMA systems have never been established for various signature-based NOMA schemes.

The above challenges can hardly be solved via traditional convex optimization methods, due to the complex communication environment with the lack of tractable mathematical formulations. Reinforcement Learning (RL), can be a promising tool to deal with this complex Partially Observable

Markov Decision Process (POMDP) problem of GF-NOMA resource configuration optimization, which solely relies on the self-learning of the environment interaction without deriving explicit optimization solutions based on a complex mathematical model.

In this paper, we aim to develop a general learning framework for GF-NOMA systems for mURLLC service. Our contributions can be summarized as follows:

- In this framework, we practically simulate the random traffics, the resource configuration, the transmission latency check, the collision detection, the data decoding, and the Hybrid Automatic Repeat reQuest (HARQ) retransmission procedures. We use this generated simulation environment to train the RL agents.
- We develop a Cooperative Multi-Agent Deep Q-Network (CMA-DQN) to dynamically optimize both the repetition parameters and MA resources, which breaks down the selection in high-dimensional parameters into multiple parallel sub-tasks with a number of DQN agents cooperatively being trained to produce each parameter.
- Our results show the superior performance of CMA-DQN for mURLLC service, especially in heavy traffic scenarios. Our general learning framework can be extended to optimize other resource configuration problems in GF-NOMA schemes.

The rest of the paper is organized as follows. Section II illustrates the system model. Section III illustrates the conventional approach. Section IV presents the CMA-DQN approach. Section V elaborates the numerical results. Section VI summarizes the conclusion.

II. System Model

We consider a BS located at the center and a set of N UEs randomly located in an area of the plane $\mathbb{R}^2$, where the UEs are in-synchronized and unaware of the status of each other. Once deployed, the UEs remain spatially static. The time is divided into short transmission time intervals (TTIs)[1], and the small packets for each UE are generated according to random inter-arrival processes over the short-TTIs, which are Markovian as defined in [8] and unknown to BS.

A. GF-NOMA Network Model

To capture the effects of the physical radio, we consider the standard power-law path-loss model with the path-loss attenuation $r^{-\eta}$, where r is the Euclidean distance between the UE and the BS and η is the path-loss attenuation factor. We consider a Rayleigh flat-fading environment, where the channel power gains h are exponentially distributed (i.i.d.) random variables with unit mean. The GF-NOMA procedure following the 3GPP standard [9] are explained in the following subsections.

[1]The simulation parameters used for this study are in line with the main guidelines for 3GPP NR performance evaluations presented in [7] with mini-slots of 7 OFDM symbols for transmissions in short TTI (0.125ms) using 60 kHz sub-carrier spacing (SCS)

1) Resources and Parameters Configuration: The MA resources, repetition values, and HARQ related parameters, etc, are configured at the BS by radio resource control (RRC) signaling and L1 signaling prior to the GF access (as Type 2 GF [4]).

a) Repetition values: We consider the K-repetition scheme as shown in Fig. 1, where the UEs are configured at the beginning of each round trip time (RTT) to autonomously transmit (T) the same packet for K^t repetitions in consecutive TTIs. The BS decodes (D) each repetition independently and the transmission in one RTT is successful when at least one repetition succeeds. After processing all the received K^T repetitions, the BS transmits the ACK/NACK feedback (F) to the UE.

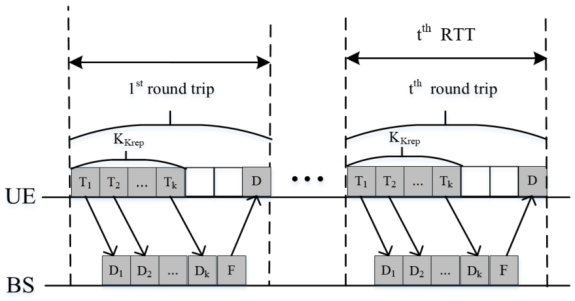


Fig. 1: K-repetition GF transmission

Considering the small packets of mURLLC traffic, we set the packet transmission time as one TTI. The BS feedback time and the BS (UE) processing time are also assumed to be one TTI as [10]. Once the repetition value is configured, the duration of one RTT is known to the UEs and the BS, which is given as

$$T^t_{\mathrm{RTT}} = (K^t + 3)\mathrm{TTIs}. \tag{1}$$

b) MA resources: A *contention-transmission unit* (CTU) as shown in Fig. 2 is defined as the basic MA resource, where each CTU may comprise of a MA physical resource and a MA signature [6]. The MA physical resources represent a set

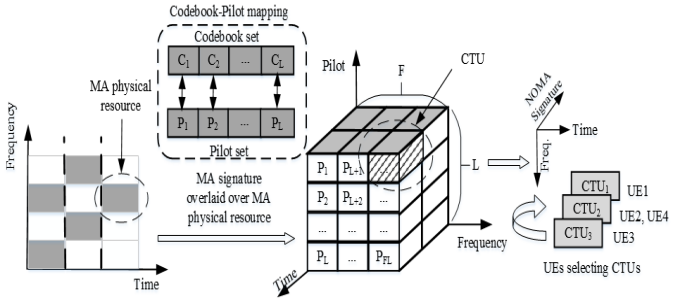


Fig. 2: GF-NOMA resource

of time-frequency resource blocks (RBs). The MA signatures represent a set of pilot sequences for channel estimation and/or UE activity detection, and a set of codebooks for robust data transmission and interference whitening, etc.

Without loss of generality, in one TTI, we consider F orthogonal RBs and each RB is overlaid with L unique codebook-pilot[2] pairs [5]. Thus, at the beginning of each RTT, the BS configures a resource pool of $C^t = F \times L$ unique CTUs, and each UE randomly choose one CTU from the pool to transmit in this RTT.

2) Latency Check: The HARQ index H_{HARQ} is included in the pilot sequence and can be detected by the BS. At the beginning of each RTT, the HARQ index and the transmission latency T_{late} will be updated. For example, for the initial RTT with initial K^1, $H_{HARQ} = 1$ and $T_{\text{late}} = RTT_{H_{HARQ}=1}$, where $RTT_{H_{HARQ}}$ is calculated by using (1). After this round time trip transmission, the BS optimizes a K^2 based on the observation of the reception and configures it to the UE for the next RTT. Then the UE updates its $H_{HARQ} = 2$ and calculated $RTT_{H_{HARQ}=2}$ with K^2, and consequently, the transmission latency T_{late} is updated as $T_{\text{late}} = RTT_{H_{HARQ}=1} + RTT_{H_{HARQ}=2}$. When $T_{\text{late}} > T_{\text{cons}}$ (T_{cons} is latency constraint), the UE fails to be served and the packets will be dropped.

3) Collision Detection: At each RTT, each active UE transmits its packets to the BS by randomly choosing a CTU. The BS can detect the UEs that have chosen different CTUs. However, if multiple UEs choose the same CTU, the BS

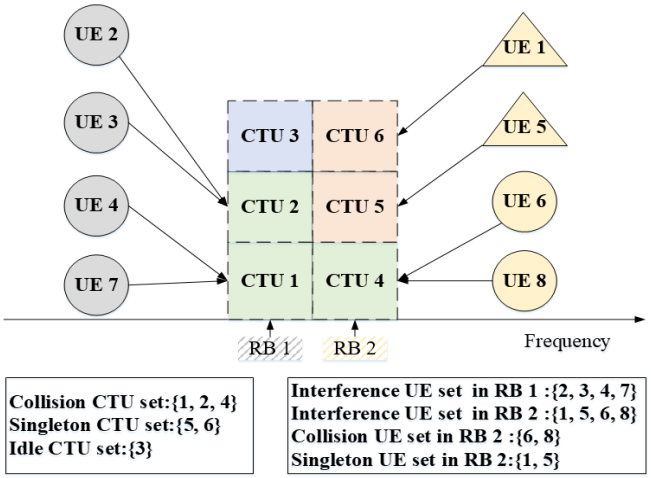


Fig. 3: Detection and Decoding case with L=2 RBs, C = 6 CTUs and N = 8 UEs.

cannot differentiate these UEs and therefore cannot decode the data. We categorize the CTUs into three types: an *idle* CTU is a CTU which has not been chosen by any UE; a *singleton* CTU is a CTU chosen by only one UE; and a *collision* CTU is a CTU chosen by two or more UEs [5]. One example is illustrated in Fig. 3. The UE 1 and UE 5 have chosen the unique CTU 6 and CTU 5, respectively, thus, the CTUs 6 and 5 are singleton CTUs. The CTU 3 is an idle CTU. The UE 4 and UE 7 have chosen the CTU 1, the UE 2 and UE 3 have chosen the CTU 2, and the UE 6 and UE 8 have chosen the CTU 4, thus, CTUs 1, 2 and 4 are the collision CTUs. After collision detection, the BS observes the set of singleton CTUs $\mathcal{C}^t_{sc}$, the set of idle CTUs $\mathcal{C}^t_{ic}$, and the set of collision CTUs $\mathcal{C}^t_{cc}$ as shown in orange, blue and green color, respectively, in Fig. 3

4) Data Decoding: After detecting UEs that have chosen the singleton CTUs (e.g., UE 1 and 5 shown as triangle in Fig. 3), the BS applies successive interference cancellation (SIC) technique to decode the data of these UEs. During the decoding, the UEs transmitting in different RBs do not interfere with each other due to the orthogonality, and only UEs transmitting in the same RB cause interference, i.e., as shown in Fig. 3, the interference UE set in RB 1 is $\{2, 3, 4, 7\}$ shown in color grey and the interference UE set in RB 2 is $\{1, 5, 6, 8\}$ shown in color yellow. In each iterative stage of SIC decoding, the CTU with the strongest received power is decoded by treating the received powers of other CTUs over the same RB as the interference. Each iterative stage of SIC decoding is successful when the signal-to-interference-plus-noise ratio (SINR) in that stage is larger than the SINR threshold γ_{th}. If the received signal is decoded successfully, the decoded signal is subtracted from the received signal[3]. Thus, in the kth repetition of the tth RTT, the sth stage of SIC decoding is successful if

$$\text{SINR}^t_{f,s}(k) = \frac{P h_{s,k} r_s^{-\eta}}{\sum\limits_{m=s+1}^{N^t_{f,sc}(k)} P_m h_{m,k} r_m^{-\eta} + \sum\limits_{n' \in \mathcal{N}^t_{f,cc}(k)} P_{n'} h_{n',k} r_{n'}^{-\eta} + \sigma^2} \geq \gamma_{th}, \tag{2}$$

where P is the transmission power, $\mathcal{N}^t_{f,sc}$ is the set of devices that have chosen the singleton CTUs over the fth RB, $\mathcal{N}^t_{f,cc}$ is the set of devices that have chosen the collision CTUs over the fth RB, σ^2 is the noise power.

5) HARQ Retransmissions: We take into account the GF-NOMA HARQ retransmissions to achieve high reliability. However, due to the latency constraint T_{cons}, the HARQ retransmission times are limited. The UE determines a re-transmission or not based on the following two different scenarios.

i) when the UE receives an ACK from the BS, it means that the BS successfully detected the UE (i.e., the UE choosing the singleton CTUs) and decoded the UE's data (i.e., SIC succeeds), no further re-transmission is needed;

ii) when the UE receives a NACK from the BS, it means that the BS successfully detected the UE but failed to decode the UE's data (i.e., SIC fails). Otherwise, when the UE does not receive any feedback at the pre-defined timing after the UE sent the packet (e.g., at the end of one RTT), it means the BS failed to identify the UE, the UE determines whether to retransmit or not based on the transmission latency check.

[2]A one-to-one mapping or a many-to-one mapping between the pilot sequences and codebooks can be predefined. Since it has been verified in [11] that the performance loss due to codebook collision is negligible for a real system, we focus on the pilot sequence collision and consider the one-to-one mapping as [5].

[3]We assume perfect SIC the same as [5], with no error propagation between iterations.

B. Problem Formulation

We focus the uplink contention-based GF-NOMA procedure over a set of preconfigured MA resources for UEs with latency constraint T_{cons} under the K-repetition GF scheme. Once actived in a given RTT t, a UE executes the GF-NOMA procedure, where the UE randomly chooses one of the preconfigured C^t CTUs to transmit its packets for K^t times. During this RTT, the GF-NOMA fails if: (i) a CTU collision occurs when two or more UEs choose the same CTU (i.e., UE detection fails); or (ii) the SIC decoding fails (i.e., data decoding fails). Once failed, UEs decides whether to retransmit in the following RTT or not based on the transmission latency check. When $T_{\text{late}} > T_{\text{cons}}$, the UE fails to be served and its packets will be dropped.

It is necessary to tackle the problem of optimizing the GF-NOMA configuration defined by parameters[4] $A^t = \{K^t, C^t\}$ for each RTT t, where K^t is the repetition value and C^t is the number of CTUs. At the beginning of each RTT t, the decision is made by the BS according to the transmission receptions $U^{t'}$ for all prior RTTs $t' = 1, ..., t-1$, consisting of the following variables: the number of the collision CTUs $V_{\text{cc}}^{t'}$, the number of the idle CTUs $V_{\text{ic}}^{t'}$, the number of the singleton CTUs $V_{\text{sc}}^{t'}$, the number of UEs that have been successfully detected and decoded under the latency constraint $V_{\text{sd}}^{t'}$, and the number of UEs that have been successfully detected but not successfully decoded $V_{\text{ud}}^{t'}$. We denote $H^t = \{O^1, O^2, ..., O^{t-1}\}$ with $O^{t-1} = \{U^{t-1}, A^{t-1}\}$ as the observation in each RTT t including histories of all such measurements and past actions.

At each RTT t, the BS aims at maximizing a long-term objective R_t (reward) related to the average number of UEs that have successfully send data under the latency constraint $V_{\text{sd}}^{t'}$ with respect to the stochastic policy π that maps the current observation history O^t to the probabilities of selecting each possible parameters in A^t. This optimization problem (P1) can be formulated as:

$$(\text{P1}:) \max_{\pi(A^t|O^t)} \quad \sum_{k=t}^{\infty} \gamma^{k-t} \mathbb{E}_{\pi}[V_{\text{sd}}^k] \tag{3}$$

$$s.t. \quad T_{\text{late}} \leq T_{\text{cons}}, \tag{4}$$

where $\gamma \in [0, 1)$ is the discount factor for the performance accrued in the future RTTs, and $\gamma = 0$ means that the agent just concerns the immediate reward.

III. Conventional Solutions

To simplify, we propose a load estimation-based uplink resource configuration (LE-URC) approach to dynamically configure the CTUs number C^t with the fixed repetition value K^t in each RTT to maximize the successfully served UEs without latency check and SIC procedure[5] described in Section II, which is expressed as

$$(\text{P2}:) \max_{\pi(C^t|O^t)} \mathbb{E}_{\pi}[V_{\text{sc}}^t], \tag{5}$$

At the RTT $t-1$, we consider that $D_{\text{UE}}^{t-1} = n$ UEs randomly choose one of C^{t-1} CTUs with an equal probability. The probability that no UE chooses a CTU c is

$$\mathbb{P}(D_c = 0 | D_{\text{UE}}^{t-1} = n) = (1 - 1/C^{t-1})^n. \tag{6}$$

The expected number of idle CTUs is given by

$$\mathbb{E}[V_{\text{ic}}^{t-1} | D_{\text{UE}}^{t-1} = n] = C^{t-1}(1 - 1/C^{t-1})^n. \tag{7}$$

Due to that the actual number of idle CTUs V_{ic}^{t-1} can be observed at the BS, the number of active UEs in the $(t-1)$th RTT is estimated as

$$\begin{aligned} \tilde{D}_{\text{UE}}^{t-1} &= f^{-1}(\mathbb{E}[V_{\text{ic}}^{t-1} | D_{\text{UE}}^{t-1} = n]) \\ &= \log_{(1-1/C^{t-1})}(V_{\text{ic}}^{t-1}/C^{t-1}). \end{aligned} \tag{8}$$

We use δ^t to represent the difference between the estimated numbers of UEs in the $(t-1)$th and the tth RTTs. That is $\delta^t = \tilde{D}_{\text{UE}}^t - \tilde{D}_{\text{UE}}^{t-1}$ and $\delta^t \approx \delta^{t-1}$ [12]. Therefore, the number of UEs in RTT t is estimated as

$$\tilde{D}_{\text{UE}}^t = \max\{2V_{\text{cc}}^{t-1}, \tilde{D}_{\text{UE}}^{t-1} + \delta^{t-1}\}, \tag{9}$$

where $2V_{\text{cc}}^{t-1}$ represents that there are at least $2V_{\text{cc}}^{t-1}$ number of UEs colliding in the last RTT.

Based on the estimated number of active UEs in the tth RTT $\tilde{D}_{\text{UE}}^t$, the expected number of the successfully served UEs in the tth RTT is given as

$$V_{\text{suss}}^t(C^t) = \mathbb{E}[V_{\text{sc}}^t | \tilde{D}_{\text{UE}}^t = n] = n(1 - 1/C^t)^{n-1}. \tag{10}$$

The maximal expected number of the successfully served UEs is obtained by choosing the number of CTUs as

$$C^{t*} = \arg\max_{C^t \in \mathcal{N}_{\text{CTU}}} V_{\text{suss}}^t(C^t). \tag{11}$$

IV. Cooperative Multi-agent DQN Approach

In this section, we propose a Cooperative Multi-Agent Deep Neural Network (DNN) based Q-learning (CMA-DQN) approach to optimize the configuration of both repetition value K^t and CTU numbers C^t simultaneously, which breaks down the selection in high-dimensional action space into multiple parallel sub-tasks.

Each DQN agent controls their own action variable, namely K^t or C^t, and receives a common reward to guarantee the objective in P1 cooperatively. We define A_x^t as the action selected by the xth agent. Each xth agent is responsible to update the value $Q(S^t, A_x^t)$ of action A_x^t in state S^t, where the state variable $S^t = [A^{t-1}, U^{t-1}, A^{t-2}, U^{t-2}, ..., A^{t-M_o}, U^{t-M_o}]$ only includes information about the last M_o RTTs. All agents receive the same reward $R^{t+1} = V_{\text{sd}}^t$ at the end of each RTT, where V_{sd}^t

[4]According to the UE detection and data decoding procedure described in Section II.A, for the same CTU number C^t, a large RB number F^t leads to fewer UEs in each RB, which increases the data decoding success probability. That is to say, the larger RB number, the better. Thus, we fix the RB number $F = 4$ in this work to optimize the CTU number.

[5]The UE is successfully transmitted if there is no CTU collision occurs. Thus, the optimization objective is V_{sc}^t.

is the observed number of successfully served UEs under the latency constraint T_{cons}.

The DQN agents are trained in parallel. Each agent x parameterizes the action-state value function $Q(S^t, A_x^t)$ by using a function $Q(S^t, A_x^t, \boldsymbol{\theta}_x)$, where $\boldsymbol{\theta}_x$ represents the weights matrix of a multiple layers DNN with fully-connected layers. The variables in the state S^t is fed in to the DNN as the input; the Rectifier Linear Units (ReLUs) are adopted as intermediate hidden layers; while the output layer is consisted of linear units, which are in one-to-one correspondence with all available actions in $\mathcal{A}$. The online update of weights matrix $\boldsymbol{\theta}_x$ is carried out along each training episode by using double deep Q-learning (DDQN). Accordingly, learning takes place over multiple training episodes, where each episode consists of several RTT periods. In each RTT, the parameters $\boldsymbol{\theta}_x$ of the Q-function approximator $Q(S^t, A_x^t, \boldsymbol{\theta}_x)$ are updated using RMSProp optimizer [13] as

$$\boldsymbol{\theta}_x^{t+1} = \boldsymbol{\theta}_x^t - \lambda_{\text{RMS}} \nabla L_x^{\text{DDQN}}(\boldsymbol{\theta}_x^t) \tag{12}$$

where $\lambda_{\text{RMS}} \in (0, 1]$ is RMSProp learning rate, $\nabla L_x^{\text{DDQN}}(\boldsymbol{\theta}_x^t)$ is the gradient of the loss function $L_x^{\text{DDQN}}(\boldsymbol{\theta}_x^t)$ used to train the state-action value function. The gradient of the loss function is defined as

$$\nabla L_x^{\text{DDQN}}(\boldsymbol{\theta}_x^t) = \text{E}_{S^i, A_x^i, R^{i+1}, S^{i+1}}[(R^{i+1} + \gamma \max_{a \in \mathcal{A}} Q(S^{i+1}, A_x^i, \bar{\boldsymbol{\theta}}_x^t) - Q(S^i, A_x^i, \boldsymbol{\theta}_x^t)) \nabla_{\boldsymbol{\theta}_x} Q(S^i, A_x^i, \boldsymbol{\theta}_x^t)], \tag{13}$$

where the expectation is taken over the minibatch, which are randomly selected from previous samples $(S_i, A_{ix}, S_{i+1}, R_{i+1})$ for $i \in \{t - M_r, ..., t\}$ with M_r being the replay memory size [14]. When $t - M_r$ is negative, it represents to include samples from the previous episode. Furthermore, $\bar{\boldsymbol{\theta}}^t$ is the target Q-network in DDQN that is used to estimate the future value of the Q-function in the update rule, and $\bar{\boldsymbol{\theta}}^t$ is periodically copied from the current value $\boldsymbol{\theta}^t$ and kept unchanged for several episodes. The detailed DQN algorithm is presented in **Algorithm 1**.

V. Simulation Results

We examine the effectiveness of our proposed GF-NOMA scheme with CMA-DQN algorithm via simulation. We adopt the standard network parameters listed in Table I following [7], and hyperparameters for the DQN learning algorithm are listed in Table II. All testing performance results are obtained by averaging over 1000 episodes. The DQN is set with two hidden layers, each with 128 ReLU units. Throughout epoch, each UE has a periodical a bursty traffic profile (i.e., the time limited Beta profile defined in [15, Section 6.1.1] with parameters (2, 4) that has a peak around the 4000th TTI.

Fig. 4 (a) shows the system convergence process of the proposed CMA-DQN by plotting the average reward. It can be intuitively seen that the proposed framework has a fast convergence speed and the episode required for system convergence is very small, even for heavy traffic scenarios (massive access).

Algorithm 1 CMA-DQN Based GF-NOMA Uplink Resource Configuration

Input: : Action space $\mathcal{A}$ and Operation Iteration I.

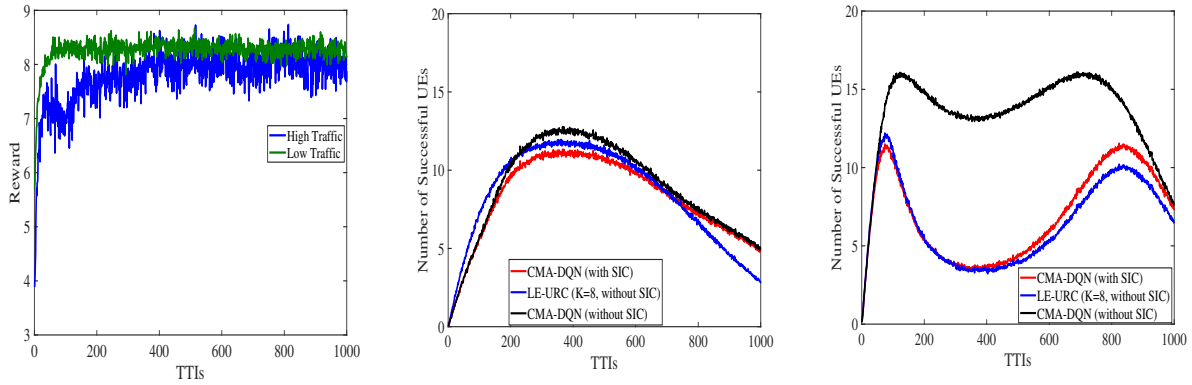
Algorithm hyperparameters: learning rate $\lambda_{RMS} \in (0, 1]$, discount rate $\gamma \in [0, 1)$, ϵ-greedy rate $\epsilon \in (0, 1]$, target network update frequency J;
Initialization of replay memory M to capacity D, the state-action value function $Q(S, A, \boldsymbol{\theta})$, the parameters of primary Q-network $\boldsymbol{\theta}$, and the target Q-network $\bar{\boldsymbol{\theta}}$;
for *Iteration* $\leftarrow$ *1 to I* **do**
  Initialization of S^1 by executing a random action A_x^0;
  for $t \leftarrow$ *1 to T* **do**
    if $p_\epsilon < \epsilon$ Then select a random action A_x^t from $\mathcal{A}_x$
    else select $A_x^t = \arg\max_{a \in \mathcal{A}_x} Q(S^t, A_x^t, \boldsymbol{\theta}_x)$. The BS broadcasts A_x^t and backlogged UEs attempt communication in the tth RTT;
    The BS observes state S^{t+1}, and calculate the related reward R^{t+1};
    Store transition $(S^t, A_x^t, R^{t+1}, S^{t+1})$ in replay memory M_x;
    Sample random minibatch of transitions $(S^t, A_x^t, R^{t+1}, S^{t+1})$ from replay memory M_x
    Perform a gradient descent step and update parameters $\boldsymbol{\theta}_x$ for $Q(S^t, A_x^t, \boldsymbol{\theta}_x)$ using (13);
    Update the parameter $\bar{\boldsymbol{\theta}} = \boldsymbol{\theta}$ of the target Q-network every J steps.
  end
end

TABLE I: Simulation Parameters

Parameters	Value	Parameters	Value
Path-loss exponent η	4	Noise power σ^2	-132 dBm
Transmission power P	23 dBm	The received SINR threshold γ_{th}	-10 dB
Duration of traffic T	2000 ms	The set of the repetition value	$\{1, 2, 4, 6, 8\}$
The set of the CTU number	$\{12, 24, 36, 48\}$	Latency constraint	2 ms
Bursty traffic parameter Beta(α,β)	(2, 4)	The number of bursty UEs N	10000 (light) / 30000 (heavy)
Cell radius	10 km	Duration of one TTI	0.125 ms

TABLE II: Learning Hyperparameters

Hyperparameters	Value	Hyperparameters	Value
Learning rate λ_{RMS}	0.0001	Minimum exploration rate ϵ	0.1
Discount rate γ	0.5	Minibatch size	32
Replay Memory	10000	Target Q-network update frequency	1000



(a) (b) (c)

Fig. 4: (a) Average received reward (b) Average number of successfully served UEs for light traffic (c) Average number of successfully served UEs for heavy traffic

Fig. 4 (b) and (c) plot the average number of successful served UEs by comparing the learning framework with the LE-URC approach. We first observe that since the LE-URC approach is not aware of the latency constraint and SIC procedure, the results are large at first, but still smaller than the number of non-collision UEs of CMA-DQN in heavy traffic scenarios. However, with increasing TTIs (above 700), the cumulated traffic increases due to unsuccessful transmissions and retransmissions, the LE-URC method becomes worse and achieve lower number of successful UEs than that of CMA-DQN due to its ignorance in latency constraint during its optimization for one time instance. The superior performance of CMA-DQN in heavy traffic scenarios also demonstrate its capability in dynamically configure lower repetition values and CTU numbers to alleviate the traffic congestion to obtain a long-term reward.

VI. Conclusions

In this paper, we developed a general learning framework for dynamic resource configuration optimization in signature-based GF-NOMA systems for mURLLC service under the K-repetition GF scheme. We designed a Cooperative Multi-Agent Deep Neural Network based Q-learning (CMA-DQN) approach to optimize the number of successfully served UEs under the latency constraint via adaptively configuring the repetition values and the contention-transmission unit (CTU) numbers. Our results have shown that: 1) the number of successfully served UEs under the same latency constraint in our proposed learning framework is up to three times more than that in the conventional load estimation-based approach (LE-URC); 2) the proposed CMA-DQN is superior to LE-URC in its capability in dynamically configuring for mURLLC in heavy traffic scenarios in long term; and 3) the proposed learning framework can be used to optimize the other resource configuration problems in GF-NOMA schemes, such as retransmission times, starting offset of the grant, and etc.

References

[1] X. Zhang, J. Wang, and H. V. Poor. Statistical delay and error-rate bounded QoS provisioning for mURLLC over 6G CF M-MIMO mobile networks in the finite blocklength regime. *IEEE J. Sel. Areas Commun.*, pages 1–1, Sep. 2020.

[2] W. Saad, M. Bennis, and M. Chen. A vision of 6G wireless systems: Applications, trends, technologies, and open research problems. *IEEE Network*, 34(3):134–142, May. 2020.

[3] Study on scenarios and requirements for next generation access technologies. *3GPP, TR 38.913 v16.0.0*, Jul. 2020.

[4] Study on physical layer enhancements for NR ultra-reliable and low latency case (URLLC). *3GPP, TR 38.824 v16.0.0*, Mar. 2019.

[5] R. Abbas, M. Shirvanimoghaddam, Y. Li, and B. Vucetic. A novel analytical framework for massive grant-free NOMA. *IEEE Trans. Commun.*, 67(3):2436–2449, Mar. 2019.

[6] M. B. Shahab, R. Abbas, M. Shirvanimoghaddam, and S. J. Johnson. Grant-free non-orthogonal multiple access for IoT: A survey. *IEEE Commun. Surveys Tutorials*, pages 1–1, May. 2020.

[7] Study on new radio access technology-physical layer aspects. *3GPP, TR 38.802 v14.0.0*, Mar. 2017.

[8] Cellular system support for ultra-low complexity and low throughput Internet of Things (CIoT). *3GPP, Sophia Antipolis, France, TR 45.820 V13.1.0,*, Nov. 2015.

[9] On MA resource and MA signature configurations. *R1-1609227, 3GPP TSG-RAN WG1 #86*, Oct. 2016.

[10] Y. Liu, Y. Deng, M. Elkashlan, A. Nallanathan, and G. K. Karagiannidis. Analyzing grant-free access for URLLC service. *IEEE J. Sel. Areas Commun.*, pages 1–1, Aug. 2020.

[11] J. Zhang, L. Lu, Y. Sun, Y. Chen, J. Liang, J. Liu, H. Yang, S. Xing, Y. Wu, J. Ma, I. B. F. Murias, and F. J. L. Hernando. PoC of SCMA-based uplink grant-free transmission in UCNC for 5G. *IEEE J. Sel. Areas Commun.*, 35(6):1353–1362, Jun. 2017.

[12] S. Duan, V. Shah-Mansouri, Z. Wang, and V. W. S. Wong. D-ACB: Adaptive congestion control algorithm for bursty M2M traffic in LTE networks. *IEEE Trans. Veh. Technol.*, Dec. 2016.

[13] Tijmen Tieleman and Geoffrey Hinton. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural Netw. Mach. Learn.*, 4(2):26–31, Oct. 2012.

[14] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[15] Study on RAN improvements for machine-type communications. *3GPP, TR 37.868 v11.0.0*, Sep. 2011.