Multi-Agent Reinforcement Learning-Based Fairness-Aware Scheduling for Bursty Traffic

Mingqi Yuan^{1,2}, Qi Cao^{1}, Man-On Pun$^{\dagger 1,2}$ and Yi Chen^{1,2}
^{1}School of Science and Engineering, The Chinese University of Hong Kong, Shenzhen, China, 518172
^{2}Shenzhen Research Institute of Big Data, Shenzhen, China, 518172

GLOBECOM 2021 - 2021 IEEE Global Communications Conference | 978-1-7281-8104-2/21/$31.00 ©2021 IEEE | DOI: 10.1109/GLOBECOM46510.2021.9685661

Abstract—**In this work, we develop practical user scheduling algorithms for downlink bursty traffic with emphasis on user fairness. In contrast to the conventional scheduling algorithms that either equally divides the transmission time slots among users or maximizing some ratios without practical physical interpretations, we propose to use the *5%-tile user data rate* (5TUDR) as the metric to evaluate user fairness. Since it is difficult to directly optimize 5TUDR, we first cast the problem into the stochastic game framework and subsequently propose a *multi-agent reinforcement learning* (MARL)-based algorithm to perform distributed optimization on the resource block group (RBG) allocation. Furthermore, each MARL agent is designed to take information measured by network counters from multiple network layers (e.g. Channel Quality Indicator, Buffer size) as the input *states* while the RBG allocation as *action* with a carefully designed *reward* function developed to maximize 5TUDR. Extensive simulation is performed to show that the proposed MARL-based scheduler can achieve fair scheduling while maintaining good average network throughput as compared to conventional schedulers.**

I. Introduction

Future wireless networks have been envisaged to provide high-quality data services simultaneously to a massive number of devices. To accomplish this demanding goal, intensive research has been devoted to investigating how to allocate limited network resources to multiple users in an effective and yet, fair manner. In particular, user scheduling in the downlink transmission that concerns which active users should be selected and allocated network resources for transmission at the current time slot has drawn much research attention. By equally dividing the transmission time to each user, the Round Robin scheduling (RRS) can achieve the time-fairness at the cost of the network throughput. In contrast, the opportunistic scheduling (OPS) was designed to allocate network resources to the advantageous users of the best instantaneous channel conditions [1]. However, OPS attains the highest network throughput by scarifying those users of poor channel conditions, which incurs unfair resources allocation among users. To balance user fairness and network throughput, the proportional fairness scheduling (PFS) algorithm was proposed in the seminal work [2] by assigning scheduling priorities to users of the largest ratios between their instantaneous feasible data rates and achieved average data rates. Recently, [3] proposed a parameterized PFS called Generalized PFS (GPFS) by generalizing the ratio defined in PFS with different weights. However, both PFS and GPFS only take advantages of information from one single layer, i.e. the physical (PHY) layer. Since the network throughput is governed by protocols across multiple network layers, more comprehensive information from different network layers is required to better characterize the dynamics of the network throughput.

To cope with this problem, a machine learning (ML) approach has recently been proposed for user scheduling [4]. In sharp contrast to the conventional model-based approach, the data-driven ML approach exploits the massive data retrieved from the network without explicitly deriving the mathematical optimization model. In [4], a reinforcement learning (RL)-based algorithm was proposed to intelligently select the best scheduling algorithm from a set of pre-defined algorithms in each transmission time interval (TTI) based on the network conditions. Despite its good performance, the RL-based scheduler proposed in [4] suffers from large computational complexity. To circumvent this problem, [5] proposed an improved RL-based scheduler by taking into account the estimated instantaneous data rate, the averaged data rate and other metrics in constructing its input state space.

However, all the aforementioned scheduling algorithms, regardless of their model-based or data-driven nature, were established upon the assumption of full-buffer traffic, i.e. all users have infinite amount of data for downlink transmission. However, all traffic in practical networks is bursty, i.e. each user's requested data volume is finite. If bursty traffic is considered, the concept of time-fairness becomes rather non-trivial. Since it takes less transmission time slots for users of a smaller amount of bursty data to finish their transmissions in general, it is reasonable to argue that equally dividing transmission time slots among all users is unfair for users requesting for more data. As a result, the concept of time-fairness becomes very subjective for the bursty traffic case. Some pioneering works on scheduling bursty traffic were reported in the literature. In [6], two novelty concepts, namely the average user perceived throughput (UPT) and the user perceived throughput-cut (UPT-cut), were proposed to measure user fairness before a percentage proportional fair scheduling (PPFS) algorithm was devised to allocate more transmission time slots to users of a larger amount of bursty data. Furthermore, PPFS improves the average UPT by assigning higher priorities to users of less remaining transmission data. How-

This work was supported, in part, by the National Key Research and Development Program of China under grant No. 2020YFB1807700 and Guangdong Research Project under grant No. 2017ZT07X152.

† Corresponding author, email: SimonPun@cuhk.edu.cn.

ever, [6] developed its scheduling algorithms by optimizing ratio-based metrics, following the traditional PFS design. Since such ratios do not have practical physical interpretations, these ratio-based algorithms cannot provide any guarantee on the actual data rate achieved by each user.

Inspired by the discussions above, we argue that rate-fairness is a more appropriate metric in designing scheduling algorithms for bursty traffic. In particular, we consider the 5%-tile user data rate (5TUDR), i.e., the 5% worst user data rate, to evaluate the user fairness. Unlike the time-fairness, the 5TUDR metric caters to inferior users without ignoring the overall network throughput. As a result, it is feasible to optimize the 5TUDR metric while implicitly maintaining a satisfactory average user data rate (AUDR). Indeed, the 5TUDR metric has been commonly employed to evaluate network performance by most wireless network operators in practice [1]. However, it is challenging to use 5TUDR in practical network design due to the following reasons. First, since AUDR is governed by factors across multiple network layers, it is technically infeasible to develop a mathematical model to optimize AUDR or 5TUDR. Furthermore, 5TUDR is a long-term performance metric over multiple TTIs after a sequence of resource allocation decisions. Thus, 5TUDR cannot be directly optimized by any single-step scheduling algorithms such as PFS. In particular, if we consider allocating multiple RBGs to multiple users, the optimization space grows exponentially with time, which makes the design problem analytically intractable.

To overcome these challenges, we propose to first cast the RBG allocation task as a cooperative game in which multiple RL agents collaboratively optimize a common objective function using the same reward function. After that, we devise a computationally efficient Multi-Agent Reinforcement Learning (MARL)-based scheduling algorithm to learn a distributed policy by decomposing the action space into multiple smaller action spaces. The main contributions of this paper are summarized as follows:

- We propose to model the RBG allocation process as a stochastic game by taking into account information from multiple network layers for both full-buffer and bursty downlink traffic. To maximize the user fairness without sacrificing the network throughput, we propose a novel reward function specifically designed to optimize 5TUDR;
- Based on the proposed stochastic game, we then develop a MARL-based algorithm to optimize the scheduling policy that achieves good 5TUDR performance while maintaining a considerable AUDR.

II. System Model and Problem Formulation

A. System Model

We consider a wireless network in which a base station (BS) schedules K RBGs to multiple active user equipments (UEs) in the downlink operating in the Frequency Division Duplexing (FDD) mode. The UEs arrive at random. Upon arrival, each UE requests a finite amount of traffic data from the BS. The BS divides its frequency resources into RBGs with each RBG being allocated to at most one UE in each TTI. Furthermore, if a user is scheduled, its requested data will be transferred to a Hybrid Automatic Repeat reQuest and Retransmission (HARQ) buffer. For each transmitted package, the ACK/NACK message is fed back from the targeted UE at a fixed time interval. Upon receiving a NACK message, the BS will re-transmit the corresponding data package. A UE departs from the network immediately after all its requested data is successfully received, which effectively emulates the bursty traffic mode. In our work, some practical network mechanisms such as the Out Loop Link Adaptation (OLLA) are included in our model.

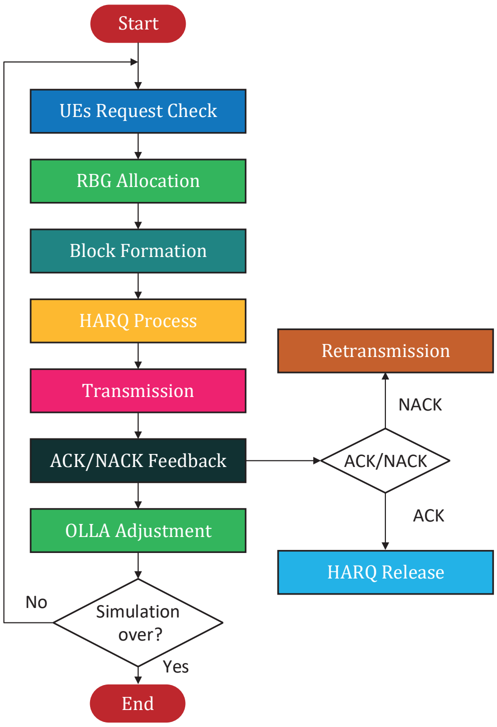

Fig. 1. Network mechanisms

B. Problem Formulation

We begin with the definitions of two key metrics employed in this work, namely AUDR and 5TUDR.

1) Average User Data Rate (AUDR): Without loss of generality, we focus on the first T TTIs. We denote by t_a^n and t_d^n the TTI indices when the n-th user enters and exits the network, respectively, with $1 \leq t_a^n \leq t_d^n \leq T$. Furthermore, we denote by N_t the number of users who have arrived by the t-th TTI with $t \in [1, T]$. It should be emphasized that some of these N_t users may have left the network by the t-th TTI if they finish their transmissions earlier than the t-th TTI. Thus, we can express the corresponding AUDR over $[1, T]$ as:

$$D(t) = \frac{1}{N_t} \sum_{n=1}^{N_t} \psi^n(t), \tag{1}$$

where $\psi^n(t)$ is the user data rate (UDR) of the n-th user over $[t_a^n, t]$ and takes the following form:

$$\psi^n(t) = \frac{\sum_{i=t_a^n}^{t} T^n[i] I^n[i]}{\min(t, t_d^n) - t_a^n}. \tag{2}$$

where $\psi^n(t) = 0$ if $t = t_a^n$. In Eq. (2), $T^n[i]$ is the packet size designated for the n-th user during the i-th TTI. Furthermore, $I^n[i]$ is the indicator function for the ACK/NACK feedback with "1" indicating a successful transmission while "0" a transmission failure. Note that the numerator in Eq. (2) stands for the total amount of successfully transmitted data while the denominator the time duration of the n-th user spent in the system.

2) 5%-Tile User Data Rate (5TUDR): Next, we denote by ϕ_t the 5%-tile user data rate and have:

$$\phi_t = P_{5\%}(\boldsymbol{\Psi}(t)), \tag{3}$$

where $P_{5\%}(\cdot)$ is the operator to find the 5%-tile element in the enclosed set $\boldsymbol{\Psi}(t) = \{\psi^1(t), \ldots, \psi^{N_t}(t)\}$.

Equipped with the two definitions above, we are ready to formulate our problem. Given a network described above, our goal is to establish an optimal scheduling policy denoted by $\boldsymbol{\pi}$ that maximizes the rate-fairness metric 5TUDR over the time interval $[1, T]$. Specifically, the optimization problem can be written as

$$\begin{aligned} \underset{\boldsymbol{\pi}}{\text{argmax}} \quad & \phi_T(\boldsymbol{\pi}) \\ \text{s.t.} \quad & C_1 : D(T) \geq \kappa, \\ & C_2 : P(N_t - N_{t-1} = n) = f(n) \quad \forall t \in [1, T], \end{aligned} \tag{OP1}$$

where κ in Constraint C_1 is the minimum required AUDR while C_2 defines that the probability of having n new arriving users follows some distribution $f(n)$. For example, when the Poisson distribution with an average arrival rate of λ is considered, $f(n) = \frac{\lambda^n}{n!} e^{-\lambda}$.

III. MARL for RBG Allocation

The RBG allocation problem can be considered as a *cooperative game*, which is the generalization of the *Markov decision process* (MDP) with multiple agents. A cooperative game can be defined as a tuple $\mathcal{E} = \langle \mathcal{K}, \mathcal{S}, \mathcal{U}, \mathcal{T}, r, \mathcal{O}, Z, \gamma \rangle$ [7], where $\mathcal{K}$ is the set of agents, $\mathbf{s} \in \mathcal{S}$ describes the global state of the environment, and $\mathcal{U}$ is the local action space. At each time step, each agent $k \in \mathcal{K}$ first draws its individual observation $\mathbf{o}_k \in \mathcal{O}$ using the observation function $Z(\mathbf{s}, k) : \mathcal{S} \times \mathcal{K} \to \mathcal{O}$ before selecting an action $u_k \in \mathcal{U}$. This forms a joint-action $\mathbf{u} \in \boldsymbol{\mathcal{U}} \equiv \mathcal{U}^{|\mathcal{K}|}$ that induces a transition in the environment following the transition probability $\mathcal{T}(\mathbf{s}'|\mathbf{s}, \mathbf{u}) : \mathcal{S} \times \boldsymbol{\mathcal{U}} \times \mathcal{S} \to [0, 1]$. Finally, $r(\mathbf{s}, \mathbf{u}) : \mathcal{S} \times \boldsymbol{\mathcal{U}} \to \mathbb{R}$ is the reward function that evaluates the team reward for a given state-action pair and $\gamma \in (0, 1]$ is a discount factor.

Fig. 2 illustrates the overview of this cooperative game for the task of RBG allocation. In this game, we arrange K agents denoted by $\mathcal{K} = \{1, \ldots, K\}$ to allocate the K RBGs separately. The k-th agent for $k \in \mathcal{K}$ learns an independent scheduling policy π_k to decide the allocation of the k-th RBG based on its partial observation. The joint policy function is $\boldsymbol{\pi} = \prod_{k=1}^{K} \pi_k$, and $\mathbf{u} = (u_1, \ldots, u_K)$ is the joint action of all K agents. In the following subsections, the basic elements: observation, action, and reward function are well defined, along the learning algorithm is also detailed.

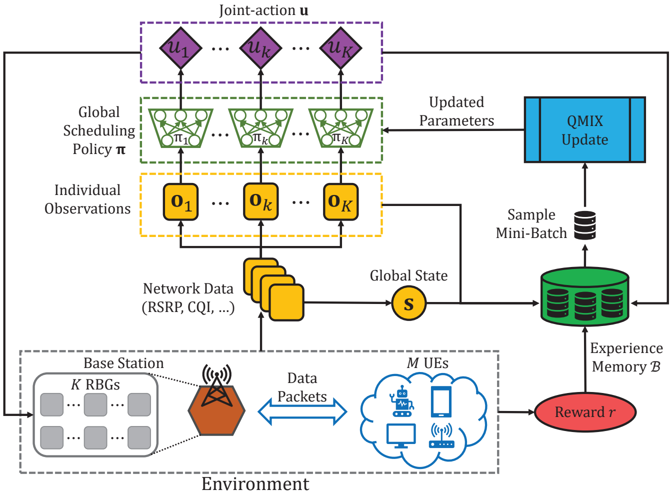

Fig. 2. The overview of the cooperative game of the RBG allocation.

A. Cross-layer Observation and Adaptive Action

For a traditional model-based scheduler, it can only utilize information from one single network layer due to the prohibitively high complexity incurred in modeling information from multiple layers. In this paper, we propose to better characterize the network status by exploiting information from multiple network layers.

TABLE I
Network Data of Observations

Counter Name	Layer	Dimension
RSRP	PHY	1
Average of RB CQI	PHY	1
Buffer size	MAC	1
Scheduled frequency	MAC	1
OLLA offset	MAC	1
Historical user data rate (HUDR)	MAC	1

Table I shows the network counters selected in our design as well as the layer that each counter belongs to. For instance, Reference Signal Receiving Power (RSRP) is selected to represent the longer-term wireless signal strength while Channel Quality Indicator (CQI) is employed to reflect the instantaneous channel quality corresponding to the channel's signal-to-noise ratio (SNR). Note that the CQI value for the same user varies in different RBGs due to the frequency-selective channels commonly encountered by the broadband wireless communication systems. Furthermore, we take into account the buffer size that stands for the amount of remaining data to be transferred to an active user as well as each user's scheduled frequency that keeps track of how often each user has been scheduled for transmission. Finally, the historical user data rate (HUDR) and the OLLA offset are also included in our observations, where the HUDR is the UDR up to the last TTI. As indicated in Table I, all these network parameters are taken from either the physical (PHY) or the multiple access (MAC) layers. It should be emphasized that more counters can be included in our observation in a straightforward manner at the cost of higher computational complexity.

At each TTI, each agent forms its observation including the counters listed in Table I. We denote by C and M_t the total number of network counters under consideration and the number of active users at the current TTI, respectively. We can then construct an $M_t \times C$ feature matrix $\mathbf{O}_k$ for each $k \in \mathcal{K}$ based on their individual observations as follows:

$$\mathbf{O}_k = \begin{pmatrix} \tilde{o}_k^{1,1} & \cdots & \tilde{o}_k^{1,C} \\ \vdots & \ddots & \vdots \\ \tilde{o}_k^{M_t,1} & \cdots & \tilde{o}_k^{M_t,C} \end{pmatrix}, \tag{4}$$

where $\tilde{o}_k^{m,c}$ stands for the c-th counter value over the k-th RBG observed by the m-th user with $m = 1, \dots, M_t$ and $c = 1, \dots, C$. For instance, assuming that CQI is labeled as the first counter, then $\tilde{o}_k^{m,1}$ represents the CQI of the m-th user over the k-th RBG.

To fully exploit the features of the cross-layer information, we propose to use the deep neural network (DNN) to represent the individual scheduling policy, which has fixed input dimension and output dimension. However, for *bursty traffic*, the dimension of the feature matrix $\mathbf{O}_k$ varies with M_t over time, which makes it incompatible with the DNN. To circumvent this problem, we propose to apply the following operation on $\mathbf{O}_k$:

$$\mathbf{o}_k = \text{Vec}\left(\mathbf{O}_k^T \mathbf{O}_k\right), \tag{5}$$

where the operator $\text{Vec}(\cdot)$ converts the enclosed matrix into a single vector row by row. Note that the resulting vector $\mathbf{o}_k$ has a fixed length of C^2. Furthermore, we define the global state $\mathbf{s}$ as the collection of all local observations $\{\mathbf{o}_k\}_{k \in \mathcal{K}}$.

Capitalizing on $\mathbf{o}_k$ of a fixed length, the DNN generates preference values to all active users before assigning its RBG to the best user. However, the number of active users M_t varies with time. In order to have a fixed size of the DNN output, we propose to pre-define a maximum allowable number of active users denoted by $M_{\max}$. Therefore at each decision step, $M_{\max}$ preference values are generated, but only the former M_t values are used to guide the RBG allocation.

B. Reward Function

To harvest the benefit of being able to accurately characterize the network status stated above, we will next devise a well-defined reward function to evaluate the performance of each agent. Since the design goal is to maximize 5TUDR while maintaining a considerable AUDR, the desired reward function should be able to evaluate the contribution of each scheduling decision towards the final 5TUDR and AUDR. A trivial 5TUDR-maximizing reward function can be designed to maximize the increment in 5TUDR between two consecutive TTIs. Unfortunately, the performance of such reward functions has been found very poor with bursty traffic. Fig. 3 shows the increment of 5TUDR as a function of time in TTI. Inspection of Fig. 3 reveals that the increment of 5TUDR is always 0 with occasional sharp increases and decreases. Careful investigation on those sharp spikes indicates that those sharp decreases were caused by the arrival of new users whose ψ_t is 0. In contrast, once this new user was scheduled, its user data rate suddenly became non-zero, which caused a sharp increase in the increment of 5TUDR. Therefore, using the increment of 5TUDR as the reward function is not appropriate for bursty traffic as the increment is a random event due to the arrival or departure of users.

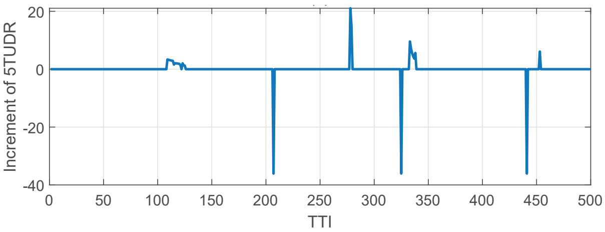

Fig. 3. An example of the increment (decrement when negative) of 5TUDR between two consecutive TTIs over time.

Finally, we propose the following reward function by exploiting the changes of the *sum* of UDR over two consecutive TTI's, respectively.

$$r(t) = h\left(\sum_{n=1}^{M_t} \psi^n(t) - \sum_{n=1}^{M_{t-1}} \psi^n(t-1)\right) - \exp\{-\mathcal{G}\left(\Delta\boldsymbol{\Psi}(t)\right)\}, \tag{6}$$

where $h(\cdot)$ is the sigmoid function and $\Delta\boldsymbol{\Psi}(t)$ stands for the UDR change of each active user given by

$$\Delta\boldsymbol{\Psi}(t) = \{\Delta\psi^1(t), \dots, \Delta\psi^{M_t}(t)\}, \tag{7}$$

with $\Delta\psi^n(t) = \psi^n(t) - \psi^n(t-1)$ for $n = 1, 2, \dots, M_t$. In addition, $\mathcal{G}(\cdot) \in [0, 1]$ is the G's fairness index that measures the level of variations of the elements of the enclosed set.

It is worth mentioning the following features of the proposed reward function in Eq. (6). First, the reward function is conditioned upon the sum of all user data rates, which makes the reward function insensitive to the time-varying number of total arrived users. In particular, when new users of zero UDR enter the system, the reward function is not affected. Second, since the reward function effectively maximizes the increment in the user data sum-rate, the resulting scheduling policy drives all agents to effectively allocate RBGs. Finally, the G's fairness index is introduced as the regularization term to discourage agents to be in favor of a few users. As a result, more users will be given transmission opportunities, which effectively improves 5TUDR.

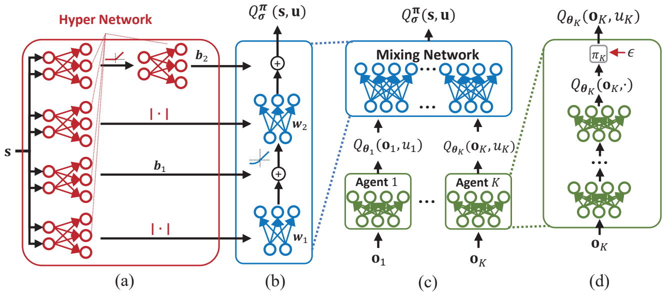
Hyper Network
$Q^{\pi}_{\sigma}(\mathbf{s},\mathbf{u})$
$Q^{\pi}_{\sigma}(\mathbf{s},\mathbf{u})$
$Q_{\theta_K}(\mathbf{o}_K,u_K)$
$\mathbf{s}$
b_2
$|\cdot|$
b_1
$|\cdot|$
w_2
w_1
Mixing Network
$Q_{\theta_1}(\mathbf{o}_1,u_1)$
$Q_{\theta_K}(\mathbf{o}_K,u_K)$
Agent 1
Agent K
$\mathbf{o}_1$
$\mathbf{o}_K$
π_K
ϵ
$Q_{\theta_K}(\mathbf{o}_K,\cdot)$
$\mathbf{o}_K$
(a)
(b)
(c)
(d)

Fig. 4. (a) The hyper-network that produces the weights and biases for mixing network layers shown in blue. (b) Mixing network structure. (c) The overall QMIX architecture. (d) Agent network structure.

C. Learning Algorithm

Finally, we introduce a QMIX algorithm (Fig. 4) to learn the scheduling policy. QMIX is a model-free, value-based, off-policy, decentralized execution but centralized training algorithm designed for cooperative tasks, and more details about the QMIX can be found in [8]. The QMIX-based algorithm updates the scheduling policy as summarized in Algorithm 1.

Algorithm 1 The QMIX algorithm for User Scheduling

```
Initialize $K$ agent networks $\{Q_{\boldsymbol{\theta}_1},\ldots,Q_{\boldsymbol{\theta}_K}\}$ with DNN parameters $\boldsymbol{\theta}=\{\boldsymbol{\theta}_1,\ldots,\boldsymbol{\theta}_K\}$;
Initialize the hyper-network $\mathcal{H}_{\boldsymbol{\delta}}$ and the mixing network $\mathcal{M}_{\boldsymbol{\sigma}}$ with parameters $\boldsymbol{\delta}$ and $\boldsymbol{\sigma}$;
Set a replay buffer $\mathcal{B}$, a learning rate $\tau$, a discount factor $\gamma$ and an exploration rate $\epsilon$;
**for** $t=1,\ldots,T$ **do**
    Each agent $k$ makes its local action $u_k(t)$ based on its local observation $\mathbf{o}_k(t)$ using the $\epsilon$-greedy policy before observing a new local state $\mathbf{o}_k(t+1)$;
    Execute the joint-action $\mathbf{u}(t)$ and evaluate the team reward $r(t)$ before collecting a new global state $\mathbf{s}(t+1)$;
    Store the following transition in $\mathcal{B}$ for $k\in\{1,\ldots,K\}$:
        $\{\mathbf{s}(t),r(t),\mathbf{s}(t+1)\},\{\mathbf{o}_k(t),\mathbf{o}_k(t+1),u_k(t)\}$;
    Sample a random mini-batch of $b$ transitions from $\mathcal{B}$:
        $\{\mathbf{s}(i),r(i),\mathbf{s}(i+1)\},\{\mathbf{o}_k(i),\mathbf{o}_k(i+1),u_k(i)\},\forall k$;
    Derive $K$ individual $Q$ values:
        $\mathbf{Q}=\{Q_{\boldsymbol{\theta}_1}(\mathbf{o}_1(i),u_1(i)),\ldots,Q_{\boldsymbol{\theta}_K}(\mathbf{o}_K(i),u_K(i))\}$,
        $\mathbf{Q}'=\{Q_{\boldsymbol{\theta}_1}(\mathbf{o}_1(i+1),u_1(i+1)),\ldots,$
            $Q_{\boldsymbol{\theta}_K}(\mathbf{o}_K(i+1),u_K(i+1))\}$;
    Compute weights for mixing network, then get $Q^{\pi}_{\boldsymbol{\sigma}}(\mathbf{s}(i),\mathbf{u}(i))$ and $Q^{\pi}_{\boldsymbol{\sigma}}(\mathbf{s}(i+1),\mathbf{u}(i+1))$:
        $\boldsymbol{\sigma}(i)\leftarrow\mathcal{H}_{\boldsymbol{\delta}}(\mathbf{s}(i))$,
        $\boldsymbol{\sigma}(i+1)\leftarrow\mathcal{H}_{\boldsymbol{\delta}}(\mathbf{s}(i+1))$,
        $Q^{\pi}_{\boldsymbol{\sigma}}(\mathbf{s}(i),\mathbf{u}(i))\leftarrow\mathcal{M}_{\boldsymbol{\sigma}(i)}(\mathbf{Q})$,
        $Q^{\pi}_{\boldsymbol{\sigma}}(\mathbf{s}(i+1),\mathbf{u}(i+1))\leftarrow\mathcal{M}_{\boldsymbol{\sigma}(i+1)}(\mathbf{Q}')$;
    Set $y(i)=r(i)+\gamma\max_{\mathbf{u}(i+1)}\{Q^{\pi}_{\boldsymbol{\sigma}}(\mathbf{s}(i+1),\mathbf{u}(i+1))\}$;
    Update all the agent networks and hyper-network by minimizing the following loss:
        $L(\boldsymbol{\theta},\boldsymbol{\sigma},\boldsymbol{\delta})=\frac{1}{b}\sum_{i=1}^{b}\left[y(i)-Q^{\pi}_{\boldsymbol{\sigma}}(\mathbf{s}(i),\mathbf{u}(i))\right]^2$.
**end for**
```

TABLE II
BS PARAMETERS USED IN SIMULATION

Parameters	Values
Transmit power for each RB	18 dBm
Number of RB for each RBG	3
Number of RBGs	3
Frequency bandwidth for each RBG	10MHz
Noise power density	-174 dBm/Hz
Minimum MCS	1
Maximum MCS	29
Mximum number of HARQ	8
Feedback period of HARQ	8
Initial RB CQI value	4

IV. Numerical Results

In this section, computer simulation is performed using the network simulator defined in Section II. Table II summarizes the parameters employed in the simulation. For illustration purposes, only three RBGs are deployed in the BS with each RBG composed of three resource blocks (RBs). In particular, each HARQ process can have at most eight re-transmissions with the ACK/NACK message is returned with a delay of seven TTIs.

A. Performance Comparison

We select the GPFS and RRS as the benchmark schemes to compare the AUDR and 5TUDR performance with the proposed MARL-based scheduler. Moreover, we build two different GPFS with different parameters, which are GPFS1 ($\alpha_1 = 0, \alpha_2 = 1$) and GPFS2 ($\alpha_1 = 0.5, \alpha_2 = 1$), respectively. The GPFS1 only focuses on the user fairness, while the GPFS2 partly takes the AUDR into consideration. A larger α_1 encourages GPFS to give higher scheduling priorities to users of better channel conditions, which initially improves both the AUDR and 5TUDR performance by more efficiently utilizing the network resources.

We compute the improvement of the proposed scheduler over the three aforementioned conventional schedulers using 100 experiments. Fig. 5 and Fig. 6 plot the cumulative distribution function (CDF) of the improvements in terms of AUDR and 5TUDR, respectively. A positive x-axis value indicates that the proposed scheduler outperforms the conventional scheduler in comparison. The average AUDR improvement achieved by the proposed MARL-based scheduler was 83.20 Bits/TTI and 147 Bits/TTI as compared to GPFS1 and RRS, respectively while the average AUDR degradation as compared to GPFS2 was 29.84 Bits/TTI. Furthermore, Fig. 6 shows the CDF of the 5TUDR improvement. Clearly, the proposed scheduler significantly outperformed all three conventional schedulers in terms of 5TUDR, which is evidenced by the fact that most x-axis values are positive. Specifically, the proposed scheduler achieved an average 5TUDR improvement of 54.70 Bit/TTI, 39.42 Bits/TTI and 46.96 Bits/TTI as compared with GPFS1, GPFS2 and RRS, respectively. Based on the discussions above, we can see that the proposed MARL-based scheduler is able to achieve substantially better 5TUDR performance (i.e. higher fairness) while maintaining a considerably large AUDR.

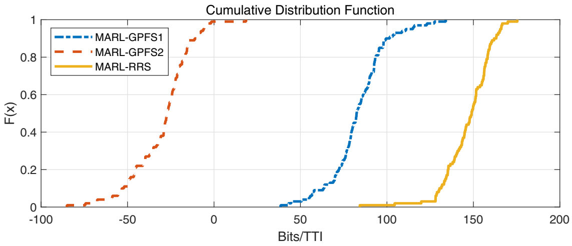

Fig. 5. CDF of the AUDR improvement achieved by the proposed algorithm as compared to PFS and RRS.

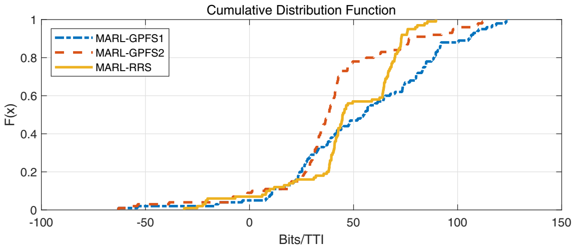

Fig. 6. CDF of the 5TUDR improvement achieved by the proposed algorithm as compared to PFS and RRS.

B. Policy Analysis

Furthermore, we attempt to compare the scheduling preference of the proposed scheduler, GPFS1, GPFS2 and RFF. In the following simulation, we consider five users competing for three RBGs for bursty traffic over 500 TTIs. We analyze the number of RBGs that each scheduler allocates to a single user in the same TTI. Fig. 7 shows the percentages of scheduled users who were allocated with "One", "Two" or "Three" RBGs in the same TTI, recalling that our simulation only considered three RBGs. As indicated in Fig. 7, GPFS1 and RRS allocated all the RBGs to a single user almost throughout the entire simulation. In contrast, the proposed scheduler and GPFS2 chose to distribute their RBGs to different users more evenly. As a result, users are given more opportunities to access RBGs, which led to higher user fairness. In particular, the proposed scheduler never allocated all three RBGs to one single user, which makes our proposed scheduler very distinctive from the three conventional schedulers.

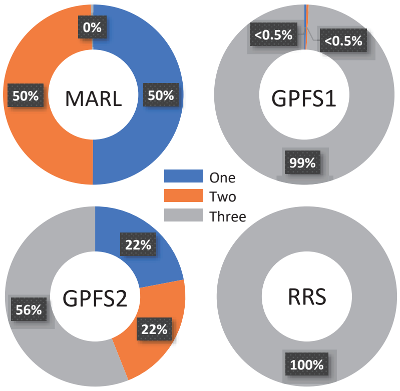

Fig. 7. Percentages of scheduled users who were allocated with "One", "Two" or "Three" RBGs.

V. Conclusion

In this work, we have investigated the problem of RBG scheduling for users of bursty traffic in wireless networks. To provide fair opportunities to all users to access the available RBGs, a fairness-oriented scheduler has been formulated as a stochastic game by incorporating information from multiple network layers such as CQI and buffer size. In particular, a reward function was devised to maximize the 5%-tile user data rate. Furthermore, a MARL-based learning approach has been proposed to learn the optimal scheduling policy by dividing the large state-space into multiple sub-problems each of which is handled by one agent. Extensive computer simulation has been performed to compare the performance of the proposed scheduler against the conventional GPFS, OPS and RRS. Simulation results have confirmed that the proposed scheduler can achieve impressive 5TUDR performance while maintaining good AUDR performance.

References

[1] X. Liu, E. Chong, and N. Shroff, "Opportunistic transmission scheduling with resource-sharing constraints in wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 19, no. 10, pp. 2053–2064, 2001.

[2] D. Tse, "Multiuser diversity in wireless networks," in *Wireless Communications Seminar, Standford University*, 2001.

[3] I.-S. Comșa, S. Zhang, M. Aydin, P. Kuonen, R. Trestian, and G. Ghinea, "A comparison of reinforcement learning algorithms in fairness-oriented ofdma schedulers," *Information*, vol. 10, no. 10, p. 315, 2019.

[4] I.-S. Comşa, S. Zhang, M. E. Aydin, P. Kuonen, Y. Lu, R. Trestian, and G. Ghinea, "Towards 5g: A reinforcement learning-based scheduling solution for data traffic management," *IEEE Transactions on Network and Service Management*, vol. 15, no. 4, pp. 1661–1675, 2018.

[5] C. Xu, J. Wang, T. Yu, C. Kong, Y. Huangfu, R. Li, Y. Ge, and J. Wang, "Buffer-aware wireless scheduling based on deep reinforcement learning," in *2020 IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 1–6, IEEE, 2020.

[6] Y. Li, X. Fang, and S. Liang, "A novel scheduling algorithm to improve average user perceived throughput for lte systems," *IEEE Access*, vol. 7, pp. 133549–133558, 2019.

[7] S. Omidshafiei, J. Pazis, C. Amato, J. P. How, and J. Vian, "Deep decentralized multi-task multi-agent reinforcement learning under partial observability," *arXiv preprint arXiv:1703.06182*, 2017.

[8] T. Rashid, M. Samvelyan, C. S. De Witt, G. Farquhar, J. Foerster, and S. Whiteson, "QMIX: Monotonic value function factorisation for deep multi-agent reinforcement learning," *arXiv preprint arXiv:1803.11485*, 2018.