Enhancing User Fairness in OFDMA Radio Access Networks Through Machine Learning

Ioan-Sorin Comșa[a], Sijing Zhang[b], Mehmet Aydin[c], Pierre Kuonen[d], Ramona Trestian[e], and Gheorghiță Ghinea[a]
[a]Department of Computer Science, Brunel University London, Kingston Lane, UB8 3PH, London, U.K.
[b]School of Computer Science and Technology, University of Bedfordshire, LU1 3JU, Luton, U.K.
[c]Dept. of Computer Science and Creative Technologies, Univ. of the West of England, BS16 1QY, Bristol, U.K.
[d]Dept. of Communications and Information Technology, HEIA-FR, CH-1700, Fribourg, Switzerland
[e]Faculty of Science and Technology, Middlesex University London, NW4 4BT, Hendon, London, U.K.
E-mails: ioan-sorin.comsa@brunel.ac.uk, sijing.zhang@beds.ac.uk, mehmet.aydin@uwe.ac.uk,
pierre.kuonen@hefr.ch, r.trestian@mdx.ac.uk, george.ghinea@brunel.ac.uk

***Abstract*—The problem of radio resource scheduling subject to fairness satisfaction is very challenging even in future radio access networks. Standard fairness criteria aim to find the best trade-off between overall throughput maximization and user fairness satisfaction under various types of network conditions. However, at the Radio Resource Management (RRM) level, the existing schedulers are rather static being unable to react according to the momentary networking conditions so that the user fairness measure is maximized all time. This paper proposes a dynamic scheduler framework able to parameterize the proportional fair scheduling rule at each Transmission Time Interval (TTI) to improve the user fairness. To deal with the framework complexity, the parameterization decisions are approximated by using the neural networks as non-linear functions. The actor-critic Reinforcement Learning (RL) algorithm is used to learn the best set of non-linear functions that approximate the best fairness parameters to be applied in each momentary state. Simulations results reveal that the proposed framework outperforms the existing fairness adaptation techniques as well as other types of RL-based schedulers.**

***Index Terms*—RRM, Resource Scheduling, Fairness Optimization, Reinforcement Learning, Neural Networks.**

I. Introduction

The next generation of mobile networks (5G) brings a sustainable support for bandwidth-hungry and delay-sensitive applications over radio access interfaces by integrating hybrid solutions and key technologies, such as beamforming, massive antenna arrays or millimeter wave communications [1]. The network operators are facing the challenges of accommodating very high heterogeneity of applications and solutions as well as providing more stringent Quality of Service (QoS) requirements for their customers [2]. An important aspect of QoS provisioning is represented by the fairness objective satisfaction when delivering the requested heterogeneous services. On one hand, the inter-class fairness is addressed when adopting a certain prioritization order among different application classes according to the stringency of their QoS requirements. On the other hand, the intra-class fairness must be adopted for users belonging to the same application class to avoid the starvation of some users with unfavorable wireless connections. To this end, smart solutions are needed to maintain high satisfaction levels for both inter-class and intra-class fairness under dynamic network conditions.

One possible solution that could improve the fairness satisfaction under various networking conditions is the flexible management of radio resources [2]. A more dynamic Radio Resource Management (RRM) will enable a smarter mobility management, adaptive energy saving techniques and power allocation schemes and more intelligent packet scheduling and resource allocation algorithms [3]. A packet scheduler aims to allocate user data packets in the frequency domain at each Transmission Time Interval (TTI) to achieve different QoS targets [4]. At each TTI, the scheduling process is conducted through a scheduling rule that aims at assuring certain fairness levels depending on the actual networking conditions. Intelligent packet scheduler implies the possibility of adapting these scheduling rules to the network conditions such that the fairness satisfaction is improved [4].

We focus on the RRM scheduler for intra-class user fairness satisfaction. The intra-class fairness can be addressed by considering two aspects: *a)* the amount of allocated resources and *b)* the users$'$ average throughput. For the first approach, the classical Round-Robin scheduler aims to allocate equal amount of radio resources to all users while ignoring the channel and network conditions [5]. As a result, some resources may be wasted for those users with unfavorable channel conditions. As a second option of addressing the intra-class user fairness, the Proportional Fair (PF) is used to achieve certain fairness level between users$'$ average throughput [5]. The fairness performance of PF scheduling rule strongly depends on the channel conditions for each user, and then, an acceptable level of fairness would be difficult to be achieved for more general wireless conditions. In this sense, the Generalized Proportional Fair (GPF) scheduler is proposed as a parameterizable version of conventional PF scheduler that can adapt to the momentary networking conditions such that the fairness performance can be improved [4].

We propose the use of scheduling functions able to map the network conditions into parameterization decisions for the GPF scheduling rule at each TTI such that the user fairness would be greatly improved. In this sense, we use

This work has been performed in the framework of the Horizon 2020 project NEWTON (ICT-688503) receiving funds from the European Union. The authors would like to acknowledge the contributions of their colleagues in the project, although the views expressed in this contribution are those of the authors and do not necessarily represent the project.

Reinforcement Learning (RL) [6] to learn the best parameterization scheme of GPF scheduler on each momentary scheduler state. Since the scheduler state space is continuous and multi-dimensional (i.e. traffic load, channel conditions, performance characteristics, mobility models), the fairness adaptation problem cannot be enumerated exhaustively. Thus, we propose the use of neural networks to approximate the best fairness parameterization decision at each momentary state.

II. Related Work

The intra-class user fairness can be evaluated by using quantitative and qualitative measures [7]. With quantitative evaluations, the average user throughput is used to compute certain fairness metrics. Jain Fairness Index (JFI) is the most well known example of quantitative measures [8]. The disadvantage of quantitative measures refers to the difficulty of setting fairness constraints globally accepted. Also, the individual user throughput is not related to the overall distribution of other users$'$ throughput. Qualitative fairness evaluation aims to reduce these drawbacks by considering each individual user throughput subject to certain fairness constraint. A typical example of qualitative measure is the Next Generation of Mobile Networks (NGMN) fairness criteria [9]. According to NGMN fairness requirement, a system is considered fair at each TTI only if at least $(100-x)\%$ of active users achieve at least $x\%$ of each normalized user throughput [9]. The scheduler is considered fair if the Cumulative Distribution Function (CDF) of users$'$ normalized throughput lies on the right side of NGMN requirement [9].

When optimizing the user fairness, a special attention must be given to the trade-off with other QoS objectives, such as system throughput maximization [10]. For example, high JFI index may indicate a good fairness at the cost of significant system throughput degradation. Also, when the CDF curve of normalized user throughput is too far from its NGMN requirement, the system can be over-fair and consequently, the system throughput can be affected [4]. In [11], different versions of PF scheduling rules are compared for orthogonal and non-orthogonal radio access schemes with imperfect channel state information and different trade-off designs on network throughput and JFI fairness. In [12], different parameterization schemes for GPF scheduling rule are tested for non-orthogonal access networks taking into account the joint power optimization problem and JFI fairness. In both cases, the obtained system throughput is higher than for orthogonal schedulers, but the proposed schedulers are unable to adapt online based on the momentary networking conditions, making the user fairness satisfaction questionable.

The JFI performance is measured in [13] by using the Simple Parameterization for GPF (SP-GPF) scheduling rule. The user throughput is predicted based on the probability mass function and consequently, the GPF parameter is adapted at the current TTI according to the predicted JFI value in the next TTI. The results suggest very good performance in terms of system throughput and user fairness trade-off but at the cost of higher system complexity. In [14], a fuzzy logic scheduler is proposed to deal with JFI and system throughput maximization. The Q-Learning algorithm with the neural networks as function approximations is used in [15] to achieve different static trade-off levels between system throughput and JFI. Imposing the fairness limit regardless the channel conditions makes the JFI-based approaches impractical.

The adaptation of NGMN qualitative fairness measure in OFDMA systems is considered in [16] in which the CDF curve is adjusted at each 1s by using the dynamic parameterization of SP-GPF scheduling rule. In [17], different RL algorithms are used to adapt the SP-GPF scheduling rule at each TTI based on the network conditions. The continuous actor-critic algorithm (CACLA-1) performs the best by increasing the number of TTIs when the system respects the NGMN fairness requirement. The Double Parameterization of GPF (DP-GPF) is addressed in [18], in which an actor-critic scheme with continuous action space is used to learn the best parameterization decision at each TTI. The actor-critic RL scheme for DP-GPF is able to outperform the RL algorithm with simple parameterization (CACLA-1) proposed in [17] with more than 6% of time when the NGMN fairness is achieved. Both parameterization algorithms proposed in [17] and [18] use the average user throughput based on Exponential Moving Filter (EMF) to compute the momentary CDF curve to match the NGMN requirement at each TTI. The EMF-based observations are used to compensate the wireless channel fluctuations rather than to measure the fairness performance [4].

In this paper, we use the Median Moving Filter (MMF) to compute the average user throughput that is used to evaluate the NGMN fairness performance. The actor-critic RL scheme is used to train the non-linear functions that map states to parameterization decisions for the DP-GPF scheduler. Also, the proposed framework is trained for different time window lengths for MMF in order to determine the optimal setting for which the NGMN fairness satisfaction is maximized.

III. Scheduler Model

We consider an OFDMA downlink transmission where the available bandwidth is divided in equal Resource Blocks (RBs), the minimum resource unit that can be allocated to one user at each TTI. Let us consider $\mathcal{B} = \{1, ..., B\}$ the set of RBs, where B is the maximum number of RBs for a given system bandwidth. We consider an User Equipment (UE) being characterized by homogeneous traffic. Let us define $\mathcal{U}_t = \{u_1, u_2, ..., u_{U_t}\}$ the set of UEs, where U_t is the maximum number of users at TTI t. The user set $\mathcal{U}_t$ is variable since users can change their status from idle/active and vice-versa during scheduling. For each user $u \in \mathcal{U}_t$, $T_u[t]$ is the user throughput at TTI t, $\overline{T}_u[t]$ is the average user throughput based on EMF and determined as [4]:

$$\overline{T}_u[t] = (1-\psi) \cdot \overline{T}_u[t-1] + \psi \cdot T_u[t], \tag{1}$$

where $\psi \in [0, 1]$ is the forgetting factor. On the other hand, the average throughput with MMF $\overline{\overline{T}}_u[t]$ is determined as [4]:

$$\overline{\overline{T}}_u[t] = 1/W \cdot \sum\nolimits_{x=0}^{W-1} T_u[t-x], \tag{2}$$

where W is the moving time window used to average the user throughput. We propose to calculate the median moving

window as a function that depends on the number of active users U_t and the maximum number of users U_{max} that can be scheduled at each TTI based on the system bandwidth:

$$W = [\rho \cdot U_t / U_{max}], \tag{3}$$

where $\rho \in \mathbb{R}^+$ is the windowing factor. We aim to find the optimal range of ρ for generalized network conditions such that the NGMN fairness performance is maximized.

The packet scheduler aims to allocate each RB $b \in \mathcal{B}$ to a given active user $u \in \mathcal{U}_t$ at each TTI such that the NGMN fairness criterion is respected. Let us define the CDF function for each normalized user throughput such as: $\Upsilon(\hat{T}_u) : \mathbb{R} \rightarrow [0,1]$, where $\hat{T}_u = \overline{\overline{T}}_u / \sum_{u'} \overline{\overline{T}}_{u'}$ is the normalized average user throughput with MMF over the sum of all average user throughputs. According to NGMN fairness requirement, there is a vector of CDF values $\Upsilon^R = [\Upsilon_1^R, ..., \Upsilon_U^R]$ that has to be respected at each TTI. The purpose of our framework is to decide each TTI the best parameterization scheme for DP-GPF scheduling rule such that $\Upsilon = [\Upsilon_1(\hat{T}_1), ..., \Upsilon_U(\hat{T}_U)]$ respects the NGMN requirement vector $\Upsilon^R[t]$.

We define by $\mathcal{P} = \{p_1, p_2, ..., p_P\}$ the set of parameterization schemes used to configure the DP-GPF scheduling rule at each TTI. Based on the momentary networking conditions, a given decision $p \in \mathcal{P}$ provides certain fairness performance when considering the NGMN requirement. In this sense, we define the utility function $\Theta_{p,u,b} : \mathbb{R} \rightarrow \mathbb{R}$ that aims to quantify the parameterization decision $p \in \mathcal{P}$ for user $u \in \mathcal{U}_t$ and RB $b \in \mathcal{B}$. Actually, this utility outputs the priority of user u to be served on each RB b. Once the scheduling process is completed, the NGMN fairness performance is measured according to the new set of CDF values $\Upsilon[t+1]$. The idea is to parameterize the utility function Θ at each TTI in order to increase the number of TTIs when the NGMN fairness criterion is satisfied.

A. Optimization Problem

Solving the NGMN fairness optimization problem is challenging since alongside the simple resource allocation problem, the parameterization of utility functions has to be decided at each TTI. The proposed optimization problem is presented as follows:

$$\max_{x,y} \sum_{p \in \mathcal{P}} \sum_{u \in \mathcal{U}_t} \sum_{b \in \mathcal{B}} x_{p,u}[t] \cdot y_{u,b}[t] \cdot \Theta_{p,u,b}(\overline{T}_u, r_{u,b}) \cdot r_{u,b}[t], \quad s.t. \tag{4}$$

$$\sum\nolimits_u y_{u,b}[t] \leq 1, \quad b = 1, ..., B, \tag{4.a}$$

$$\sum\nolimits_p x_{p,u}[t] = 1, \quad u = u_1, ..., u_U, \tag{4.b}$$

$$\sum\nolimits_u x_{p^*,u}[t] = U_t, \quad p^* \in \mathcal{P}, \tag{4.c}$$

$$\sum\nolimits_u x_{p^\otimes,u}[t] = 0, \quad \forall p^\otimes \in \mathcal{P} \backslash \{p^*\}, \tag{4.d}$$

$$x_{p,u}[t] \in \{0,1\}, \quad \forall p \in \mathcal{P}, \forall u \in \mathcal{U}_t, \tag{4.e}$$

$$y_{u,b}[t] \in \{0,1\}, \quad \forall u \in \mathcal{U}_t, \forall b \in \mathcal{B}, \tag{4.f}$$

$$\Upsilon_u(\hat{T}_u) \leq \Upsilon_u^R, \quad u = u_1, ..., u_U, \tag{4.g}$$

where $r_{u,b}$ is the achievable user rate obtained according to the received Channel Quality Indicator (CQI) for each user u and RB b. In (4), $y_{u,b}[t]$ is the RB allocation variable (i.e. $y_{u,b}[t] = 1$ if RB b is allocated to user u and $y_{u,b}[t] = 0$, otherwise) and $x_{p,u}[t]$ is the parameterization decision variable (i.e. $x_{p,u}[t] = 1$ if the parameters set $p \in \mathcal{P}$ is selected to perform the scheduling for user u and $x_{p,u}[t] = 0$, otherwise). Constraints (4.a) allocate for each RB b at most one user. Constraints (4.b) show that only one parameterization scheme is decided for each user at each TTI. The same parameterization scheme is used for all active users at each TTI, statement denoted by constraints (4.c) and (4.d). Constraints (4.e) and (4.f) make the proposed optimization problem combinatorial. Finally, constraints (4.g) denote the NGMN fairness requirements.

The utility function from (4) is defined as [4]: $\Theta_{p,u,b}(\overline{T}_u, r_{u,b}) = r_{u,b}^{\beta_t - 1} / \overline{T}_u^{\alpha_t}$, where the parameterization decision is $p_t = (\alpha_t, \beta_t) \in \mathcal{P}$. When $p = (0,1)$ keeps constant for all TTIs, the optimization problem aims to maximize the system throughput. When $p = (1,1)$ is set for the entire transmission, users with the best metrics $r_{u,b}/\overline{T}_u$ are selected for each RB $b \in \mathcal{B}$, and the obtained scheduling rule is PF. The user fairness can be improved since users with lower average throughput are preferred to be scheduled. However, by increasing α and keeping $\beta = 1$ for the entire scheduling session, fairer schedulers can be obtained while degrading the system throughput. If the scheduler keeps $\beta = 1$ constant and α_t variable such as $p_t = (\alpha_t, 1)$, then the obtained scheme is SP-GPF. When both parameters are tunable each TTI such that $p_t = (\alpha_t, \beta_t)$, then the scheduling rule is DP-GPF. In this paper, we propose the use of actor-critic RL framework to learn over time the best fairness parameters for the DP-GPF scheduler at each TTI.

B. Proposed NGMN Fairness Requirement

Figure 1 presents a benchmark for the fairness evaluation for a 60-user scenario equally distributed from the base station to the edge of the cell with the radius of 1km [18]. Figure 1.a presents the quantitative measure for the trade-off between system throughput maximization and JFI-based user fairness. As expected, when $p = (0,1)$, the system throughput is maximized while the JFI fairness is very poor. The PF scheduling rule (dark blue points) achieves a certain trade-off while the maximum fairness scheduling rule ($p = (10,1)$) maximizes the JFI-based fairness measure at the cost of strongly degrading the overall system throughput. Trade-off values higher than those imposed by the maximum limit (black dotted curve) cannot be obtained. As mentioned, this quantitative evaluation (Fig. 1.a) is not able to provide a certain fairness requirement under dynamic networking conditions. When using the RL framework to learn the best fairness parameters for SP-GPF in order to meet the NGMN requirement, the obtained policy shows that the optimal values are ($\alpha \in [0.4; 0.5], \beta = 1$).

Figure 1.b shows the CDF representations of the same schedulers and the NGMN fairness requirement. At each TTI, the CDF values are calculated to verify the NGMN fairness condition (black continuous and oblique line). As shown in Fig. 1.b, the maximum throughput scheduler crosses the NGMN fairness requirement line and then, it is considered

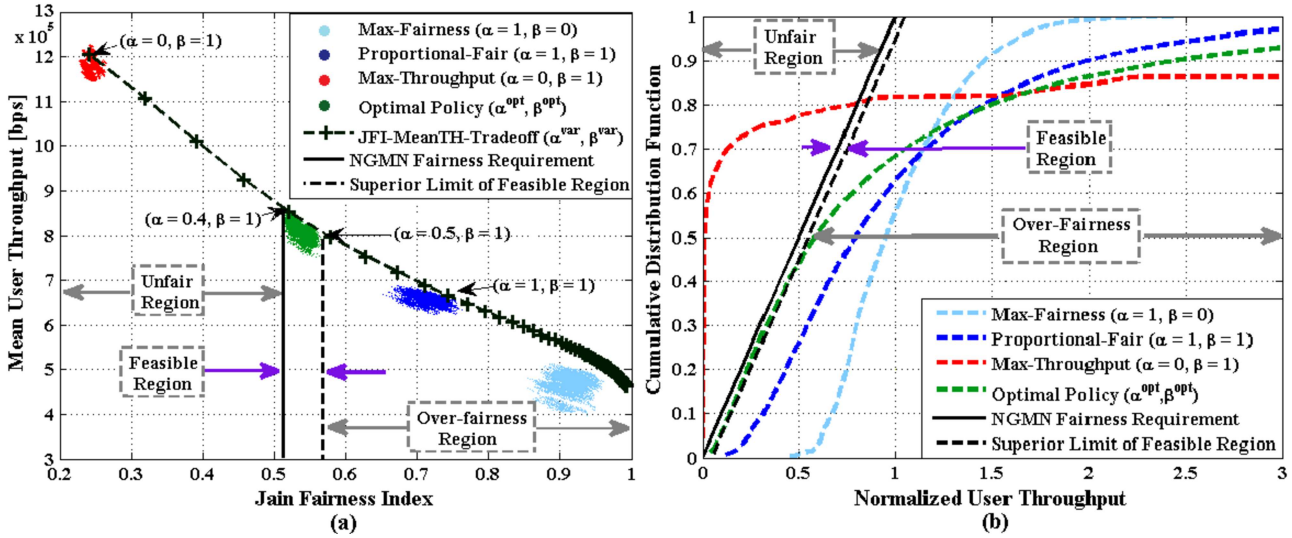

Fig.1 System Throughput Maximization vs. User Fairness [18]: a) Quantitative Evaluation; b) Qualitative Evaluation

unfair. The policy obtained by using reinforcement learning is fair since the corresponding CDF values are situated on the right side without crossing the NGMN requirement. The PF and maximum fairness schedulers are fair but the associated CDF curves are situated too far from the NGMN requirement. As seen from Fig. 1.a, this fact leads to serious degradation for the overall system throughput. The aim would be to learn a scheduling policy able to adapt to more general networking conditions and to have the CDF values at each TTI as close as possible to the NGMN requirement on the right side. Then, we need to define a maximum NGMN fairness limit for which the scheduler can be considered optimal or feasible. We define this maximum limit (oblique dotted line) as follows:

$$\Upsilon_u^M(\hat{T}_u) = \begin{cases} \Upsilon_u^R(\hat{T}_u) - \zeta, & \text{if } \hat{T}_u \leq 1+\zeta, \\ 1, & \text{if } \hat{T}_u > 1+\zeta, \end{cases} \quad (5)$$

where the NGMN requirement function is $\Upsilon_u^R(\hat{T}_u) = \hat{T}_u$ if $\hat{T}_u \leq 1$ and $\Upsilon_u^R(\hat{T}_u) = 1$, otherwise. The confidence factor $\zeta \in [0,1]$ sets the maximum limit for which the system is considered feasible. If the CDF curve exceeds this maximum limit on the right side, then the scheduler is considered over-fair. We propose the actor-critic RL framework to learn taking proper parameterization decisions, such that the CDF curves obtained at each TTI to be located in the CDF feasible zone.

At each TTI, for each user with the normalized throughput $\hat{T}_u \leq 1$, we calculate the difference $d_u = \Upsilon_u^R(\hat{T}_u) - \Upsilon_u(\hat{T}_u)$. If there is at least one user for which $d_u < 0$, then we declare the scheduler unfair. If $d_u \geq 0$ holds for each user, then the system is fair. In this case, we determine the maximum difference $d_{max} = max_u(d_u)$. If $d_{max} \leq \zeta$, then the scheduler is feasible; otherwise, when $d_{max} > \zeta$, the system is considered over-fair. We train our functions in order to minimize the number of TTIs when the system is unfair. Then, the second objective would be to decrease as much as possible the number of TTIs when the scheduler is over-fair.

C. Controller and Scheduler Interaction

When the scheduler is situated in one of the unfair or over-fair regions, parameters α_t and β_t must be adapted properly in order to reach the feasibility zone in the CDF domain as fast as possible. For example, as seen in Figs. 1.a and 1.b, if the scheduler is unfair, then α must increase and β decrease to reach the fairness zone. On the other hand, when the system is over-fair, α must decrease and β increase to get closer to the feasibility region. The adaptation of the parameterization scheme is achieved at each TTI based on the following recurrence:

$$p_t = \begin{cases} \alpha_t = \alpha_{t-1} + \Delta\alpha_t, \\ \beta_t = \beta_{t-1} + \Delta\beta_t, \end{cases} \quad (6)$$

where $\{\Delta\alpha_t, \Delta\beta_t\} \in [-1, 1]$ are the fairness steps that need to be decided at each TTI in order to respect the NGMN feasibility region. The fairness parameters are decided based on the interaction with an intelligent controller. The proposed controller makes use of an actor-critic RL framework that is able to learn over time the most suitable fairness steps to be applied on each momentary scheduler state.

IV. Proposed RL Framework

We propose the actor-critic RL framework to learn the non-linear functions for different settings of parameter ρ in order to approximate the best fairness steps $\{\Delta\alpha_t, \Delta\beta_t\}$ based on momentary scheduler states. The proposed framework works iteratively, such that: at each TTI t, the controller observes a momentary state and takes an action; the scheduler makes use of indicated fairness steps and performs the scheduling procedure; at TTI $t+1$, a new state is observed and the reward value is determined according to the calculated CDF values reported to the NGMN fairness requirement. Based on the goodness of the applied action in the previous state, the neural networks are reinforced and adapted accordingly.

A. States and Actions

We denote by $\mathbf{s}[t] \in \mathcal{S}$ the momentary scheduler state at TTI t, where $\mathcal{S}$ is the scheduler state space being considered as multi-dimensional with variable dimension. The dimension variability is given by the number of active users U_t that can change from one TTI to another. Under its original form, the scheduler momentary state can be represented as $\mathbf{s} = [\mathbf{s}_u, \mathbf{s}_c]$,

where $\mathbf{s}_u[t] \in \mathcal{S}_U$ is the uncontrollable scheduler state which is not depending on the applied fairness parameters. This sub-state contains the CQI indicators and the number of active users U_t. The controllable momentary state is represented by: $\mathbf{s}_c = [\alpha_{t-1}, \beta_{t-1}, \overline{\overline{\mathbf{T}}}[t], \overline{\mathbf{T}}[t], d]$, where $\overline{\overline{\mathbf{T}}}[t] = [\overline{\overline{T}}_1, ..., \overline{\overline{T}}_U]$, $\overline{\mathbf{T}}[t] = [\overline{T}_1, ..., \overline{T}_U]$ and d is determined as follows: if the scheduler is unfair, then $d = max_{u'}(d_{u'})$, where $u' \in \mathcal{U}_t$ are those users that are crossing the NGMN requirement on the left side; if the scheduler is fair, then $d = min_u(d_u)$. In this way, the controller is aware of how far the scheduler is from the feasible region. The dimension variability is given by the CQI sub-state, $\overline{\overline{\mathbf{T}}}[t]$, and $\overline{\mathbf{T}}[t]$. Compression methods from [4] are applied in order to get a fixed dimension for these sub-components. However, we are referring to $\mathbf{s}[t]$ as the momentary compressed scheduler state at TTI t.

The momentary controller action is defined as $\mathbf{a}[t] \in \mathcal{A}$, where $\mathcal{A}$ is continuous and two-dimensional action space and $\mathbf{a}[t] = [\Delta\alpha_t, \Delta\beta_t]$. Actually, the momentary controller action $\mathbf{a}[t]$ represents the output of the actor neural network.

B. Reward Functions

The reward evaluates at each TTI t the performance of applying action $\mathbf{a}[t] \in \mathcal{A}$ in momentary state $\mathbf{s} \in \mathcal{S}$. The reward is established based on how far the CDF curve is from the NGMN feasible zone at each TTI. For our convenience, we denote by $\mathbf{s} \in \mathcal{UF}$ the momentary state when the scheduler is unfair; $\mathbf{s} \in \mathcal{OF}$ when the scheduler is over-fair and $\mathbf{s} \in \mathcal{FS}$ when the CDF curve lies in the feasible zone. The proposed reward $\mathbf{r} : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ is calculated as follows:

$$\mathbf{r}_{t+1}(\mathbf{s}, \mathbf{a}) = \begin{cases} \mathbf{r}_u(\mathbf{s}, \mathbf{a}), & \textit{if } \mathbf{s} \in \mathcal{UF}, \\ 1, & \textit{if } \mathbf{s} \in \mathcal{FS}, \\ \mathbf{r}_o(\mathbf{s}, \mathbf{a}), & \textit{if } \mathbf{s} \in \mathcal{OF}, \end{cases} \tag{7}$$

where $\{\mathbf{r}_u, \mathbf{r}_o\}$ are the reward functions corresponding to unfair and over-fair regions, respectively.

When the the system is unfair, then the fairness decisions $\{\Delta\alpha_t, \Delta\beta_t\}$ taken in the current TTI should drive the scheduler to the fairness region. When $\beta_t > \alpha_t$, then the trade-off is balanced more on the throughput maximization and less on user fairness. Consequently, β_t must be decreased and α_t increased. When $\beta_t \leq \alpha_t$, then only α_t should be increased in order to reach the fairness region. When $\beta_t > \alpha_t$, the reward function associated to the unfair region is:

$$\mathbf{r}_u = \begin{cases} -1, & \textit{if } \Delta\alpha_t \leq 0,\ \Delta\beta_t \geq 0, \\ -0.5 \cdot (1 + |\Delta\beta_t|), & \textit{if } \Delta\alpha_t \leq 0,\ \Delta\beta_t < 0, \\ -0.5 \cdot (|\Delta\alpha_t| + 1), & \textit{if } \Delta\alpha_t > 0,\ \Delta\beta_t \geq 0, \\ 0.5 \cdot (|\Delta\alpha_t| + |\Delta\beta_t|), & \textit{if } \Delta\alpha_t > 0,\ \Delta\beta_t < 0. \end{cases} \tag{8}$$

When $\alpha_t \geq \beta_t$, then the proposed reward function becomes:

$$\mathbf{r}_u = \begin{cases} \Delta\alpha_t, & \textit{if } \Delta\alpha_t > 0, \\ -\Delta\alpha_t, & \textit{if } \Delta\alpha_t \leq 0. \end{cases} \tag{9}$$

When the momentary scheduler state is $\mathbf{s}[t] \in \mathcal{OF}$, then the fairness parameters must be adapted in order to increase the system throughput and decrease the JFI-based user fairness. We define two cases: a) when $\alpha_t \geq \beta_t$, the best practice is to decrease α_t and increase β_t; b) when $\alpha_t < \beta_t$, we must decrease β_t until the feasibility is met. For the first case, the reward associated to the over-fair region becomes:

$$\mathbf{r}_o = \begin{cases} -1, & \textit{if } \Delta\alpha_t \geq 0,\ \Delta\beta_t \leq 0, \\ -0.5 \cdot (1 + |\Delta\beta_t|), & \textit{if } \Delta\alpha_t \geq 0,\ \Delta\beta_t > 0, \\ -0.5 \cdot (|\Delta\alpha_t| + 1), & \textit{if } \Delta\alpha_t < 0,\ \Delta\beta_t \leq 0, \\ 0.5 \cdot (|\Delta\alpha_t| + |\Delta\beta_t|), & \textit{if } \Delta\alpha_t < 0,\ \Delta\beta_t > 0. \end{cases} \tag{10}$$

For the second case when $\beta_t > \alpha_t$, the associated reward is determined based on the following formula:

$$\mathbf{r}_o = \begin{cases} \Delta\beta_t, & \textit{if } \Delta\beta_t > 0, \\ -\Delta\beta_t, & \textit{if } \Delta\beta_t \leq 0. \end{cases} \tag{11}$$

The general idea is to learn the RL framework in such a way that the number of punishment rewards ($\mathbf{r} < 0$) and moderate rewards ($0 \leq \mathbf{r} < 1$)) is minimized and the number of maximum rewards ($\mathbf{r} = 1$) is maximized.

C. Value and Action-Value Functions

According to the tuple $\{\mathbf{s}[t], \mathbf{a}[t], \mathbf{r}_{t+1}, \mathbf{s}[t+1]\}$ received at each iteration, the RL framework aims to learn over time the best fairness decisions to be applied each TTI. We propose to use the actor-critic approach that makes use of two functions: a) value or critic function that keeps track of the value of the states and criticize the actions; b) action-value or actor function that aims to learn over time the best parameters to be applied in each state. As per original definition, the value function $V : \mathcal{S} \rightarrow \mathbb{R}$ is determined as follows [19]:

$$V(\mathbf{s}) = \mathbb{E}\Big[\sum\nolimits_{t=0}^{\infty} \gamma^t \mathcal{R}_{t+1} | \mathbf{s}[0] = \mathbf{s}\Big], \tag{12}$$

where, $\mathcal{R}_{t+1} = \mathbf{r}(\mathbf{s}, \mathbf{a})$; $(\gamma^t \mathcal{R}_{t+1}; t \geq 0)$ is the accumulated reward value being averaged from state to state by the discount factor $\gamma \in [0, 1]$; $\mathbf{s}[0]$ is considered as random such that $\mathbb{P}(\mathbf{s}[0] = \mathbf{s}) > 0$ holds for every $\mathbf{s} \in \mathcal{S}$. The action-value function $Q : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}^2$ considers in addition that the first action $\mathbf{a}[0]$ of the whole process is randomly chosen, and then the function becomes [19]:

$$Q(\mathbf{s}, \mathbf{a}) = \mathbb{E}\Big[\sum\nolimits_{t=0}^{\infty} \gamma^t \mathcal{R}_{t+1} | \mathbf{s}[0] = \mathbf{s}, \mathbf{a}[0] = [\Delta\alpha_0, \Delta\beta_0]\Big]. \tag{13}$$

In order to update these functions in each state, we use the recursions from [19] and the value function becomes:

$$V(\mathbf{s}) = \mathbf{r}(\mathbf{s}, \mathbf{a}) + \gamma \cdot V(\mathbf{s}'), \tag{14}$$

where $\mathbf{s}' = \mathbf{s}[t+1]$. For our convenience, we consider $\mathbf{s}' \in \mathcal{S}$ the current state and $\mathbf{s} = \mathbf{s}[t]$ the previous one.

D. Neural Network Approximations

Ideally, we would like to learn the optimal functions $V^*(\mathbf{s})$ and $Q^*(\mathbf{s}, \mathbf{a})$ that can provide the best values every state. But, both states and actions are multi-dimensional and continuous variables. Hence, we can only learn these functions and get over time the best approximations of optimal values. Then, we define by $\bar{V}^*(\mathbf{s}) : \mathcal{S} \rightarrow [-1; 1]$ the approximation of optimal critic function and by $\bar{Q}^*(\mathbf{s}) : \mathcal{S} \rightarrow [-1; 1]^2$ the approximation for the optimal action-value function. These

functions are non-linear representations defined as follows:

$$\begin{aligned} \bar{V}^*(\mathbf{s}) &= h^v(\theta_t^v, \psi(\mathbf{s})), \\ \bar{Q}^*(\mathbf{s}) &= h^a(\theta_t^a, \psi(\mathbf{s})), \end{aligned} \tag{15}$$

where $\{h^v, h^a\}$ are the neural networks for the value and action-value functions, respectively, $\psi(\mathbf{s})$ is the feature vector and $\{\theta_t^s, \theta_t^s\}$ are the weights$'$ vectors that must be learned by the actor-critic RL algorithm in order to get the best approximations for the optimal functions.

In general, a neural network is composed by L layers and N_l number of nodes for each layer $l \in \{1, 2, .., L\}$. Layers $l-1$ and l are interconnected with weights while each node is characterized by a non-linear transformation. Before the learning stage takes place, the best set of parameters in terms of $(L, N_l), l = \{1, 2, ..., L\}$ must be a priori decided.

E. Continuous Actor-Critic Learning Automata (CACLA)

CACLA algorithm aims to train the neural network weights at each TTI according to the received tuple $\{\mathbf{s}, \mathbf{a}, \mathbf{r}_{t+1}, \mathbf{s}'\}$. The learning stage is conducted by reinforcing at each TTI two errors through the critic and actor neural networks.

1) Critic Error: At each TTI, the state values $\{\bar{V}^*(\mathbf{s}), \bar{V}^*(\mathbf{s}')\}$ are obtained based on the critic weights that are updated so far. The critic error aims to find the impact of the applied action $\mathbf{a} \in \mathcal{A}$ in state $\mathbf{s} \in \mathcal{S}$ and it is determined according to the following formula [20]:

$$E_c(\mathbf{s}, \mathbf{s}') = V^T(\mathbf{s}') - \bar{V}^*(\mathbf{s}), \tag{16}$$

where $V^T(\mathbf{s}')$ is calculated in a way similar to (14). The critic error is back-propagated through the neural network reversely from the output to the input layer and the weights are updated based on the gradient descent principle [20].

2) Actor Error: If the critic error is $E_c < 0$, then the previous action is not a good choice and the actor neural network is not updated in order to avoid taking bad decisions in future. However, when $E_c \geq 0$, the actor neural network is updated by back-propagating the following actor error [20]:

$$E_a(\mathbf{s}) = 1/2 \cdot \sum\nolimits_{g=0}^{1} \left[a_g - \bar{Q}_g^*(\mathbf{s})\right]^2, \tag{17}$$

where $g = \{0, 1\}$ is the output index of actor neural network and $\mathbf{a}[t] = [a_g]$. The actor error is back-propagated in the same way by using the gradient descent principle.

In the learning stage, the RL framework must decide the strategy to follow at each TTI in terms of improvement and exploitation. When exploiting the actor neural network, the fairness parameters provided by the output layer are applied to the scheduling process and $\mathbf{a}[t] = \bar{Q}^*(\mathbf{s})$. If the improvement step is preferred, then a random set of fairness parameters $\mathbf{a} \in \mathcal{A}$ is used to enhance the NGMN fairness satisfaction. For an optimal learning, it is preferred to dynamically change the improvement and exploitation steps according to some probability distributions that are decided in advance [19].

V. Simulation Results

The packet scheduler and the proposed RL framework are implemented in RRM-Scheduler [4], a C/C++ object oriented tool that makes use of basic functions and methods imported from LTE-Sim simulator [21]. An infrastructure of 10 Intel(R) 4-Core(TM) machines with i7-2600 CPU at 3.40GHz, 64 bits, 8GB RAM and 120 GB HDD Western Digital storage is used to evaluate the performance of the proposed framework. First, the proposed actor-critic RL framework learns to approximate the best fairness parameters in each state; and second, the learnt functions are exploited and compared with the state-of-the-art schedulers and other RL-based approaches.

We compare our proposed actor-critic scheme entitled CACLA-2 (adapts both fairness parameters) with other RL algorithms proposed in [19]. From the literature review, we choose to compare our method with the most relevant approaches, such as: Maximizing Throughput (MT) [13] and Adaptive Scheduling (AS) [16]. To get a set of non-linear functions trained based on the same input observations but with different algorithms, we aim to use in the learning stage the same network conditions for CACLA-2 and other comparative RL frameworks. Also, the exploitation stage runs all approaches with the same conditions and the obtained results are averaged over 10 simulation runs and the STandard Deviations (STDs) are analyzed to prove the veracity of the proposed approach.

In order to find the optimum windowing factor ρ that can maximize the effective time when the scheduler state $\mathbf{s} \in \mathcal{FS}$ is feasible, different configurations are tested in the interval of $\rho \in [2.0; 5.5]$. Above this interval, the schedulers are not able to assure acceptable performance. For each of these settings, a different set of non-linear approximators is learnt each time. The NGMN confidence factor is set to $\zeta = 0.05$. The number of users is varied in the range of $U_t \in [15, 120]$ in both learning and exploitation stages. Based on a priori simulations, the optimum number for maximum schedulable users is $U_{max} = 10$. Each active user is characterized by homogeneous traffic with full buffer model. Once the optimum windowing factors are determined for this traffic type, the same values can be used for different types of traffic for the same range of users.

The considered system bandwidth is $20MHz$ with a number of $B = 100$ disposable RBs at each TTI. The cell radius is 1km, and we consider a cluster with 7 macro cells where the scheduling performance is evaluated in the central cell while other cells provide the interference levels. The user speed is 120km/h with random direction in both learning and exploitation stages in order to explore a high variety of CQI distributions. On top of that, the Jakes fading model is considered with a downlink power level of 43dBm for each resource block. The transmissions scheme is Frequency Division Duplex (FDD) and the CQI reports are considered full-band, periodical and error-less. The RLC layer is modeled by using the transmission mode without re-transmissions due to the fact that the results are oriented more on the decisions of fairness parameters for the first transmission.

A. Learning Stage

The duration of the learning stage is set to 3000s. Above this period, it was noticed that the critic error E_c keeps a constant level. When using higher numbers of layers and hidden nodes for both neural networks, it was noticed that

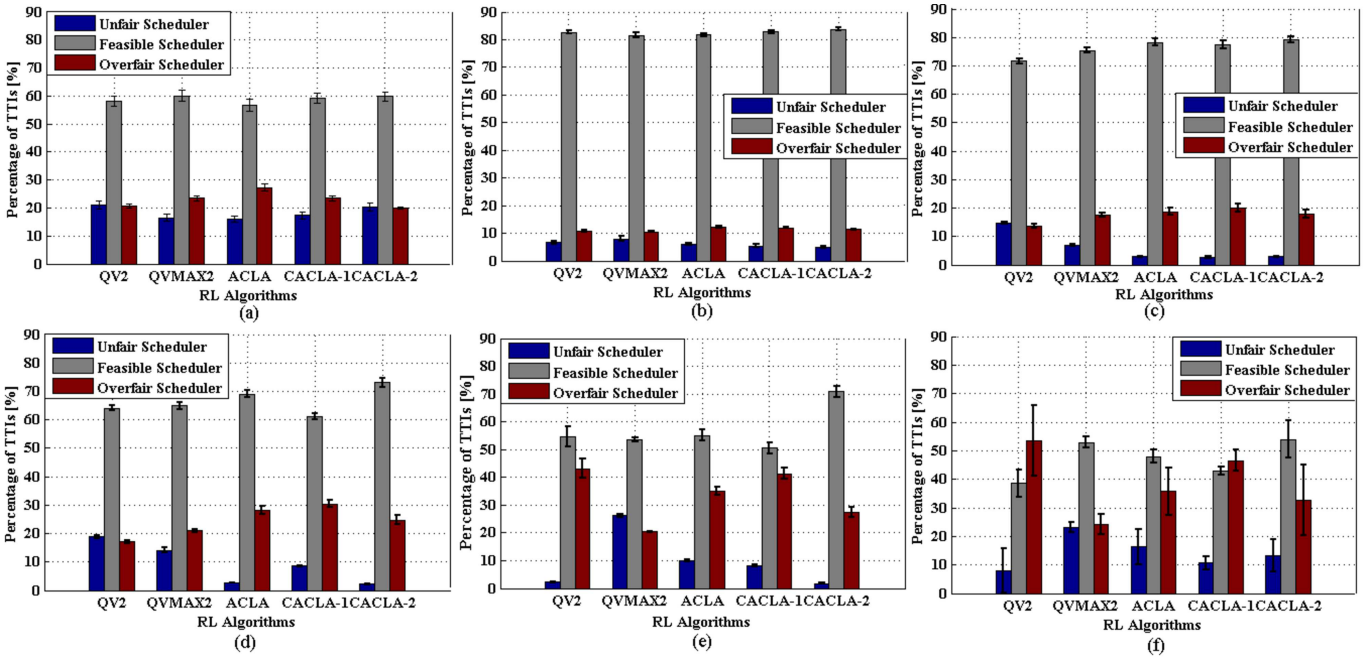

Fig.2 CACLA-2 vs. other RL Candidates: a) $\rho = 2.0$; b) $\rho = 3.0$; c) $\rho = 4.0$; d) $\rho = 4.5$; e) $\rho = 5.0$; f) $\rho = 5.5$

the system complexity is higher, the RL framework learns slower but the learnt structure is more flexible when deciding the fairness parameters. Also, a flexible structure presents higher risk of over-fitting the input data in the sense that, the learnt function represents very well the scheduler state but also the noisy data that can be inferred due to radio channel errors or when switching the number of active users. When using lower configurations in terms of numbers of hidden layers and hidden nodes (L, N_l), the framework can learn faster and the system complexity is lower but the learning performance is poorer since the learnt non-linear functions may not represent very well the entire scheduler state space, process entitled under-fitting. In both under-fitting and over-fitting cases, the critic error starts to increase at a certain point in the learning stage. According to a priori simulations, it is found that a configuration of $(L = 3, N = 60)$ is enough to represent the scheduler state for NGMN fairness criterion while avoiding the under-fitting and over-fitting. The activation functions for the nodes of input and output layers are linear. We use tangent hyperbolic representation for the nodes belonging to the hidden layer.

B. Exploitation Stage

For each of the considered scheduling schemes, we monitor the duration in the exploitation stage when the scheduler state is unfair ($\mathbf{s} \in \mathcal{UF}$), feasible ($\mathbf{s} \in \mathcal{FS}$), and over-fair ($\mathbf{s} \in \mathcal{OF}$). Let us denote by $p(\mathbf{s} \in \mathcal{UF})$ the mean value of the percentage of TTIs from the exploitation stage when the scheduler is unfair. By $p(\mathbf{s} \in \mathcal{OF})$ we denote the mean percentage of TTIs when the system is over-fair and $p(\mathbf{s} \in \mathcal{FS})$ when the scheduler CDF function is located in the feasible zone of the NGMN fairness requirement. For each mean values, we highlight the outliers in terms of standard deviations to test the stability of the learnt functions.

Figure 2 compares the actor non-linear function learnt by using CACLA-2 algorithm with other functions trained by RL algorithms applied in scheduling decision problems. RL algorithms such as QV2, QVMAX2 and ACLA [19] make use of predefined fairness steps $\{\Delta\alpha_t, \Delta\beta_t\}$ and the actions of these RL frameworks are discrete. CACLA-1 [17] parameterizes the SP-GPF scheduling rule for which $\beta = 1$ is static over the entire scheduling session and α_t is adapted based on continuous $\Delta\alpha_t$ steps at each TTI. When $\rho = 2.0$ (Fig. 2.a), QVMAX2 provides the best performance when measuring $p(\mathbf{s} \in \mathcal{FS})$ and $p(\mathbf{s} \in \mathcal{UF})$. CACLA-2 and CACLA-1 provide the best results in terms of the mean percentage of TTIs when the scheduler is unfair for $\rho \in \{3.0, 4.0\}$ (Figs. 2.b and 2.c). When $\rho = 4.5$ (Fig. 2.d), ACLA and CACLA-2 perform the best when measuring $p(\mathbf{s} \in \mathcal{FS})$ and $p(\mathbf{s} \in \mathcal{UF})$. By increasing the windowing factor to $\rho = 5.0$, CACLA-2 provides the highest amount of TTIs when the scheduler stays feasible while minimizing the number of TTIs when the unfair zones are detected. However, for larger time windows, such as $\rho = 5.5$ (Fig. 2.f), the stability of the RL-based schedulers is strongly affected, as indicated by the increasing STD values.

In Fig. 3, the proposed CACLA-2 framework is compared to other state-of-the-art approaches such as MT and AS adaptation schemes. It can be observed that for the entire domain of $\rho \in [2.0, 5.5]$, the proposed framework outperforms the other candidates when considering $p(\mathbf{s} \in \mathcal{UF})$ and $p(\mathbf{s} \in \mathcal{FS})$ as performance measures. When compared to AS scheme, maximum gains higher than 10% can be obtained when monitoring $p(\mathbf{s} \in \mathcal{FS})$ ($\rho = 5.0$) and higher than 60% when measuring $p(\mathbf{s} \in \mathcal{UF})$ ($\rho = 5.0$). When compared to MT adaptation scheme, the gains are much higher, such as: for example, when $\rho = 2.5$, gains higher than 30% when $p(\mathbf{s} \in \mathcal{FS})$ and gains of about 32% when $p(\mathbf{s} \in \mathcal{UF})$. For low time windows ($\rho \leq 2.0$), the performance of all adaptation techniques is affected due to high dynamics of channel conditions. For large windowing factors ($\rho \geq 5.5$), the NGMN fairness criterion is more difficult to be respected

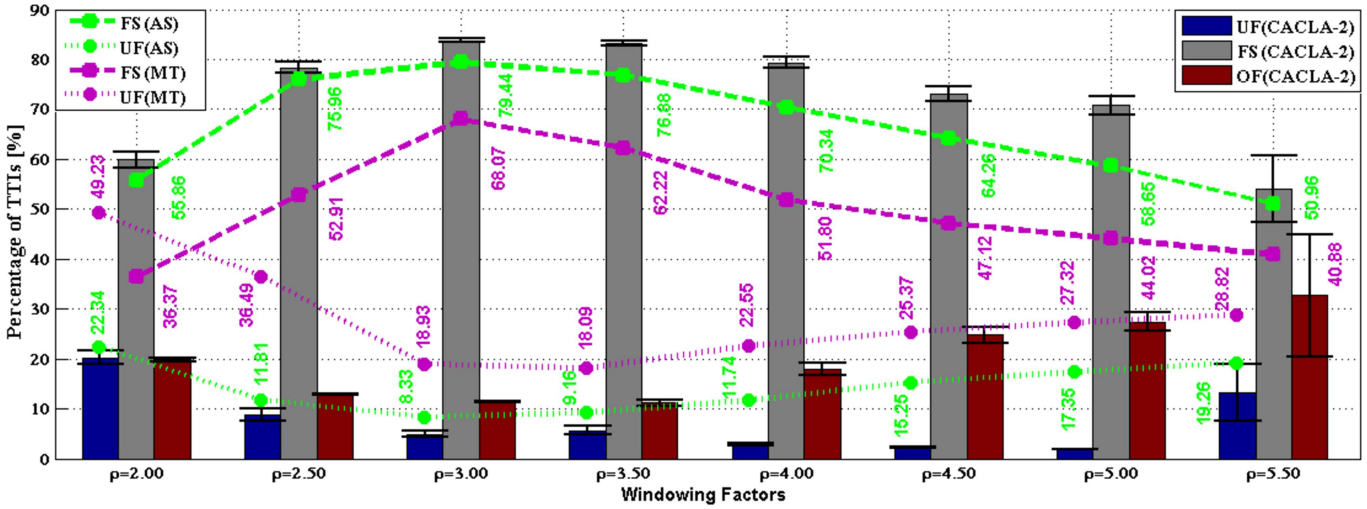

Fig.3 System Performance of the Proposed CACLA-2 Framework and State-of-the-Art Schedulers

since the impact of the parameterization decisions cannot be sensed immediately. To conclude, the optimal windowing factor must be chosen from $\rho \in (2.0; 5.5)$ for which CACLA-2 shows the best performance when the NGMN fairness requirement is considered.

VI. Conclusions

This paper proposes a dynamic scheduling framework able to adapt to the changeable networking conditions in order to maximize the satisfaction of NGMN fairness requirement. To deal with the framework complexity, neural networks are used to map the momentary scheduler states into parameterization decisions for the DP-GPF scheduler. The weights of neural networks are trained by using the continuous actor-critic RL algorithm with two-dimensional action space. The simulation results show the efficiency of the proposed actor-critic scheme when compared to other RL algorithms. When compared to other state-of-the-art adaptive schedulers, the proposed approach significantly increases the time when the scheduler is feasible for a larger range of windowing factors.

References

[1] J. Huang, C. X. Wang, R. Feng, J. Sun, W. Zhang, and Y. Yang, "Multi-frequency mmWave Massive MIMO Channel Measurements and Characterization for 5G Wireless Communication Systems,"*in IEEE Journal on Selected Areas in Communications*, vol. 35, no. 7, pp. 1591 – 1605, July 2017.

[2] J. G. Andrews, S. Buzzi, W. Choi, S. V. Hanly, A. Lozano, A. C. K. Soong, and J. C. Zhang, "What Will 5G Be?"*in IEEE Journal on Selected Areas in Communications*, vol. 3, no. 6, pp. 1065 – 1082, 2014.

[3] F. D. Calabrese, L. Wang, E. Ghadimi, G. Peters, L. Hanzo, and P. Soldati, "Learning Radio Resource Management in RANs: Framework, Opportunities, and Challenges,"*in IEEE Journal on Selected Areas in Communications*, vol. 56, no. 9, pp. 138 – 145, 2018.

[4] I.-S. Comșa, *Sustainable Scheduling Policies for Radio Access Networks Based on LTE Technology*. University of Bedfordshire, Luton, U.K., 2014.

[5] F. Capozzi, G. Piro, L. A. Grieco, G. Boggia, and P. Camarda, "Downlink Packet Scheduling in LTE Cellular Networks: Key Design Issues and a Survey,"*in IEEE Communications Surveys Tutorials*, vol. 15, no. 2, pp. 678 – 700, 2013.

[6] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. IMT Press Cambridge, 2012.

[7] H. Shi, R. V. Prasad, E. Onur, and I. Niemegeers, "Fairness in Wireless Networks: Issues, Measures and Challenges,"*in IEEE Communications Surveys and Tutorials*, vol. 16, no. 1, pp. 5–24, 2014.

[8] R. Jain, D. Chiu, and W. Hawe, "A Quantitative Measure of Fairness and Discrimination for Resource Allocation in Shared Computer Systems,"in *Technical Report TR-301*, September 1984, pp. 1 – 38.

[9] N. G. of Mobile Networks, "NGMN Radio Access Performance Evaluation Methodology,"January 2008, Accessed: December, 16, 2018. [Online]. Available: https://www.ngmn.org/fileadmin/user˙upload/NGMN˙Radio˙Access˙Performance˙Evaluation˙Methodology.pdf.

[10] F. Zabini, A. Bazzi, B. M. Masini, and R. Verdone, "Optimal Performance Versus Fairness Tradeoff for Resource Allocation in Wireless Systems,"*in IEEE Transactions on Wireless Communications*, vol. 16, no. 4, pp. 2587 – 2600, 2017.

[11] J. He, Z. Tang, Z. Tang, H.-H. Chen, and C. Ling, "Design and Optimization of Scheduling and Non-Orthogonal Multiple Access Algorithms with Imperfect Channel State Information,"*in IEEE Transactions on Vehicular Technology*, vol. 67, no. 11, pp. 10 800 – 10 814, 2018.

[12] M. Kaneko, H. Yamaura, Y. Kajita, K. Hayashi, and H. Sakai, "Fairness-Aware Non-Orthogonal Multi-User Access with Discrete Hierarchical Modulation for 5G Cellular Relay Networks,"*in IEEE Access*, vol. 3, no. 1, pp. 2922 – 2938, 2015.

[13] S. Schwarz, C. Mehlfuhrer, and M. Rupp, "Throughput Maximizing Multiuser Scheduling with Adjustable Fairness,"in *IEEE International Conference on Communications (ICC)*, June 2011, pp. 1 – 5.

[14] H. Soy and O. Ozdemir, "A Fuzzy Logic based Scheduling Approach to Improve Fairness in Opportunistic Wireless Networks,"in *IEEE 8th International Conference on Application of Information and Communication Technologies (AICT)*, October 2014, pp. 1 – 5.

[15] I.-S. Comșa, M. Aydin, S. Zhang, P. Kuonen, and J.-F. Wagen, "A Novel Dynamic Q-learning-based Scheduler Technique for LTE-advanced Technologies Using Neural Networks,"in *IEEE Local Computer Networks (LCN)*, October 2012, pp. 332–335.

[16] M. Proebster, C. M. Mueller, and H. Bakker, "Adaptive Fairness Control for a Proportional Fair LTE Scheduler,"in *IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, September 2010, pp. 1504 – 1509.

[17] I.-S. Comșa, M. Aydin, S. Zhang, P. Kuonen, J.-F. Wagen, and Y. Lu, "Scheduling Policies Based on Dynamic Throughput and Fairness Tradeoff Control in LTE-A Networks,"in *IEEE Local Computer Networks (LCN)*, September 2014, pp. 418–421.

[18] I.-S. Comșa, S. Zhang, M. Aydin, J. Chen, P. Kuonen, and J.-F. Wagen, "Adaptive Proportional Fair Parameterization Based LTE Scheduling Using Continuous Actor-Critic Reinforcement Learning,"in *IEEE Global Communications Conference (GLOBECOM)*, 2014, pp. 4387–4393.

[19] I.-S. Comsa, S. Zhang, M. Aydin, P. Kuonen, L. Yao, R. Trestian, and G. Ghinea, "Towards 5G: A Reinforcement Learning-based Scheduling Solution for Data Traffic Management,"*in IEEE Transactions on Network and Service Management*, vol. 15, no. 4, pp. 1661 – 1675, 2018.

[20] H. Van Hasselt and M. Wiering, "Using Continuous Action Spaces to Solve Discrete Problems,"in *International Joint Conference on Neural Networks*, 2009, pp. 1149 – 1156.

[21] G. Piro, L. A. Grieco, G. Boggia, F. Capozzi, and P. Camarda, "Simulating LTE Cellular Systems: An Open-Source Framework,"*in IEEE Transactions on Vehicular Networks*, vol. 60, no. 2, pp. 498 – 513, 2011.