Multiple Configured-Grants Optimization in Grant-Free NOMA for mURLLC Service

Yan Liu*†, Yansha Deng‡, Maged Elkashlan†, Arumugam Nallanathan† and George K. Karagiannidis§
*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
§Department of Electrical and Computer Engineering, Aristotle University of Thessaloniki, Greece

ICC 2022 - IEEE International Conference on Communications | 978-1-5386-8347-7/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICC45855.2022.9838541

***Abstract*—Realizing efficient, delay-bounded, and reliable communications for a massive number of user equipments (UEs) in massive Ultra-Reliable and Low-Latency Communications (mURLLC) is extremely challenging as it needs to simultaneously take into account the latency, reliability, and massive access requirements. To support these requirements, the third generation partnership project (3GPP) has introduced grant-free non-orthogonal multiple access (GF-NOMA) with multiple configured-grants (MCGs), where UE can choose any of these grants as soon as the data arrives. In this paper, we develop a novel learning framework for MCG-GF-NOMA systems. We first design the MCG-GF-NOMA model by characterizing each CG. We then formulate the MCG-GF-NOMA resources configuration problem taking into account three constraints. Finally, we propose a Cooperative Multi-Agent based Double Deep Q-Network (CMA-DDQN) algorithm to allocate the channel resources among MCGs to maximize the number of successful transmissions under the latency constraint. Our results show that the MCG-GF-NOMA framework can simultaneously improve the low latency and high reliability performances for mURLLC.**

***Index Terms*—Multiple configured-grants, massive URLLC, NOMA, deep reinforcement learning, resource configuration.**

I. Introduction

Massive URLLC (mURLLC), which integrates Ultra-Reliable and Low-Latency Communications (URLLC) with massive access, is becoming a new and important service class in the next generation (6G) for the time-sensitive traffics and has received tremendous research attention [1]. However, addressing the need in mURLLC is fundamentally challenging as it needs to simultaneously guarantee the latency, reliability, and massive access requirements. To support these requirements, several new features were standardized by the third generation partnership project (3GPP).

1) To reduce the latency, the grant-free (GF) (a.k.a. configured-grant (CG)) transmission is proposed in 3GPP Release 15 [2] as an alternative for traditional grant-based (GB) (a.k.a. dynamic-grant (DG)) in Long Term Evolution (LTE). In GF transmission, the User Equipment (UE) is allowed to transmit data to the Base Station (BS) in an arrive-and-go manner without scheduling request and uplink (UL) resource grant to reduce latency.

2) To increase the reliability, the K-repetition GF transmission has been proposed by 3GPP, where a pre-defined number of consecutive replicas of the same packet are transmitted in the consecutive time slots [2]. More details about K-repetition GF transmission can be found in [3].

3) To mitigate the serious transmission delay and network congestion problems caused by collision events in contention-based GF transmission and enhance the UL connectivity, non-orthogonal multiple access (NOMA) has been proposed to synergize with GF transmission [4], where GF-NOMA allows multiple UEs to transmit over the same physical resource by employing user-specific signature patterns (e.g, codebook, pilot sequence, demodulation reference signal, power, etc.) [4].

4) To support different starting offsets of the resources with respect to UL packet arrival time, 3GPP proposed multiple CGs (MCG) transmission in Release 16 [5]. On the one hand, there is a chance of reducing the latency in cases where the data of an UE arrives after the starting slot offset of one CG. On the other hand, there is a chance of mitigating the collision events when multiple UEs are active and waiting for the CG period to transmit the packet. MCGs also support different resource sizes, repetitions, and periodicity, to suit different data requirements, respectively [6].

As mentioned before, research on the MCG-GF-NOMA networks to support mURLLC is fundamental and essential, which is an untreated and challenging problem. To cope with it, accurately modeling, analyzing, and optimizing the MCG-GF-NOMA resource is fundamentally important. In this paper, we address the following fundamental questions: 1) how to design the MCG-GF-NOMA network; 2) how to quantify the performances of the MCG-GF-NOMA network; 3) how to formulate the MCG-GF-NOMA resources configuration problem; and 4) how to balance the allocations of channel resources among MCGs so as to provide maximum successful transmissions in mURLLC service with bursty traffic. The main contributions of this paper are as follows:

- We develop a novel MCG-GF-NOMA learning framework for attaining the long-term successfully served UEs under the latency constraint in mURLLC service. In this framework, we design a MCG-GF-NOMA system, where we characterize each CG using the parameters including the number of contention-transmission units (CTUs), the starting slot of each CG within a subframe, and the number of repetitions of each CG. We then formulate the

MCG-GF-NOMA resource configuration problem taking into account three constraints.

- We propose a Cooperative Multi-Agent learning technique based Double Deep Q-Network (CMA-DDQN) algorithm to balance the allocations of resources among MCGs so as to maximize the number of successful transmissions under the latency constraint, which breaks down the selection of high-dimensional parameters into multiple parallel sub-tasks with a number of agents cooperatively being trained to produce each parameter.
- Our results show that the MCG-GF-NOMA learning framework can improve the mURLLC performances. First, the number of successfully served UEs in the MCG-GF-NOMA system is up to four times more than that in the SCG-GF-NOMA system. Second, the MCG-GF-NOMA can also increase the CTU resource utilization efficiency compared to the SCG-GF-NOMA system.

The remainder of this paper is structured as follows. Section II illustrates the system model of MCG-GF-NOMA system. Section III describes the problem analysis and formulation. Section IV elaborates on the proposed CMA-DDQN algorithm for solving the formulated problem. The simulation results are illustrated in Section V. Finally, Section VI concludes the main concept, insights and results of this paper.

II. System Model

We consider a single-cell UL wireless network with a coverage radius of R. Particularly, a BS is located at the center of the cell, and a number of N_{UE} static UEs are randomly distributed around the BS in an area of the plane $\mathbb{R}^2$. The BS is unaware of the status of these UEs, hence no UL channel resource is scheduled to them in advance. 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. In addition, we consider a Rayleigh flat-fading environment, where the channel power gains h are exponentially distributed (i.i.d.) random variables with unit mean.

We consider the MCG-GF-NOMA system as shown in Fig. 1. The BS configures N_{CG} UL CGs at each subframe. The UE chooses the configuration with the earliest starting point to transmit data. The smallest transmission unit that a UE can compete for is called a contention transmission unit (CTU). A CTU may comprise of a MA physical resource and a MA signature [7]. Without loss of generality, we consider that there are N_{CTU} unique CTUs over F time-frequency RBs configured by the BS in each CG configuration period. Each CG is consist of different resources in the CTU domains, and is associated with the following transmission parameters:

- Number of CTUs (N_{CTU})
- Starting slot within a subframe (N_{start})
- Number of repetitions (N_{repe})
- Number of slots in a subframe (N_{slot})

Without loss of generality, we consider the number of slots in each subframe N_{slot} is the same for each CG. Thus,

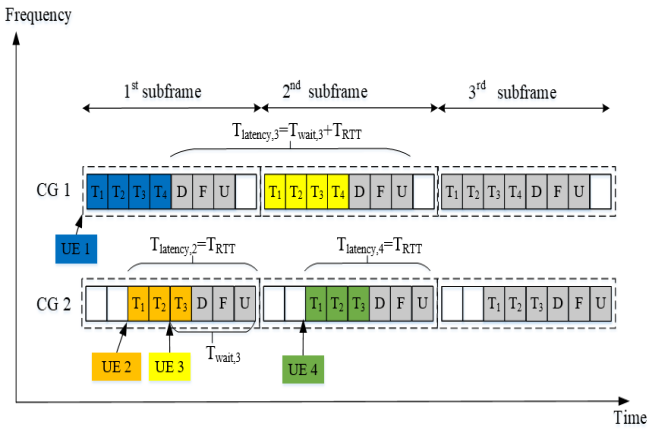


Fig. 1: Multiple CGs (MCG) configurations for K-repetition GF transmission, T: packet transmission, D: DL processing, F: ACK/NACK feedback, and U: UL processing.

for ease of presentation, we represent each CG_i in the tth subframe by $\mathrm{CG}_i^t\{N_{\mathrm{CTU},i}^t, N_{\mathrm{start},i}^t, N_{\mathrm{repe},i}^t\}$. As illustrated in Fig. 1, $\mathrm{CG}_1^t\{1,0,4\}$ and $\mathrm{CG}_2^t\{1,2,3\}$ are two CGs in the 1st subframe.

III. Problem Analysis and Formulation

In a given subframe t, the BS preconfigured N_{CG} CGs for UEs to transmit data. As soon as the URLLC data arrives, a UE can choose the CG_i^t with the earliest starting point (i.e., the smallest $N_{\mathrm{start},i}^t$) to transmit data. Suppose that the UE choose the $\mathrm{CG}_i^t\{N_{\mathrm{CTU},i}^t, N_{\mathrm{start},i}^t, N_{\mathrm{repe},i}^t\}$, then the UE randomly choose a CTU from $N_{\mathrm{CTU},i}^t$ available CTUs and start transmit at slot $N_{\mathrm{start},i}^t$ for $N_{\mathrm{repe},i}^t$ repetitions. The BS decodes each repetition independently and the transmission is successful when at least one repetition succeeds. After processing all the received $N_{\mathrm{repe},i}^t$ repetitions, the BS transmits the ACK/NACK feedback to the UE. Considering the small packets for URLLC, we set the packet transmission time as one TTI. The feedback time and processing time are also assumed to be one TTI like our previous work [3].

A. MCG-GF-NOMA Reliability Analysis

During each RTT, if the GF-NOMA procedure fails, the UE fails to be served and its packets will be dropped. 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 successive interference cancellation (SIC) decoding fails (i.e., data decoding fails).

1) CTU dectection: At each RTT, each active UE transmits its packets to the BS by randomly choosing a CTU from the earliest CG_i. The BS can detect the UEs that have chosen different CTUs. However, if multiple UEs choose the same CTU, the BS cannot differentiate these UEs and therefore cannot decode the data. We categorize the CTUs from each CG_i into three types [8]:

- *idle* CTU: a CTU which has not been chosen by any UE;

- *singleton* CTU : a CTU chosen by only one UE;
- *collision* CTU : a CTU chosen by two or more UEs.

After collision detection at the tth subframe for the CG_i, the BS observes the set of singleton CTUs $\mathcal{N}^t_{\mathrm{SC},i}$, the set of idle CTUs $\mathcal{N}^t_{\mathrm{IC},i}$, and the set of collision CTUs $\mathcal{N}^t_{\mathrm{CC},i}$ for each CG_i.

2) SIC decoding: After detecting UEs that have chosen the singleton CTUs, the BS performs the SIC to decode the data of these UEs. Based on the NOMA principles, at each iterative stage of SIC, the BS first decodes the UE with the strongest received power and then subtracted the successfully decoded signal from the received signal (we assume perfect SIC the same as [8]). It worth noting that during the decoding, the UEs that transmit on different RBs do not interfere with each other due to the orthogonality, and only UEs that transmit on the same RB cause interference. Thus, in order to characterize the UEs transmitting with CG_i on the fth RB, we represent the $\mathcal{N}^t_{f,\mathrm{SU},i}$ as the set of UEs that have chosen the singleton CTUs for the CG_i on the fth RB, the $N^t_{f,\mathrm{SU},i} = \left|\mathcal{N}^t_{f,\mathrm{SU},i}\right|$ as the number of UEs that have chosen the singleton CTUs for the CG_i on the fth RB ($|\cdot|$ denotes the number of elements in any vector $\cdot$), and $N^t_{f,\mathrm{CU},i}$ as the number of UEs that have chosen the collision CTUs using the CG_i on the fth RB. We define the received power of the sth UE in the nth repetition of the CG_i on the fth RB as

$$P^t_{s,f,i} = P h^t_{s,f,i} {r_s}^{-\eta}, \tag{1}$$

where P is the transmission power, r is the Euclidean distance between the UE and the BS, η is the path-loss attenuation factor, h is the Rayleigh fading channel power gain from the UE to the BS.

Suppose that the received power obeys $P^t_{1,f,i} \geq P^t_{2,f,i} \geq ... \geq P^t_{N^t_{f,\mathrm{SU},i}}$, the decoding order should be from the 1st UE to the $N_{f,\mathrm{SU},i}$th UE. 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. Thus, at the tth subframe, in the nth repetition of the CG_i on the fth RB, the signal-to-interference-plus-noise ratio (SINR) of the sth stage of SIC decoding of the sth UE is derived as

$$\mathrm{SINR}^t_{s,f,i} = \frac{P^t_{s,f,i}}{\sum\limits_{m=s+1}^{N^t_{f,SU,i}} P^t_{m,f,i} + \sum\limits_{n'=1}^{N^t_{f,CU,i}} P^t_{n',f,i} + \sigma^2}, \tag{2}$$

where σ^2 is the noise power.

Each iterative stage of SIC decoding is successful when the SINR in that stage is larger than the SINR threshold, i.e., $\mathrm{SINR}^t_{s,f,i} \geq \gamma_{th}$. The SIC procedure stops when one iterative stage of the SIC fails or when there are no more signals to decode. The SIC decoding procedure for each CG_i is described in the following.

- Step 1: Start the nth repetition with the initial $n = 1$, $\mathcal{N}^t_{f,\mathrm{SU},i}$, $N^t_{f,\mathrm{SU},i}$ and $N^t_{f,\mathrm{CU},i}$;
- Step 2: Decode the sth UE with the initial $s = 1$ using (2);
- Step 3: If the sth UE is successfully decoded, put the decoded UE in set $\mathcal{N}^t_{f,\mathrm{suc},i}(n)$ and go to Step 4, otherwise go to Step 5;
- Step 4: If $s \leq N^t_{f,\mathrm{SU},i}$, do $s = s + 1$, go to Step 2, otherwise go to Step 5;
- Step 5: SIC for the nth repetition stops;
- Step 6: If $n \leq N_{\mathrm{repe},i}$, do $n = n + 1$, go to Step 1, otherwise go to the end.

Finally, the set of successfully served UEs using the CG_i on the fth RB at the tth subframe is derived as

$$\mathcal{N}^t_{f,\mathrm{suc},i} = \bigcup_{n=1}^{N_{\mathrm{repe},i}} (\mathcal{N}^t_{f,\mathrm{suc},i}(n)), \tag{3}$$

the set of the successfully served UEs using the CG_i at the tth subframe is obtained as

$$\mathcal{N}^t_{\mathrm{suc},i} = \bigcup_{f=1}^{F^t} (\mathcal{N}^t_{f,\mathrm{suc},i}), \tag{4}$$

and the set of the successfully served UEs at the tth subframe is obtained as

$$\mathcal{N}^t_{\mathrm{suc}} = \bigcup_{i=1}^{N_{\mathrm{CG}}} (\mathcal{N}^t_{\mathrm{suc},i}). \tag{5}$$

Then, $N^t_{\mathrm{suc}} = |\mathcal{N}^t_{\mathrm{suc}}|$ is the number of successfully served UEs.

B. Problem Formulation

In this work, we aim to tackle the problem of optimizing the MCG-GF-NOMA configuration defined by parameters $\mathrm{CG}^t_i\{N^t_{\mathrm{CTU},i}, N^t_{\mathrm{start},i}, N^t_{\mathrm{repe},i}\}$ for each subframe t. At each subframe t, the BS aims at maximizing a long-term objective R_t related to the average number of UEs that have successfully send data 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:

$$(\mathrm{P1}:) \max_{\pi(A^t|O^t)} \quad \sum_{k=t}^{\infty} \gamma^{k-t} \mathbb{E}_\pi[N^k_{\mathrm{suc}}] \tag{6}$$

$$s.t. \quad \sum_{i=1}^{N_{\mathrm{CG}}} N^t_{\mathrm{CTU},i} = N^t_{\mathrm{CTU,SCG}}, \tag{7}$$

$$N^t_{\mathrm{start},i} + N^t_{\mathrm{repe},i} + 3 = N_{\mathrm{slot}}, \forall i \in [1, N_{\mathrm{CG}}], \tag{8}$$

$$N^t_{\mathrm{start},i} < N^t_{\mathrm{start},i+1} < N_{\mathrm{slot}} - 3, \forall i \in [1, N_{\mathrm{CG}}], \tag{9}$$

where $\gamma \in [0, 1)$ is the discount factor for the performance accrued in the future subframes, and $\gamma = 0$ means that the agent just concerns the immediate reward. The CTU resource constraint in (7) is set to compare with the SCG-GF-NOMA scheme, where $N^t_{\mathrm{CTU,SCG}}$ is the configured CTU numbers for the SCG-GF-NOMA. That is to say, the MCG-GF-NOMA configuration uses the same frequency resources but overlap in time and have different starting points so they do not require the additional resources compared to the conventional SCG-GF-NOMA scheme. The latency constraint in (8) is set to

satisfy the latency requirement. That is to say, the transmission must be completed in one subframe (1 ms). Otherwise, the packet will be dropped. The starting slot constraint in (9) is set to support different UL packet arrival times.

IV. Proposed Optimization Solution

In this section, we propose a Cooperative Multi-Agent Double Deep Q-Network (CMA-DDQN) approach to tackle the problem (P1), which breaks down the selection in high-dimensional action space into multiple parallel sub-tasks. The state space, action space, reward function design of the proposed CMA-DDQN based algorithm are specified.

A. Reinforcement Learning Framework

We define $S \in \mathcal{S}$, $A \in \mathcal{A}$, and $R \in \mathcal{R}$ as any state, action, and reward from their corresponding sets, respectively. At the beginning of each subframe t, the RL-agent first observes the current state S^t corresponding to a set of previous observations $U^{t'}$ for all prior subframes ($t' = 1, ..., t-1$) in order to select an specific action $A^t \in \mathcal{A}(S^t)$. After carrying out the action A^t, the RL-agent transits to a new observed state S^{t+1} and obtains a corresponding reward R^{t+1} as the feedback from the environment, which is designed based on the new observed state S^{t+1} and guides the agent to achieve the optimization goal. After enough iterations, the BS can learn the optimal policy that maximizes the long-term rewards.

The detailed descriptions of the state, action and reward of problem (P1) are introduced as follows.

1) States in the Q-learning Model: In terms of the state space of the proposed CMA-DDQN model, it contains five parts: the number of the collision CTUs $N_{\mathrm{CC}}^{t'}$, the number of the idle CTUs $N_{\mathrm{IC}}^{t'}$, the number of the singleton CTUs $N_{\mathrm{SC}}^{t'}$, the number of UEs that have been successfully detected and decoded under the latency constraint $N_{\mathrm{suc}}^{t'}$, and the number of UEs that have been successfully detected but not successfully decoded $N_{\mathrm{fdec}}^{t'}$.

2) Actions in the Q-learning Model: Practically, the MCG-GF-NOMA system is always configured with multiple CGs to serve UEs with random traffic. In this section, we study the problem (P1) of optimizing the resource configuration for multiple CGs each with parameters $CG^t = \{N_{\mathrm{CTU},i}^t, N_{\mathrm{start},i}^t, N_{\mathrm{repe},i}^t\}_{i=1}^{N_{\mathrm{CG}}}$, where $N_{\mathrm{CTU},i}^t$ is chosen from the set of the number of the CTUs $\mathcal{N}_{\mathrm{CTU}}$, $N_{\mathrm{start},i}^t$ is chosen from the set of the value of the repetitions $\mathcal{N}_{\mathrm{start}}$, and $N_{\mathrm{repe},i}^t$ is chosen from the set of the value of the repetitions $\mathcal{N}_{\mathrm{repe}}$. This joint optimization by configuring each parameter in each CG can improve the overall data transmission performance. However, considering multiple CGs results in the increment of observations space, which exponentially increases the size of state space. For example, the number of available actions corresponds to the possible combinations of configurations $|\mathcal{A}| = \prod_{i=1}^{N_{\mathrm{CG}}} (|\mathcal{N}_{\mathrm{CTU},i}| \times |\mathcal{N}_{\mathrm{start},i}| \times |\mathcal{N}_{\mathrm{repe},i}|)$. To train Q-agent with this expansion, the requirements of time and computational resources greatly increase. In view of this, we revise the configured parameters by considering the constraints from (7) to (9).

First, considering the CTU resource constraint $\sum_{i=1}^{N_{\mathrm{CG}}} N_{\mathrm{CTU},i}^t = N_{\mathrm{CTU,SCG}}^t$ as presented in (7), we could obtain the action set $\mathcal{A}_{\mathrm{CTU}}^t$, which consists of the actions $A_{\mathrm{CTU}}^t \in \mathcal{A}_{\mathrm{CTU}}^t$ with $A_{\mathrm{CTU}}^t = \{N_{\mathrm{CTU},1}^t, ..., N_{\mathrm{CTU},N_{\mathrm{CG}}}^t\}$. In addition, considering the starting slot constraint $N_{\mathrm{start},i}^t < N_{\mathrm{start},i+1}^t < N_{\mathrm{slot}} - 3, \forall i \in [1, N_{\mathrm{CG}}]$ in (9), we could obtain the action set $\mathcal{A}_{\mathrm{start}}^t$, which consists of the actions $A_{\mathrm{start}}^t \in \mathcal{A}_{\mathrm{start}}^t$ with $A_{\mathrm{start}}^t = \{N_{\mathrm{start},1}^t, ..., N_{\mathrm{start},N_{\mathrm{CG}}}^t\}$. According to the latency constraint in (8), we have $N_{\mathrm{repe},i}^t = N_{\mathrm{slot}} - 3 - N_{\mathrm{start},i}^t, \forall i,$. Therefore, two actions set $\mathcal{A}_{\mathrm{CTU}}^t$ and $\mathcal{A}_{\mathrm{start}}^t$ is enough to characterize the multiple CG configurations defined by parameters $\mathrm{CG}_i^t\{N_{\mathrm{CTU},i}^t, N_{\mathrm{start},i}^t, N_{\mathrm{repe},i}^t\}$.

3) Reward Function in the Q-Learning Model: As the optimization goal is to maximize the number of the successfully served UEs under the latency constraint, we define the reward R^{t+1} as

$$R^{t+1} = N_{\mathrm{suc}}^t, \tag{10}$$

where N_{suc}^t is the number of UEs that have been successfully detected and decoded under the latency constraint.

B. Cooperative Multi-Agent DDQN Approach

A large number of actions and states will inevitably result in massive computation latency and severely affect the performance of the RL algorithm. To address this issue, deep reinforcement learning (DRL) is introduced, where DRL can directly control the behavior of each agent and solve complex decision-making problems, through interaction with the environment [9]. In addition, Multi-Agent RL (MA-RL) is introduced with centralized or decentralized rewards. To convert this selfishness into cooperative behavior, the same reward may be assigned to all agents [10]. In this section, we apply the Cooperative Multi-Agent technique based DDQN (CMA-DDQN) to prevent the selfish behavior of agents.

The CMA-DDQN algorithm utilizes the experience replay technique to enhance the convergence performance of RL. When updating the CMA-DDQN algorithm, mini-batch samples are selected randomly from the experience memory as the input of the neural network, which breaks down the correlation among the training samples. In addition, through averaging the selected samples, the distribution of training samples can be smoothed, which avoids the training divergence. We define A_x^t as the action selected by the xth agent. Each xth agent is responsible for updating 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} at the end of each subframe.

The DDQN 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 into 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 DDQN [11]. 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 [12] as

$$\boldsymbol{\theta}_x^{t+1} = \boldsymbol{\theta}_x^t - \lambda_{\mathrm{RMS}} \nabla L_x^{\mathrm{DDQN}}(\boldsymbol{\theta}_x^t) \tag{11}$$

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

$$\begin{aligned} &\nabla L_x^{\mathrm{DDQN}}(\boldsymbol{\theta}_x^t) \\ &= \mathrm{E}_{S^j, A_x^j, R^{j+1}, S^{j+1}}[(R^{j+1} + \gamma \max_{a \in \mathcal{A}} Q(S^{j+1}, A_x^j, \bar{\boldsymbol{\theta}}_x^t) \\ &- Q(S^j, A_x^j, \boldsymbol{\theta}_x^t)) \nabla_{\boldsymbol{\theta}_x} Q(S^j, A_x^j, \boldsymbol{\theta}_x^t)], \end{aligned} \tag{12}$$

where the expectation is taken over the minibatch, which are randomly selected from previous samples $(S^j, A_x^j, S^{j+1}, R^{j+1})$ for $j \in \{t - M_r, ..., t\}$ with M_r being the replay memory size [9]. 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.

Through calculating the expectation of the selected previous samples in minibatch and updating the $\boldsymbol{\theta}^t$ by (11), the DDQN value function $Q(s, a, \boldsymbol{\theta})$ can be obtained. The detailed CMA-DDQN algorithm is presented in **Algorithm 1.** We consider ϵ-greedy approach to balance exploitation and exploration in the actor of the Q-Agent, where ϵ is a positive real number and $\epsilon < 1$. In each subframe t, the Q-agent randomly generates a probability $P_\epsilon^t$to compare with ϵ. Then, with the probability ϵ, the algorithm randomly chooses an action from the remaining feasible actions to improve the estimate of the non-greedy action's value. With the probability $1-\epsilon$, the algorithm exploits the current knowledge of the Q-value table to choose the action that maximizes the expected reward.

V. Simulation Results

In this section, we examine the effectiveness of our proposed MCG-GF-NOMA system with CMA-DDQN algorithm via simulation. We adopt the standard network parameters listed in Table I following [13], and hyperparameters for the DQN learning algorithm are listed in Table II. All testing performance results are obtained by averaging over 1000 episodes. The BS is located at the center of a circular area with a 10 km radius, and the UEs are randomly located within the cell. The DQN is set with two hidden layers, each with 128 ReLU units. In the following, we present our simulation results

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

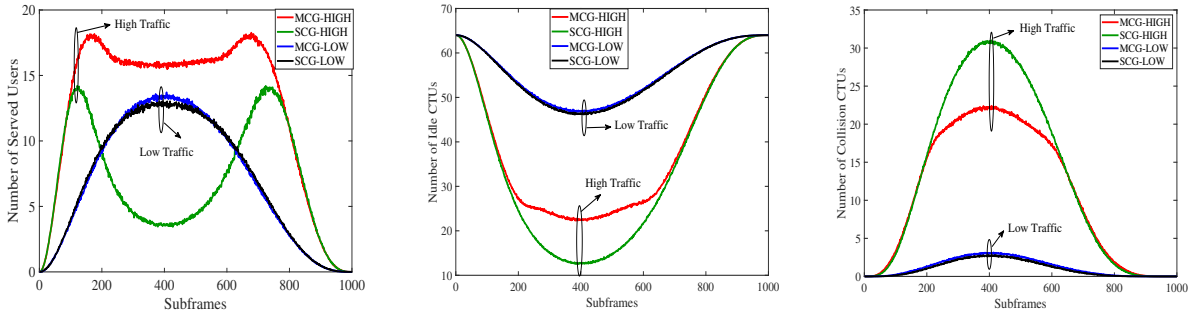
Input: : Action space $\mathcal{A}$ and Operation Iteration I.
Algorithm hyperparameters: learning rate $\lambda_{RMS} \in (0, 1]$, discount rate $\gamma \in [0, 1)$, ϵ-greedy rate $\epsilon \in (0, 1]$, target network update frequency Y;
Initialization of replay memory M to capacity D, the state-action 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 subframe;
    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 (12);
    Update the parameter $\bar{\boldsymbol{\theta}} = \boldsymbol{\theta}$ of the target Q-network every Y steps.
  end
end

TABLE I: Simulation Parameters

Parameters	Value	Parameters	Value
Number of symbols in a slot N_{sym}	7	Number of static UEs N_{UE}	10000(low)/ 50000(high)
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	1000 ms	Number of CTUs for the SCG-GF-NOMA $N_{\mathrm{CTU,SCG}}$	64
The set of the number of CTUs $\mathcal{N}_{\mathrm{CTU}}$	$\{8, 16, 24, 32$ $40, 48, 56\}$	The set of the starting slot $\mathcal{N}_{\mathrm{start}}$	$\{0, 1, 2, 3, 4\}$
Number of time-frequency RBs F	4	Number of slots in a subframe N_{slot}	8

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. 2: (a) Average number of served UEs (b) Average number of idle CTUs (c) Average number of collision CTUs

of multiple CG configurations in MCG-GF-NOMA system. Throughout epoch, each UE has a bursty traffic profile (i.e., the time limited Beta profile defined in [14, Section 6.1.1] with parameters (3, 4)) that has a peak around the 400th subframe.

Fig. 2 (a) compares the number of successfully served UEs in MCG-GF-NOMA and SCG-GF-NOMA systems with both high traffic and low traffic scenarios, respectively. Unless otherwise stated, we consider $N_{\mathrm{CG}} = 5$ for the MCG-GF-NOMA system. It is obvious that the MCG-GF-NOMA can increase the successfully served UEs compared with the SCG-GF-NOMA, especially for the high traffic scenario (i.e., massive access simultaneously). Particularly, at the peak traffic, the number of successfully served UEs in the MCG-GF-NOMA system is circa four times more than that in the SCG-GF-NOMA system. However, in low traffic scenario, this advantage of MCG is not obvious. This indicates that the MCG solution can ensure the massive access performance of GF-NOMA in a massive URLLC scenario.

Fig. 2 (b) and (c) compare the average number of idle and collision CTUs in MCG-GF-NOMA and SCG-GF-NOMA systems with both high traffic and low traffic scenarios, respectively. We observe that the multiple CGs solution can obtain better reliability performance of MCG-GF-NOMA only by using smaller CTU resources than the SCG-GF-NOMA, especially for the high traffic scenario. This is due to the fact that the MCG solution mitigates the heavy traffic backlog in the SCG-GF-NOMA system, where multiple UEs are active after the starting slot offset of one CG will wait for the next CG period to transmit the packet. Consequently, the collision events are mitigated in the MCG-GF-NOMA system.

VI. Conclusion

In this paper, we proposed a novel MCG-GF-NOMA learning framework for attaining the long-term successfully served UEs under the latency constraint in mURLLC service, where bursty traffic of UEs was considered. We first designed and modeled the MCG-GF-NOMA system, where we characterize each CG using the parameters including the number of CTUs, the starting slot of each CG within a subframe, and the number of repetitions of each CG. We then formulated the MCG-GF-NOMA resources configuration problem taking into account three constraints. Finally, we proposed a CMA-DDQN algorithm to balance the allocations of resources among MCGs so as to maximize the number of successful transmissions under the latency constraint, which breaks down the selection of high-dimensional parameters into multiple parallel subtasks with a number of DDQN agents cooperatively being trained to produce each parameter. Our results have shown that the MCG-GF-NOMA framework can improve the realibity performances for mURLLC. In detail, the number of successfully served UEs in the MCG-GF-NOMA system is circa four times more than that in the SCG-GF-NOMA system.

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.*, pp. 1–1, Sep. 2020.

[2] "5G; NR; physical layer procedures for data," *3GPP TS 38.214 v15.9.0*, Mar. 2020.

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

[4] 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*, pp. 1–1, May. 2020.

[5] T.-K. Le, U. Salim, and F. Kaltenberger, "An overview of physical layer design for ultra-reliable low-latency communications in 3GPP releases 15, 16, and 17," *IEEE Access*, vol. 9, pp. 433–444, Dec. 2020.

[6] "Enhanced UL configured grant transmissions for URLLC," *R1-1906151, 3GPP TSG RAN WG1 #97*, May. 2019.

[7] N. Ye, H. Han, L. Zhao, and A.-H. Wang, "Uplink nonorthogonal multiple access technologies toward 5G: A survey," *Wireless Commun. Mobile Comput.*, vol. 2018, Jun. 2018.

[8] R. Abbas, M. Shirvanimoghaddam, Y. Li, and B. Vucetic, "A novel analytical framework for massive grant-free NOMA," *IEEE Trans. Commun.*, vol. 67, no. 3, pp. 2436–2449, Mar. 2019.

[9] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[10] L. Liang, H. Ye, and G. Y. Li, "Spectrum sharing in vehicular networks based on multi-agent reinforcement learning," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 10, pp. 2282–2292, Aug. 2019.

[11] H. Van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double q-learning," *arXiv preprint arXiv:1509.06461*, Dec. 2015.

[12] T. Tieleman and G. Hinton, "Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude," *COURSERA: Neural Netw. Mach. Learn.*, vol. 4, no. 2, pp. 26–31, Oct. 2012.

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

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