Achieving the Highest Average User Data Rate in Wireless Network using Reinforcement Learning

Qi Cao[1*], Mingqi Yuan[2*], Siliang Zeng[1], Man-On Pun[1†] and Yi Chen[1]
[1]The Chinese University of Hong Kong, Shenzhen
Guangdong, China, 518172
[2]Mingqi's affiliation

***Abstract*—The frequency resources allocation in wireless communication is aimed at improving some predefined network-level key performance indicator (KPI) such as the network capacity, the average user data rate, and the 5%-tile user data rate. This allocation is essentially realized by matching candidate users with resource block group (RBG) each occupying a transmission time interval and a fixed bandwidth, and this process can be referred as user scheduling. When users are with full buffers, it is known that *opportunistic* scheduling is optimal in terms of achieving the highest average user date rate (AUDR). However, in the bursty buffer case, the optimal scheduling has not been revealed, especially when it is influenced by mechanisms from both physical (PHY) layer and media access control (MAC) layer. In this context, we leverage reinforcement learning (RL) to train a convolutional neural network (CNN), namely RBGNet, which is able to flexibly take dynamic number of users' states as input and produce feasible scheduling solutions leading to high AUDR. With a self-built simulator involving essential mechanisms in PHY and MAC layer, the superiority of our proposed RBGNet is verified, compared to *opportunistic* and *proportional fairness (PF)* scheduling. As such, this work provides a remarkable scheduling algorithm in achieving the highest AUDR in bursty buffer cases.**

***Index Terms*—RBG allocation, Reinforcement learning, CNN, Highest average user data rate**

I. Introduction

User scheduling is a classical problem in wireless communications, which can be traced back decades ago [1]. That being said, the optimal strategy has never been found. It can be explained from two different angles which are: first the optimality is difficult to determine when it comes to the comprehensive performance of throughput, fairness and delay; and second even if the key performance indicator (KPI) is well defined, the optimality only exists in theoretical models but never a practical system involving numerous transmission mechanisms. On the other hand, the meaning of so-called user scheduling can be extended to a wide range, which is not limited to power allocation, user pairing, transmission time interval (TTI) and resource block group (RBG) arrangement.

Among the early works, proportional fair (PF) scheduling is an essential one as it balances between throughput and fairness [2]. The PF scheduling is proved to maximize the sum of logarithmic average user rates $\sum \log R_i$, being R_i the individual user average data rate [3], and its multi-carrier version is given in [4], forming the basic RBG allocation algorithm for today's wireless network. More recently, the PF scheduling is widely adopted or further investigated in various scenarios, for instance, the non-orthogonal multiple access system [5], [6]. And it triggers gradient scheduling algorithm studies aimed to maximize a utility function named α-fairness, which generalizes the objective of the scheduling [7].

In this paper, we focus on the RBG allocation but as far as we know, the optimal strategy achieving the highest average user data rate is never revealed in bursty buffer cases. Even so, the optimal for full buffer cases is known as opportunist algorithm, which simply greedily allocates RBGs to the users with highest expected data rate in each TTI [1]. It is worth mentioning that mathematical modeling is almost impossible to solve the problem. The technical challenges come from that: first, a wireless communication network is highly complex with many components and mechanisms, rendering the whole system analytically intractable; second, it is difficult to accurately describe channel transfer function, inter-cell interference and user distribution and movement etc; finally, network events are mostly stochastic such as user arrival, traffic load and channel variations.

On the other hand over the recent years, artificial intelligence (AI), specifically machine learning, has entered many industrial fields, be it traditional or emerging, and plays the role of a mighty competitor challenging existing rules and stereotypes. The exploding demand of data over the air loads increasingly more pressure on current mobile Internet, while model-driven solutions are leading us to the bottleneck. In such a context, the combination of machine learning and wireless communications is an essential and promising next step, where there are already studies trying the methodology in power allocation [8], [9] and modulation and coding rate selection [10].

In this paper, we attempt to leverage deep reinforcement learning to allocate network-level performance using both MAC and PHY layer information. Our main contributions are as follows:

- In a sophisticated wireless network simulator, we collect the feasible features and define the framework of our proposed deep reinforcement algorithm. Even though the long-term objective is achieving the highest average user data rate, it is not clear what should the reward be according to an action. As such, we developed a deviation based reward function which is shown to be effective in achieving our objective.
- The number of users attached on a base station (BS) is always dynamic. To facilitate the practical implementation of our proposed RBGNet. We construct a flexible CNN

*Authors contribute equally. †Corresponding author, email: SimonPun@cuhk.edu.cn. This work was supported by

configuration, which could cope with dynamic number of users and arrange them to fixed number of RBGs. Besides, a training algorithm accordingly is developed to contribute the convergence and efficacy of the RBGNet.
- With a large number of various discrete/continuous counters and complicated operating mechanisms such as Out Loop Link Adaptation (OLLA) and Hybrid Automatic Repeat reQuest (HARQ) and retransmission in wireless communication network, we demonstrate that using the proposed RBGNet is possible to outperform traditional PF or opportunist scheduling in terms of average user data rate.

In what follows, we introduce the task specifications and dataset establishment in Sec. II and the designed DNN configuration is detailed in Sec. III. The proposed training algorithm is elaborated in Sec. IV. Finally Sec. V presents the simulation results before conclusion is given in Sec. VI.

II. Simulator Basics

This section introduces the details of the built simulator which follows the protocol of LTE [11]. Specifically, K RBGs share a frequency bandwidth B to serve Poisson-arriving downlink users with expected arrival rate λ. The base station is equipped with two transmit/receive antennas respectively while UEs are equipped with one transmit antenna and two receive antennas. The transmit power is beyond the scope of this paper and so assumed to be the same to all users. The LTE protocol allows a user to suggest an appropriate MCS via feeding back a channel quality indicator (CQI), which eventually stabilizes the block error rate (BLER) of the user to a certain level. The simulator randomly generates an ACK/NACK message by a probability to every transmission, indicating its successful/unsuccessful delivery. The probability is related to the MCS chosen and the actual signal-to-noise (SNR) ratio and an offset per user. In what follows, some critical mechanisms are elaborated, from which it can be seen that building mathematical models to derive a user scheduling in any sense of optimality is intractable.

1) Out Loop Link Adaptation: On the user side, a CQI level representing an SNR interval is periodically computed and reported. Thus the received instantaneous SNR directly influences the selection of MCS. However it should be noted that there is an offset reflecting the detection sensitivity which is a natural property per user and unknown to the BS. Thus, to compensate the discrepancy between the chosen MCS and the optimal MCS for different users. An OLLA process occurred on BS to carry out to offset a CQI value q is used and given by

$$\bar{q} = [q + \alpha], \tag{1}$$

being $[\cdot]$, the rounding operation and $\bar{q}$ the offset CQI readily for MCS selection. Note that α is the adjustment coefficient updated every time an ACK/NACK message is captured, i.e.

$$\alpha = \begin{cases} \alpha + s_A, & \mathfrak{A} = 1, \\ \alpha + s_N, & \mathfrak{A} = 0, \end{cases} \tag{2}$$

being $\mathfrak{A} = 1$ the ACK message and $\mathfrak{A} = 0$ the NACK message being 1 for ACK and 0 for NACK. The update rate s_A being positive and s_N being negative can be customized, and it is trivial to show that the final BLER converges to $-s_N/s_A$.

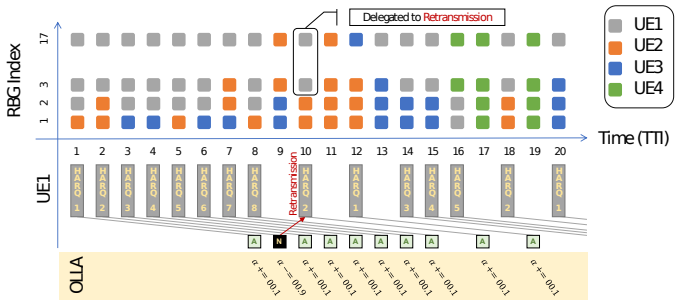

Fig. 1. Network Mechanisms

2) Transmission Block Formation: Due to the relationship among CQI, MCS and spectral efficiency (SE) is fixed. Assume $\mathcal{M}(\bar{q})$ denotes the mapping from the offset CQI $\bar{q}$ to its SE, then, the estimated data rate of UE n in RBG k can be obtained by

$$R_{n,k} = \sum_{\ell \in \mathcal{G}(k)} B\mathcal{M}(\bar{q}_{n,\ell}), \tag{3}$$

where $\mathcal{G}(k)$ is the set of all RBs in RBG k, and ℓ is the index of an RB, and $\bar{q}_{n,\ell}$ is the offset CQI level of UE n in RB ℓ. The BS adopts a default MCS when the CQI is not given, which especially happens at the beginning of a transmission.

When multiple RBGs are allocated to the a UE, they are utilized to convey one data packet (Transport Block, TB) with the same MCS. That is to say, even though the estimated data rate $R_{n,k}$ is adopted in the user scheduling, but the size of the TB loading on the RBG might differ. Suppose $\{\bar{q}_{n,\ell}\}$ with size L is the set of all RBs allocated to UE n, then the actual TB size is written as

$$T_n = LB\mathcal{M}\left(\left\lfloor \frac{1}{L}\sum_{\ell} \bar{q}_{n,\ell} \right\rfloor\right), \tag{4}$$

where $\lfloor\cdot\rfloor$ is the floor function. However if a UE fails to acquire any RB, T_n will be equal to zero at the current TTI.

3) Hybrid Automatic Repeat reQuest and Retransmission: When the TB is set for a UE, the corresponding data will be loaded to the HARQ buffer and held there until being dropped as an outcome of either successful transmission or expiration. The convenience for the mechanism is that whenever a retransmission is required, the data can be directly fetched in HARQ buffer. The BS prepares eight HARQ processes for each UE. The BS will see an ACK/NACK message from the corresponding user in seven TTIs. In the case of ACK, the HARQ process terminates, while in the case of NACK, a retransmission is triggered. The RBGs initially delegated to the first transmission will conduct the retransmission, thus being unavailable for user scheduling. The MCS selection and TB size remains the same for the retransmission and the HARQ process will expire at five times of consecutively failure, which literally causes the so-called packet loss.

III. Objective

Assume a user n arrives at TTI $t_{n,a}$, and the service is completed at TTI $t_{n,t}$. We have the user data rate equal to

$$C_n = \frac{S_n}{t_{n,t} - t_{n,a}}. \tag{5}$$

However, as some users may not finish transmitting all data in the buffer at the moment of compute the data rate, thus we could alternatively use the following temporal user data rate written as

$$\bar{C}_n[t] = \frac{\sum_{i=t_{n,a}}^{t} T_n[i]\mathfrak{A}_n[i]}{\min(t, t_{n,t}) - t_{n,a}}, \quad t > t_{n,a}. \tag{6}$$

Note that $\bar{C}_n[t]$ has no meaning when $t \leq t_{n,a}$. Our objective in this paper is to find an effective user scheduling maximizing the expectation of all arrival users, i.e.

$$\max \lim_{t\to\infty} \mathbb{E}\{\bar{C}_n[t]\}, \quad \forall n.$$

In the sequel, we sometimes omit the TTI index for brevity.

IV. Framework of Reinforcement Learning

A. Preliminaries

Considering the intrinsic properties of wireless communication, this thesis model this RBG allocation problem as a *Markov decision process*(MDP), which comprises: a *state space* $\mathcal{S}$, an *action space* $\mathcal{A}$, an *initial state distribution* with density $p_1(s_1)$, a stationary *transition dynamics distribution* with conditional density $p(s_{t+1}|s_t, a_t)$ satisfying the Markov property $p(s_{t+1}|s_1, a_1, ..., s_t, a_t) = p(s_{t+1}|s_t, a_t)$, for any trajectory $s_1, a_1, s_2, a_2, ..., s_T, a_T$ in state-action space, and a *reward function*:$\mathcal{S} \times \mathcal{A} \to \mathbb{R}$.

In the following section, the basic elements: state, action, reward signal, etc, of this customized reinforcement learning system for RBG allocation will be elaborated respectively.

B. Elements of RL system

1) State: The performance of a communication system is affected by the status of mutiple layers.However, it's difficult to establish model across different layers in the traditional research.In order to break that restriction, cross-layer information was adopted to characterize the state of environment, and I illustrates the details of those attributes.

TABLE I
Attributes of State

Attr.	Layer	Dim.
RSRP	PHY	1
buffer size	MAC	1
historical throughput	MAC	1
Olla offset	MAC	1
scheduled times	MAC	1
RB CQI	PHY	Num. of RB
average of RB CQI	PHY	1
std. of RB CQI	PHY	1

According to the I, mutiple attributes were carefully selected to compose the state, which are frequently varied in the communication process.And the data shape of the state can be calculated as:

$$ds_{state} = (K, N_{RB} + 7) \tag{7}$$

where the ds is short for data shape,N_{RB} and K are the quantity of RB and users.The construction of the subsequent deep neural network is also related to the specific data shape of the state.

2) Action: In our RL system, action is the scheduling result of RBG, which is generated by the actor network defined in **.Let $\mathbf{O}_{K\times N_{RB}} = (o_1, o_2, ..., o_{N_{RBG}})$ denotes the output of actor network, the action can de calculated as below:

$$\begin{aligned} \hat{o_i} &= \frac{exp(o_i)}{\sum_j^{N_{RB}} exp(o_j)} \\ a_{tti} &= \arg \max_{\hat{o_i}, i=1,2,...,N_{RBG}} \hat{\mathbf{O}} \end{aligned} \tag{8}$$

3) Reward: The reward signal indicates the goal in a RL system, which motivates the agent to explore any possible solutions to maximize the optimization target.This thesis aims to maximize the AUDR, to this end, a $\Delta r_n[t]$ defined in () was taken to configure reward function.From the point of view of a user n, the optimistic data rate regardless of the outcome of ACK/NACK in TTI $t+1$ can be written as

$$\begin{aligned} \hat{C}_n[t] &= \frac{\bar{C}_n[t-1] * (t - 1 - t_{n,a}) + R_n[t]}{t - t_{n,a}} \\ &= \bar{C}_n[t-1]\frac{t-1-t_{n,a}}{t-t_{n,a}} + \frac{R_n[t]}{t-t_{n,a}} \\ &= const. + \Delta r_n[t] \end{aligned} \tag{9}$$

where the first term is irrelevant to any actions but the second term is fully controllable. We define the second term as the *data rate deviation* $\Delta r_n[t] = R_n[t]/(t - t_{n,a})$, Thus we design the reward function as

$$r_t = \log_\eta \left(\sum_{i=1}^{N} \Delta r_i[t] + 1 \right), \tag{10}$$

where the log function and η are introduced, the former ensure that the reward function is strictly incremental, and the latter controls the growth rate of reward rate.It's worth mentioning that when the η is a huge number, the output of reward function varies in a smaller range, such as $[0, 1]$, which will promote the convergence of the RL model.

4) Learining algorithm: The learning algorithm gives the method of exploration and adjustment. A *Deep Deterministic Policy Gradient* algorithm is introduced in this study, which is model-free, off-policy actor-critic algorithm uses deep function approximators that can learn policies in high-dimensional, continuous action spaces [12]. The whole algorithm requires two important roles, namely an actor and a critic. The actor network generates action and the critic network estimates the reward, so let $\mu(s|\mathbf{\Theta}^\mu)$ denote the actor and $Q(s, a|\mathbf{\Theta}^Q)$ the critic respectively. Yet, the application of DDPG for RBG allocation is illustrated in the Algorithm 0.

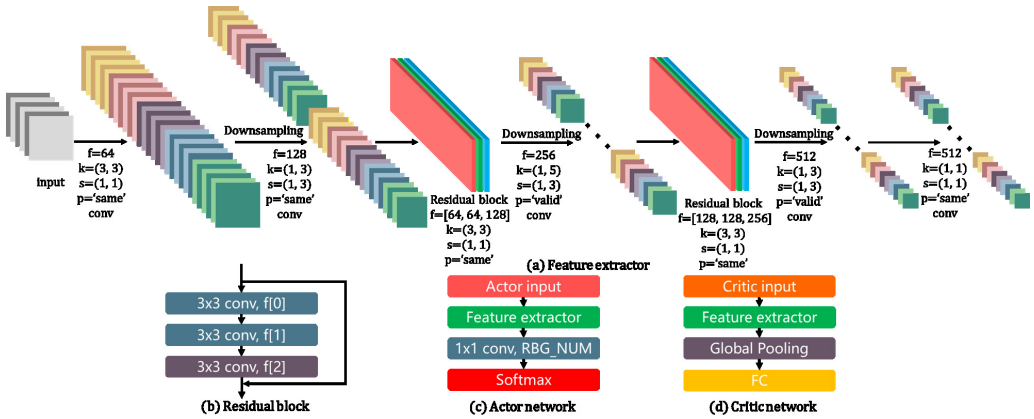

Fig. 2. Architecture of RBGNet.The parameters of downsampling layers were configured skillfully based on ds_{state}

Algorithm 1 DDPG for RBG Allocation

Initialization: Randomly initialize the weights in actor and critic network $\mathbf{\Theta}^Q$ and $\mathbf{\Theta}^\mu$; Set a replay buffer B and learning rate τ.
for Epoch$= 1 : M$ **do**
  Randomly Generate K users.
  for $t = 1 : T$ **do**
    Add users to base station according to Poisson distribution.
    Execute action $a_t = \mu(s_t|\mathbf{\Theta}^\mu)$ and observe new state s_{t+1}.
    Receive reward r_t and store the transitions (s_t, a_t, r_t, s_{t+1}) in B.
    Sample a random minibatch of N transitions (s_i, a_i, r_i, s_{i+1}) from B.
    Set $y_i = r_i + \gamma Q'(s_{i+1}, \mu'(s_{i+1}|\mathbf{\Theta}^{\mu'})|\mathbf{\Theta}^{Q'})$
    Update critic by minimizing the loss function:
$L = \frac{1}{N}\sum_i (y_i - Q(s_i, a_i|\mathbf{\Theta}_Q)^2)$
    Update the actor policy using the sampled gradient:

$$\nabla_{\mathbf{\Theta}^\mu}\mu|_{s_i} \approx \frac{1}{N}\sum_i \nabla_a Q(s, a|\mathbf{\Theta}^Q)|_{s=s_i, a=\mu(s_i)} \nabla_{\mathbf{\Theta}^\mu}\mu(s|\mathbf{\Theta}^\mu)|_{s_i}$$

    Update the target networks:

$$\mathbf{\Theta}^{Q\prime} \leftarrow \tau\mathbf{\Theta}^Q + (1-\tau)\mathbf{\Theta}^{Q\prime}$$

$$\mathbf{\Theta}^{\mu\prime} \leftarrow \tau\mathbf{\Theta}^\mu + (1-\tau)\mathbf{\Theta}^{\mu\prime}$$

C. RBGNet

Traditional fully connected neural network is widely used to play the actor network in actor-critic algorithms, but it's limited by its fixed input and output.However, considering the real communication process, the quantity of users in the base station varying frequently, which causes the data shape of state is unfixed.For the sake of construction of network which can accept scalable input, the conventional neural network(CNN) is introduced to address the problem.CNN is a kind of feedforward neural networks with convolution computation and deep structure, which is one of the representative algorithms of deep learning.In the following sections, we are honored to introduce the *RBGNet* to you, a deep, scalable, fully conventional, joint-prediction network designed for RBG allocation.

1) Backbone: Figure 2 illustrates the architecture of proposed RBGNet. This networks consists of 11 conventional layers, which is fully conventional that it can accept scalable input. [13] We leverage 3 conventional layers to realize downsampling operation by customizing its convolutional parameters, like strides, padding mode, etc.Morever, the downsampling operation is only applied to the second dimension of input.According to the formula **(7)**, the output of this feature extractor is:

$$ds_{feature} = (K, 1) \tag{11}$$

That is to say, the data shape of output feature has nothing to do with the sum of attributes in the state but is only determined by the quantity of users.

The whole network is so deep that it's easy to get trouble with gradient vanishing, so that the residual structure is adopted to address the problem, which can prevent the network from gradient vanishing by identity mapping. [14]

In particular, to simplify the design of architectur, we let the actor and critic take the same feature extractor to handle the input, then setting special layers to achieve different functions.

D. Actor network and Critic network

For actor network, a conventional layer with N_{RBG} filters and softmax function is set as the output layer, and the data shape of output is:

$$ds_{actor\ output} = (K, 1, N_{RBG}) = (K, N_{RBG}) \tag{12}$$

Based on this configuration, each RBG was allocated by one independent filter but the prediction is still joint.After being processed by softmax function, the final output is a matrix with shape (K, N_{RBG}), the RBG will be allocated to the highest scoring user.

For critic network, we leverage an unique operation named *Global Pooling* to process the feature from backbone. [15]Figure 3 illustrates the workflow of *Global Pooling*.Thus the output of *Global Pooling* is only related to the quantity of filters in the last convolution layer of feature extractor, so that the critic network can also accept scalable input.Finally, this map is received by a fully connected layer to predict the reward.

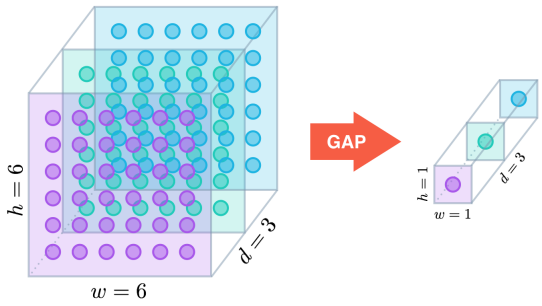

Fig. 3. Architecture of RBGNet.The parameters of downsampling layers were configured skillfully based on $d_{feature}$

E. Training strategy

The training of reinforcement learning is a difficult and long-time job. In the early stage of the research, two models were established to solve the problem of full buffer case and bursty case respectively.

Algorithm 2 Training strategy

function TRAIN($step, \mathbf{\Theta}_{RL}, \mathbf{\Theta}_{Oppo}, \varphi$)
  Initialize $p(\mathbf{\Theta}_{RL}) \leftarrow 0, p(\mathbf{\Theta}_{Oppo}) \leftarrow 0$, where $p()$ is the performance of agent.
  while $|p(\mathbf{\Theta}_{RL}) - p(\mathbf{\Theta}_{Oppo})| > \varphi$ **do**
    for Epoch=1, M **do**
      Two agents interact with the environment to generate available experience E_{RL} and E_{Oppo}.
      if step=1 **then**
        Update $\mathbf{\Theta}_{RL}$ with $(E_{RL} \bigcup E_{Oppo})$.
      else
        Update $\mathbf{\Theta}_{RL}$ with E_{RL}.
      Update $p(\mathbf{\Theta}'_{RL})$ and $p(\mathbf{\Theta}'_{Oppo})$: $p(\mathbf{\Theta}_{RL}) \leftarrow p(\mathbf{\Theta}'_{RL}), p(\mathbf{\Theta}_{Oppo}) \leftarrow p(\mathbf{\Theta}'_{Oppo})$
    if $p(\mathbf{\Theta}'_{RL}) > p(\mathbf{\Theta}_{RL})$ **then**
      Update $\mathbf{\Theta}_{RL}$: $\mathbf{\Theta}_{RL} \leftarrow \mathbf{\Theta}'_{RL}$
    else
      continue
  return agent $\mathbf{\Theta}_{RL}$
Step 1: full buffer case training
Initialize agent $\mathbf{\Theta}_{RL}$ and $\mathbf{\Theta}_{Oppo}$, threshold φ_1.
$\mathbf{\Theta}^{s1}_{RL} \leftarrow$TRAIN(1, $\mathbf{\Theta}_{RL}, \mathbf{\Theta}_{Oppo}, \varphi_1$)
Step 2: bursty case training
Initialize agent:$\mathbf{\Theta}_{RL} \leftarrow \mathbf{\Theta}^{s1}_{RL}$, $\mathbf{\Theta}_{Oppo}$, threshold φ_2.
$\mathbf{\Theta}^{s2}_{RL} \leftarrow$TRAIN(2, $\mathbf{\Theta}_{RL}, \mathbf{\Theta}_{Oppo}, \varphi_2$)

For bursty case, the agent can explore any possible solutions to maximize the AUDR since the optimal solutions is uncertain, and the performance of the agent successfully surpasses the *Opportunistic Schedule* after lots of training.

For full buffer case, the agent was hoped to achieve the effect close to that method, but we've found it's difficult to achieve that only by the agent's own exploration. *Opportunistic Schedule* is a fixed method, so it's difficult and impractical for a network with millions of parameters to learn a fixed method by trial-and-error. Howerver, it's clear that the *Opportunistic Schedule* is the optimal solution to maximize the AUDR, which provides the best experience to guide the agent. Therefore, the **Mixed Experience** was introduced to train the agent, which is composed of the experience both from self-exploration and *Opportunistic Schedule*. With lots of training and *fine-tuning* [16], the agent successfully simulated the *Opportunistic Schedule*.

In order to make the model compatible with both full buffer case and bursty case, a step-by-step training strategy was introduced to address this problem. First of all, the agent is trained with **Mixed Experience** in full buffer case, which can simulate the behavior of *Opportunistic Schedule*. At this stage, the *fine-tuning* operation should be repeated again and again to get the best performance.

Secondly, using bursty case to train the previous model with low learning rate.Now, the experience is not mixed, which is only from the self-exploration.Through this step-by-step training strategy, the scheduling algorithm based on reinforcement learning has successfully surpassed the *Opportunistic Schedule* in AUDR.

V. Numerical Results

A. Beach Mark Schemes

1) Opportunistic User Scheduling: As the optimal scheduling in terms of systematic throughput maximization, opportunistic user scheduling is quite straightforward, which can be expressed as

$$P^*(k) = \arg\max_n R_{n,k}. \tag{13}$$

It means each RBG selects the user suggesting the highest estimated data rate, which also implies the optimal scheduling in achieving the highest AUDR in full buffer case.

2) Propotional Fairness User Scheduling: To balance the system throughput and fairness, a user scheduling called proportional fairness is proposed in [2], where the scheduling is achieved by allocate RBGs to different UEs. Any UE with a non-empty buffer on the BS side is a candidate to each RBG, and each RBG independently sort all UEs according to their PF values. Denote the PF value of UE n on RBG k within transmission time interval (TTI) i by $\beta_{n,k}[i]$ computed as

$$\beta_{n,k}[i] = \frac{R_{n,k}[i]}{\bar{T}_n[i]}, \tag{14}$$

where $\bar{T}_n[i]$ is the UE's moving average historical throughput which can be expressed as

$$\bar{T}_n[i] = (1-\gamma)\,\bar{T}_n[i-1] + \gamma T_n[i-1]. \tag{15}$$

In Eq. (15), γ is the moving average coefficient, which is usually very small, and $T_n[i-1]$ is the actual TB size of UE n in TTI $i-1$ interpreted as how many number of bits is sent in last TTI. We sometimes omit the TTI index for brevity in the sequel. The UEs with higher PF value are with higher priorities to capture the RBG, and the PF value of a UE to different RBGs can be different. Specifically, the PF scheduling can be expressed as

$$P^*(k) = \arg\max_n \beta_{n,k}, \tag{16}$$

B. Performance Comparison

In the following, the simulator sets 50 RBs and 17 RBGs using a frequency bandwidth of 10 MHz. The noise power density is at -174 dBm/Hz, and the transmit power of each RB is fixed at 18 dBm. All performances of RL is based on a well-trained RBGNet. For both full buffer and bursty buffer cases, 100 simulations are realized and all performance data are collected independently.

The table II shows the out-performance statistics among the three user schedulings. The number in each cell indicates the percentage that the former scheduling outperforms the latter one. As expected, the opportunist scheduling beats RL and PF all in full buffer in terms of AUDR, and PF scheduling beats the rest in terms of 5%-tile, which testifies the validity of our simulation. We highlight that in bursty buffer, our proposed RBGNet has won 96% simulation against PF and 97% simulations against opportunistic scheduling.

TABLE II
OUTPERFORMANCE COMPARISON

	Full buffer case		**Bursty buffer case**	
	AUDR	**5%-tile**	**AUDR**	**5%-tile**
RL vs PF	96%	0%	**96%**	29%
RL vs Oppo	0%	1%	**97%**	79%
PF vs Oppo	0%	100%	66%	82%

The performance gain distribution is illustrated in Fig.4, where the blue line is with reference to opportunist scheduling and the red line is with reference to the PF scheduling. The performance discussed here is AUDR only. As can be seen from the figure, even though opportunistic scheduling outperforms RL in all 100 simulation in full buffer, the RL scheduling can still provide satisfactory solutions. The worst scenario only shows a 20% shy to the optimal, whereas actually the mean value of the AUDR degradation is 8.71% compared to the optimal. Nevertheless, in the bursty buffer case, the RL scheduling manifest a strong superiority against other scheduling, respectively achieving a 12.17% and 13.17% AUDR gain compared to opportunist and PF scheduling.

VI. CONCLUSION

In this paper, a model-free RL user scheduling algorithm is developed to pursue the highest AUDR, where the optimal

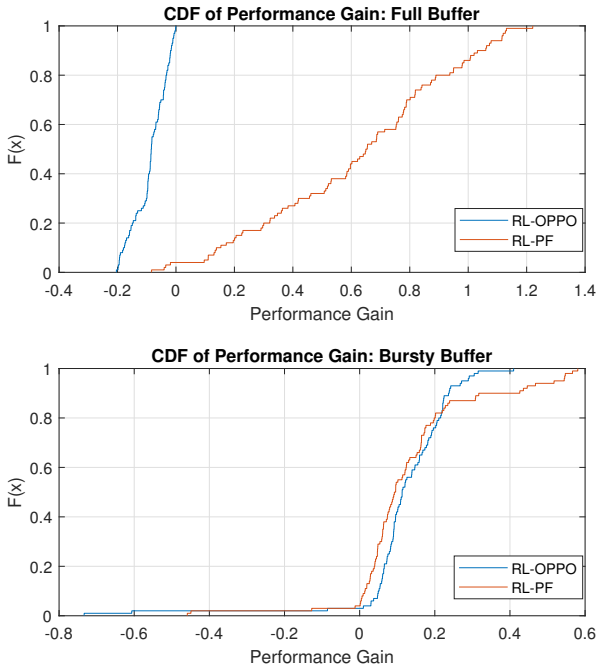

Fig. 4. CDF of Performance Gain

scheduling is never derived in bursty buffer case. In other words, the practical user scheduling aiming at AUDR is still a blank area. We then demonstrated the feature collection, derived a reward function and introduced the configuration of proposed RBGNet. Using our DDPG training algorithm, the RBGNet successfully converges and demonstrates outstanding performance with about 13% AUDR gain in contrast with existing schedulings.

REFERENCES

[1] S. Borst and P. Whiting, "Dynamic rate control algorithms for hdr throughput optimization," in *Proceedings IEEE INFOCOM 2001. Conference on Computer Communications. Twentieth Annual Joint Conference of the IEEE Computer and Communications Society (Cat. No. 01CH37213)*, vol. 2, pp. 976–985, IEEE, 2001.

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

[3] F. P. Kelly, A. K. Maulloo, and D. K. Tan, "Rate control for communication networks: shadow prices, proportional fairness and stability," *Journal of the Operational Research society*, vol. 49, no. 3, pp. 237–252, 1998.

[4] H. Kim and Y. Han, "A proportional fair scheduling for multicarrier transmission systems," *IEEE Communications letters*, vol. 9, no. 3, pp. 210–212, 2005.

[5] Ö. F. Gemici, F. Kara, İ. Hökelek, and H. A. Çırpan, "User scheduling and power allocation for nonfull buffer traffic in noma downlink systems," *International Journal of Communication Systems*, vol. 32, no. 1, p. e3834, 2019.

[6] Y. Saito, A. Benjebbour, A. Li, K. Takeda, Y. Kishiyama, and T. Nakamura, "System-level evaluation of downlink non-orthogonal multiple access (noma) for non-full buffer traffic model," in *2015 IEEE Conference on Standards for Communications and Networking (CSCN)*, pp. 94–99, IEEE, 2015.

[7] P. Ameigeiras, Y. Wang, J. Navarro-Ortiz, P. E. Mogensen, and J. M. Lopez-Soler, "Traffic models impact on ofdma scheduling design," *EURASIP Journal on Wireless Communications and Networking*, vol. 2012, no. 1, p. 61, 2012.

[8] H. Sun, X. Chen, Q. Shi, M. Hong, X. Fu, and N. D. Sidiropoulos, "Learning to optimize: Training deep neural networks for interference management," *IEEE Transactions on Signal Processing*, vol. 66, no. 20, pp. 5438–5453, 2018.

[9] R. Sato, M. Yamada, and H. Kashima, "Approximation ratios of graph neural networks for combinatorial problems," in *Advances in Neural Information Processing Systems*, pp. 4083–4092, 2019.

[10] R. Bruno, A. Masaracchia, and A. Passarella, "Robust adaptive modulation and coding (amc) selection in lte systems using reinforcement learning," in *2014 IEEE 80th Vehicular Technology Conference (VTC2014-Fall)*, pp. 1–6, IEEE, 2014.

[11] 3GPP, "The mobile broadband standard." https://www.3gpp.org/technologies/keywords-acronyms/97-lte-advanced, 2013.

[12] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," *Computer Science*, vol. 8, no. 6, p. A187, 2015.

[13] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3431–3440, 2015.

[14] K. He, X. Zhang, S. Ren, and S. Jian, "Deep residual learning for image recognition," in *IEEE Conference on Computer Vision & Pattern Recognition*, 2016.

[15] M. Lin, Q. Chen, and S. Yan, "Network in network," *Computer Science*, 2013.

[16] J. Howard and S. Ruder, "Universal language model fine-tuning for text classification,"