Weskley V. F. Mauricio, Tarcisio F. Maciel, Anja Klein and F. Rafael M. Lima, "Learning-Based Scheduling: Contextual Bandits For Massive MIMO Systems," in *Proc. of the IEEE International Conference on Communications Workshops (ICC Workshops)*, June 2020.

# Learning-Based Scheduling: Contextual Bandits For Massive MIMO Systems

Weskley V. F. Mauricio*†, Tarcisio F. Maciel*, Anja Klein† and F. Rafael M. Lima*
*Wireless Telecomunnications Research Group (GTEL), Federal University of Ceará, Brazil. E-mail: {weskley,rafaelm,maciel}@gtel.ufc.br
†Communications Engineering Lab, TU Darmstadt, Germany. E-mail: a.klein@nt.tu-darmstadt.de

***Abstract*—Lately, Reinforcement Learning (RL) solutions appear as a tool with great potential to solve wireless communications problems. In this work, the scheduling problem in multiuser massive Multiple Input Multiple Output (MIMO) systems is investigated using RL-based techniques, wherein we propose a novel approach to multiuser scheduling in massive MIMO as a contextual bandit problem. The scheduler aims at maximizing the system throughput considering Quality of Service (QoS) constraints and multiple services. Firstly, we use the User Equipments (UEs)' spatial covariance matrices as the input of the K-means algorithm to split the UEs into spatially compatible clusters. Then, the scheduler defines each cluster as a virtual agent capable of making its own decision, which drastically reduces the search space. Lastly, the scheduler uses past information to learn how to satisfy the QoS requirements and maximize the system throughput. Our simulation results show that our solution outperforms a baseline algorithm obtaining 22.5% and 20% more throughput and system satisfaction, respectively. Furthermore, our solution also reduces the UEs' Channel State Information (CSI) feedback.**

***Index Terms*—Reinforcement Learning, Contextual Bandits, Scheduling, Massive MIMO.**

## I. Introduction

Massive Multiple Input Multiple Output (MIMO) is expected to play an important role in the 5$^{\text{th}}$ Generation (5G) of wireless communications [1]. With it, Base Stations (BSs) are equipped with tens to hundreds of antennas, enabling them to create many narrow beams to serve multiple User Equipments (UEs) at the same time-frequency Resource Block (RB). However, massive MIMO using classical digital beamforming requires a dedicated Radio Frequency (RF) chain per antenna, which leads to high hardware costs and low energy-efficiency. In order to reduce hardware cost and energy consumption, hybrid analog/digital beamforming appears as an alternative to the fully digital architecture [2]. Hybrid beamforming splits precoding into analog and digital domains, with beam generation performed by a concatenation of two matrices. The first one for the analog part, and the second one for digital beamforming with a reduced dimension compared to the analog part, which leads to a lower number of RF chains.

One of the major issues in Frequency Division Duplex (FDD) massive MIMO systems is the acquisition of Channel State Information (CSI) at the transmitter side, due to the massive number of antennas at BS [1]. Therefore, the schedulers in massive MIMO needs to operate under a lack of information. Reinforcement Learning (RL) tools emerge as a promising solution to lead with this issue since it is capable of works without accurate information and a complete model of the environment [3]. Also, RL solutions are very suitable for scheduling since it aims to make an agent learn how to behave in an environment to optimize a predetermined objective, such as the system performance metrics considered by the scheduler, such as Quality of Service (QoS) [3]. In the sequel, we will discuss the special case of RL solutions called Contextual Bandits (CB) [4]. The CB is a sequential decision-making solution where a learning agent has to take an action according to a context aiming at maximizing the long-term rewards and receives a reward for it. Then, at each time step, the agent repeats the process of observing contexts, taking actions, and receiving rewards accumulating information about them. Afterward, the agent has to estimate the action values, which determine the quality of the actions based on their average reward. Therefore, the agent learns through trials about how to make better long-term decisions.

## II. Related Works and Contributions of the Paper

The works in [1], [5]–[10] consider the two-stage precoding scheme to reduce the CSI feed back overhead of massive MIMO systems. Aiming at maximizing the system throughput, the authors in [1] propose a schedule based on Signal to Leakage plus Noise Ratio (SLNR). In [6], the authors proposed a schedule based on reinforcement learning aiming at maximizing the system throughput with fairness guarantee. In [7], the authors proposed a scheduling based on deep-learning for capacity, QoS, and coverage optimization. Aiming at maximizing the system throughput and fairness, the authors in [8] proposed a schedule based on graph theory. However, the authors in [1], [6]–[8] consider available the instantaneous channel of all UEs in the system. In [5], UEs are firstly clustered based on their spatial channel covariance using K-means. Afterward, a subset of the UEs of each cluster is pre-selected to feed back their CSI. This approach reduces the CSI feed back used by the scheduler to decide which ones are being scheduled. However, still, a considerable number of UEs needs to feed back their CSI. Also, another drawback of the aforementioned works [1], [5]–[8] is the consideration of full digital beamforming, which can be unpractical in massive MIMO systems. The works in [9], [10] propose new scheduling methods for hybrid precoding massive MIMO systems. In [9], after the clustering step, a different way to schedule UEs is proposed aiming at maximizing the system throughput. Therein, UEs are chosen based on a metric that balances

spatial channel correlation and channel gain using a parameter $\beta$, whose optimal value is scenario-dependent and therefore chosen based on trials. In [10], the authors proposed a new scheduling method to maximize system throughput. However, the works [9], [10] do not deal with QoS nor multiple service scenarios that are mandatory features of modern wireless networks.

Therefore, in this work, we proposed novel RL-based scheduling for massive MIMO systems that jointly consider hybrid precoding, QoS requirements, multiple services and a scheduler that does not use instantaneous CSI. Firstly, we apply a clustering algorithm to split the UEs into low-correlated clusters based on their statistical channel. After that, we consider the BS as a physical agent responsible for scheduling the UEs and each cluster as a logical virtual agent responsible for selecting the UEs to be scheduled at the BS. Each virtual agent acts independently without sharing information, and the compatibility check to select the UEs is made based on the sum throughput (reward) of UEs scheduled together in the past. This strategy drastically reduces the scheduler search space since the selection of UEs is divided among different clusters. Next, the scheduler uses the outdated CSI (context) of the previously scheduled UEs to model a utility function determining weights to the UEs based on their QoS requirements. Also, depending on the system requirements, the proposed scheduler is dynamically adapted to prioritize either the system throughput or QoS satisfaction. Afterward, only the scheduled UEs feed back their CSI, which leads to a reduction in the signaling overhead. We show in our results that this approach learns fast and outperforms the baseline solution in terms of system throughput and QoS provisioning.

## III. System Model

In this work, we consider the downlink (DL) of an Orthogonal Frequency Division Multiple Access (OFDMA) system composed of a single cell whose BS is equipped with a Uniform Planar Array (UPA) having a large number $N$ of antenna elements. The BS covers a cell sector serving $J$ omnidirectional single-antenna UEs, which are distributed uniformly within hot spots of a given radius. The hotspots, for instance, are evenly distributed within the sector area. We also assume $S$ different services which have different throughput requirements. For simplicity, we consider that each user is using only one service.

In the sequel, we define the signal model and related metrics adopted in this work. Let $\boldsymbol{H} \in \mathbb{C}^{J\times N}$ denote the DL channel frequency response between the $N$ antennas of the BS and the $J$ single-antenna UEs for a given OFDMA subcarrier, whose index is omitted herein and in the sequel for simplicity of notation. Notice that in this work, we adopt row vectors to represent the channels of the UEs. Indeed, each row of $\boldsymbol{H}$ corresponds to the channel between the BS antennas and the the $j$-th UE.

We consider an RB composed of $N_{\text{sc}}$ adjacent OFDMA subcarriers and $N_{\text{symb}}$ consecutive Orthogonal Frequency Division Multiplexing (OFDM) symbols and consider the channel frequency response to be nearly flat within a RB. Thus, we represent $\boldsymbol{H}$ for an RB by the frequency response of its middle subcarrier and first OFDM symbol. In the sequel, when referring to channel and precoding matrices, these will be associated to a specific RB.

We consider that the BS can spatially multiplex $K$ different UEs at the same RB using a linear precoding matrix $\boldsymbol{W} \in \mathbb{C}^{N\times K}$ to transmit the information vector $\boldsymbol{x} \in \mathbb{C}^{K\times 1}$ to $K$ UEs (selected out of the $J$ ones). Since we consider hybrid precoding [11], $\boldsymbol{W} = \boldsymbol{W}_{\text{RF}}\boldsymbol{W}_{\text{BB}}$ is composed of the product of two factors: $\boldsymbol{W}_{\text{RF}} \in \mathbb{C}^{N\times K}$, which is the analog precoder, and $\boldsymbol{W}_{\text{BB}} \in \mathbb{C}^{K\times K}$ which is the digital precoder.

Through this approach, we can create a reduced equivalent channel $\boldsymbol{H}_{\text{eq}} = \tilde{\boldsymbol{H}}\boldsymbol{W}_{\text{RF}} \in \mathbb{C}^{K\times K}$ to be used as the effective DL channel, where $\tilde{\boldsymbol{H}} \in \mathbb{C}^{K\times N}$ is the channel matrix of the $K$ UEs selected out of the $J$ existing UEs. The matrix $\tilde{\boldsymbol{H}}$ is formed by taking the $K$ rows of $\boldsymbol{H}$ corresponding to the $K$ UEs selected out of the $J$ existing ones. Indeed, $\boldsymbol{H}_{\text{eq}}$ has a much lower dimension than $\tilde{\boldsymbol{H}}$, since, in general, $K \ll N$.

As previously mentioned, we assume that $\boldsymbol{H}$ is nearly constant during the $N_{\text{symb}}$ OFDM symbols of an RB, i.e., during one Transmission Time Interval (TTI). Moreover, we assume that $\|\boldsymbol{W}_{\text{RF}}\boldsymbol{W}_{\text{BB}}\sqrt{\boldsymbol{P}}\|_{\text{F}}^2 = P_{\text{RB}}$ to satisfy the power constraint, where $\boldsymbol{P} \in \mathbb{R}_{+}^{K\times K}$ is a diagonal power matrix with the power allocated to each selected UE and $P_{\text{RB}}$ is the power available at the BS for an RB. More details on the definitions of $\boldsymbol{W}_{\text{RF}}$ and $\boldsymbol{W}_{\text{BB}}$ will be given in Section V.

The receive signal vector $\boldsymbol{y} \in \mathbb{C}^{K\times 1}$ of the $K$ served UEs is given by

$$\boldsymbol{y} = \tilde{\boldsymbol{H}}\boldsymbol{W}\sqrt{\boldsymbol{P}}\boldsymbol{x} + \boldsymbol{z}, \tag{1}$$

where $\boldsymbol{z} \in \mathbb{C}^{K\times 1}$ is an additive Gaussian noise vector whose elements are Independent and Identically Distributed (IID) as $\mathcal{CN}(0, \sigma^2\boldsymbol{I}_K)$ and $\boldsymbol{I}_K$ is a $K\times K$ identity matrix, with standard deviation $\sigma$.

Now, defining $\boldsymbol{M} = [m]_{i,j} = \tilde{\boldsymbol{H}}\boldsymbol{W}\sqrt{\boldsymbol{P}} \in \mathbb{C}^{K\times K}$, we can calculate the average Signal to Interference-plus-Noise Ratio (SINR) perceived by the $k$-th selected UE as

$$\gamma_k = \frac{\left|m_{k,k}\right|^2}{\sigma^2 + \sum\limits_{j\neq k}^{K}\left|m_{k,j}\right|^2}. \tag{2}$$

We consider that the data rate $r_k$ of the $k$-thUE on an RB is given according to Shannon's formula and is upper bounded by the data rate achievable using 256-Quadrature Amplitude Modulation (QAM), which is the highest modulation order supported by 5G New Radio (NR) systems [12], i.e.,

$$r_k = N_{\text{sc}}N_{\text{symb}}\min\left\{\log_2(1+\gamma_k), 8\right\} \text{ bits/TTI}. \tag{3}$$

## IV. User Clustering

Among the many existing clustering methods for Multi-User (MU) MIMO systems, we focus here on [5], which uses the statistical CSI as the main input for the classical K-means user clustering algorithm [5]. For MU MIMO systems, the main motivation to split the UEs into clusters is to reduce the

problem search space by selecting the served UEs individually per cluster [9], as we are going to explain later in Section VI.

Following [5], [9], we represent the statistical CSI by the spatial covariance matrix

$$\boldsymbol{R}_j = \frac{1}{T}\sum_{t=1}^{T} \boldsymbol{h}_{t,j}^{\mathrm{H}} \boldsymbol{h}_{t,j}, \tag{4}$$

where $\boldsymbol{h}_{j,t} \in \mathbb{C}^{1\times N}$ is the channel of the $j$-th UE ($j$-th row of $\boldsymbol{H}$) at TTI $t$ and $T$ is the number of TTIs considered to average (and hence approximate) the channel covariance matrix. Then, we can decompose

$$\boldsymbol{R}_j = \boldsymbol{U}_j \boldsymbol{\Lambda}_j \boldsymbol{U}_j^{\mathrm{H}}, \tag{5}$$

where $\boldsymbol{U}_j \in \mathbb{C}^{N\times N}$ and $\boldsymbol{\Lambda}_j \in \mathbb{R}^{N\times N}$ contain the eigenvectors and eigenvalues of $\boldsymbol{R}_j$, respectively.

In the sequel, we describe how clustering is performed in our work using the K-means algorithm, which is a well-known iterative algorithm that splits the $J$ UEs into $C$ clusters. Other clustering algorithms could have been used, such as agglomerative clustering in [1]. Like other clustering algorithms, K-means requires beforehand the number $C$ of clusters to be created [13]. In this work, the dominant column eigenvector $\boldsymbol{u}_{j,1}$ of each UE $j$ is used as input by the K-means algorithm [5] to cluster the UEs. The centroid of each cluster is updated using the mean of the dominant eigenvectors of the UEs currently belonging to that cluster. Then, the association of UEs to the clusters is updated by assigning each UE to the cluster whose centroid is closest (in Euclidean distance terms) to its dominant eigenvector $\boldsymbol{u}_{j,1}$. This process repeats until there are no more significant changes in the clusters' centroids. Due to space limitations, we refer the reader to [5] for more detailed information about K-means.

## V. Design of Hybrid Precoder

For hybrid beamforming, the analog part is, in general, implemented by phase shifters. After clustering, when selecting a UE $k$ from a cluster $c$, its analog precoder $\boldsymbol{w}_{\mathrm{RF},k}$ is built using only the phases of the components of the dominant eigenvector $\boldsymbol{u}_{k,1}$, which we denote by $\boldsymbol{w}_{\mathrm{RF},k} = \frac{1}{\sqrt{N}} e^{j\angle \boldsymbol{u}_{k,1}}$. Consequently, the analog precoder for the $K$ selected UEs is given as

$$\boldsymbol{W}_{\mathrm{RF}} = \frac{1}{\sqrt{N}} \begin{bmatrix} e^{j\angle \boldsymbol{u}_{1,1}} & e^{j\angle \boldsymbol{u}_{2,1}} & \dots & e^{j\angle \boldsymbol{u}_{K,1}} \end{bmatrix}. \tag{6}$$

To suppress the residual MU interference, we use the Zero-Forcing (ZF) for digital precoder $\boldsymbol{W}_{\mathrm{BB}}$, which is given by [14]

$$\boldsymbol{W}_{\mathrm{BB}} = \frac{\boldsymbol{H}_{\mathrm{eq}}^{H}(\boldsymbol{H}_{\mathrm{eq}}\boldsymbol{H}_{\mathrm{eq}}^{H})^{-1}}{\|\boldsymbol{H}_{\mathrm{eq}}^{H}(\boldsymbol{H}_{\mathrm{eq}}\boldsymbol{H}_{\mathrm{eq}}^{H})^{-1}\|_{\mathrm{F}}}. \tag{7}$$

In Section VI, we are going to see that we can exploit the ZF precoder characteristics to reduce the problem complexity.

## VI. Proposed Scheduling

In this section, we describe the search space reduction through clustering, digital precoder, and virtual agents. Afterward, we present the modeling of the scheduler priority. Finally, we show the proposed scheduling based on CB.

### A. Search Space Reduction

In the sequel, we describe how the ZF precoder can reduce the interference among clusters. As stated in Section V, ZF is used as the digital precoder herein, thus each UE signal is sent in the joint null space of the other UEs signals so that the served UE effective channel gain is tightly related to the channel correlation among UEs [14]. Since UEs belonging to different clusters are supposed to be low-correlated, interference among UEs' from different clusters becomes negligible [9].

Besides, this assumption drastically reduces the number of possible compositions of groups of UEs to be served on a given TTI, as discussed in the sequel. Let the BS be the learning agent (physical) that schedules $K$ UEs as to achieve a reward $d$, which is defined herein as the system data rate achieved by the $K$ scheduled UEs. Thus, there are

$$A = \binom{J}{K}. \tag{8}$$

possible actions, making the action set become impractical due to its combinatorial increase in $J$ and $K$.

Using the previous assumption of negligible interference among UEs of distinct clusters, each cluster can be seen as a virtual agent (logical) that selects UEs belonging to it to compose the group of UEs scheduled by the BS. Each virtual agent $c$ has its own set of actions $\mathcal{A}_c$ (with $A_c = |\mathcal{A}_c|$ actions), which is given by

$$A_c = \binom{J_c}{K_c}, \tag{9}$$

where $J_c$ and $K_c$ are the total number of UEs and the number of scheduled UEs of cluster $c$, respectively. Since usually $\sum_{c}^{C} A_c \ll A$, the search space is strongly reduced. Also, each virtual cluster has its action values stored in $\boldsymbol{d}_c \in \mathbb{R}_{+}^{A_c\times 1}$. The real action value of an action is defined as the mean received reward when that action is selected. By the law of large numbers, if the number of selected actions goes to infinity, the action value converges to the optimal one. This way, using the incremental average updating method we can define the action values $\boldsymbol{d}_c$ as [4]

$$\boldsymbol{d}_c(a_c) = \boldsymbol{d}_c(a_c) + \frac{1}{\boldsymbol{n}_c(a_c)}(d - \boldsymbol{d}_c(a_c)), \tag{10}$$

where $a_c$ is a given action and $\boldsymbol{n}_c \in \mathbb{Z}_{+}^{A_c\times 1}$ is the vector containing the number of times that each action was selected.

### B. Modeling the UEs Priority

In the following, we describe how UEs priority is modeled. We also consider that a UE is satisfied when it achieves a throughput target. We aim to use a function capable of mapping three behaviors. Almost no priority after the UE exceeds its target (step function), UE priority decreases rapidly when its throughput approaches or exceeds its target (sigmoidal function), and UEs have almost the same priority (flat line). In order to cope with UEs QoS requirements, we propose to

use, as in [15], a sigmoidal function to model UEs' priority as

$$P_j(v_j) = \frac{1}{1 + e^{-\delta(v_j - v_j^{\text{req}})}}, \tag{11}$$

where $\delta < 0$ controls the logistic function shape, $v_j^{\text{req}}$ and $v_j$ are the required throughput and the throughput of the $j$-th UE', respectively. $P_j(v_j)$ is a decreasing function of the UEs throughput with controllable shape and centered at $v_j^{\text{req}}$, as shown in Figure 1. We normalize both $v_j$ and $v_j^{\text{req}}$ to map the throughput of $v_j$ as a portion of its throughput requirement. As can be seen therein, depending on the $P(\cdot)$ shape and current throughput, an unsatisfied UE ($v_j < v_j^{req}$) can have a priority to be scheduled between 0.5 and 1 while a satisfied UE can have a priority between 0 and 0.5.

The shape of $P(\cdot)$ is associated with system satisfaction, where [15] obtained good results for $\delta = -9.1912$. In this work, we consider the system satisfaction as the ratio between the number of satisfied UEs and the total number of UEs. Following a similar approach as therein, we created 21 shapes for $P(\cdot)$ based on this basis value of $\delta$, which are shown in Figure 1, $s \in \{-10, -9, \ldots, 9, 10\}$ and $\delta = -9.1912 \times 2^s$ were used to defined $P(\cdot)$ shapes.

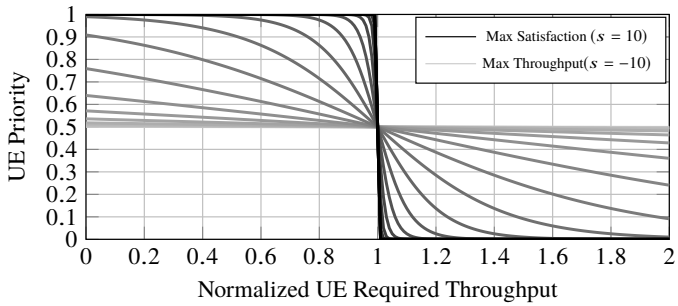


Figure 1. UE prioritization function.

### C. Scheduling Based on Contextual Bandits (SBCB)

In the sequel, we describe how the context (side information) is used by our scheduler, and we explain the scheduler itself. The SBCB proposed here aims to maximize the system throughput while satisfying QoS constraints. We consider as context the outdated CSI used to obtain the priority of each UE (11). Then, the mean priority $q_c(n)$ of UEs in the action $n$ belonging to cluster $c$ is given as

$$q_c(n) = \frac{1}{K_c} \sum_{j \in \mathcal{A}_c(n)} P(v_j), \tag{12}$$

where $\mathcal{A}_c(n)$ is the UEs in the action $n$ and cluster $c$. Therefore, we can define $\boldsymbol{q}_c \in \mathbb{R}_+^{A_c \times 1}$ as mean UEs priority vector of each action in a cluster $c$.

At each TTI, for each virtual agent, our SBCB selects a random action from $\mathcal{A}_c$ with probability $\epsilon$ (exploration phase) or takes the action that maximizes the trade-off between the UEs throughput and their priorities with probability $1 - \epsilon$ (exploitation phase). Note that, the exploitation phase corresponds to take the action of the maximum value in the Hadamard (element-wise) product $\boldsymbol{q}_c \odot \boldsymbol{d}_c$. In our work, we are using the $\epsilon$-decaying method to calculate the $\epsilon$, where the $\epsilon$ value decays over time until it reaches the desired value, we refer the reader to [16] for more details of $\epsilon$-greedy algorithm and $\epsilon$-decaying method. The UEs chosen by the virtual agents are then scheduled by the BS, which polls them using the analog precoder to obtain as CSI their equivalent channels on top of which the digital precoder is applied. After that, the BS calculates the achieved data rate using (3). At this point, all virtual agents at the BS can store their action values according to (10). Lastly, if the smaller UE throughput is $\lambda$% greater than its target, the $s$ value is reduced in 1 to make $P(\cdot)$ in (11) approach a maximum throughput policy (flat line). Otherwise, increases the $s$ value in 1 to make the shape $P(\cdot)$ in (11) approach a maximum QoS provisioning policy (step function). Hence, each virtual agent learns over time its best groups of UEs to maximize system throughput, a knowledge that is combined with the context information on prioritizing certain UEs to improve QoS provisioning. The pseudo-code of the SBCB algorithm is presented in Algorithm 1.

**Algorithm 1** SBCB Algorithm.

Input: $\mathcal{A}_c$, $C$ and $T$
Initialize: $\boldsymbol{q}_c = \mathbf{0}_{A_c \times 1}$ and $\boldsymbol{d}_c = \mathbf{0}_{A_c \times 1}$, $\forall c$
Initialize: set of scheduled UEs $\mathcal{S} = \emptyset$
Initialize: counter vectors $\boldsymbol{n}_c = \mathbf{0}_{A_c \times 1}$ of each action $\forall c$
Initialize: shape control variable $s = 10$ ▷ Max. priority.
**for** Each TTI **do**
  **for** $c = 1$ to $C$ **do**
    Calculate the vector of weights $\boldsymbol{q}_c$ using (12)
    $a_c \leftarrow \begin{cases} \underbrace{\text{action that maximizes } (\boldsymbol{q}_c \odot \boldsymbol{d}_c)}_{\text{Exploitation}}, & \text{probability } 1 - \epsilon \\ \underbrace{\text{random action from } \mathcal{A}_c}_{\text{Exploration}}, & \text{probability } \epsilon \end{cases}$
    $\mathcal{S} \leftarrow \mathcal{S} \cup \mathcal{A}_c(a_c)$ ▷ Schedule the UEs.
  **end for**
  Scheduled UEs $\mathcal{S}$ feed back their CSI
  Compute hybrid (analog and digital) precoder using (6) and (7)
  $d \leftarrow$ sum of scheduled UEs data rate using (3) ▷ Reward.
  **for** $c = 1$ to $C$ **do**
    $\boldsymbol{n}_c(a_c) \leftarrow \boldsymbol{n}_c(a_c) + 1$ ▷ Number of times that $a_c$ was chosen.
    $\boldsymbol{d}_c(a_c) \leftarrow \boldsymbol{d}_c(a_c) + \frac{1}{\boldsymbol{n}_c(a_c)}(d - \boldsymbol{d}_c(a_c))$ ▷ Action values.
  **end for**
  **if** Smaller UE throughput $\geq$ $\lambda$% its throughput requirement **then**
    **if** $s \geq -10$ **then**
      $s = s - 1$ ▷ Prioritize more the throughput.
    **end if**
  **else**
    **if** $s \leq 10$ **then**
      $s = s + 1$ ▷ Prioritize more the satisfaction.
    **end if**
  **end if**
**end for**

The decisions that the BSs take at each TTI are: (1) schedule the UEs using the SBCB algorithm; (2) poll UEs for their CSI using analog precoders; and (3) compute digital precoders and send data to the scheduled UEs. In general, before step (1), most works in literature considers the CSI of all UEs available or they select a subset of the total UEs to poll for their CSI, from which the scheduled UEs are selected for data reception, such as [1], [5]–[8]. Thus, as our scheme only polls the already scheduled UEs for CSI, it demands less feed back

than most of those works. Furthermore, the proposed scheme avoids computing digital precoders for every possible group of UEs to be served.

## VII. Numerical Results

In this section, we compare the proposed SBCB algorithm with the QoS-aware Proportional Fair QoS (PF QoS) algorithm [17]. The PF QoS is similar to the traditional PF scheduling algorithm, however, it works with two sets of UEs: the priority set with UEs that do not meet their QoS requirements, which are prioritized and; the low priority set with the rest of the UEs (currently satisfied UEs). It is important to notice that the PF QoS algorithm also performs the clustering before scheduling and only selects $K_c$ UEs per clusters, which jointly with the hybrid precoder in Section V deals with the interference among clusters. Furthermore, the dominant eigenvalue and eigenvector of each UEs were used in PF QoS to estimate the instantaneous rate of the PF algorithm, since this is the same CSI employed by the SBCB and keeps comparisons between it and PF QoS fair.

The simulated scenario considers a MIMO BS equipped with an $8 \times 8$ UPA ($N = 64$). It services 20 UEs inside a 60° sector with 200 m of radius, which are uniformly distributed inside two hotspots with a radius of 15 m. The hotspots' centers are 100 m away from the BS and 30° apart.

UEs choose between 2 services, where the required throughput of service 2 is that of service 1 plus 200 kbps. Moreover, a satisfaction level of $\mu = 90\%$ is required in the system. We also consider the value of $\lambda = 120\%$ of the smaller UE throughput to trigger the change of shape conditions. The reason for chose this $\lambda$ value is that the SBCB can start to change the UEs' priority before they get unsatisfied.

We adopt Quasi Deterministic Radio Channel Generator (QuaDRiGa) Urban Micro (UMi) Line Of Sight (LOS) channel model [18], 3 km/h average UE speed, and assume the BS power to be evenly divided among 125 RBs. However, in this work, we are assuming only one RB available for transmission. The most relevant parameters used in our simulations are shown in Table I.

Table I
Simulation Parameters.

| Parameter | Value |
|---|---|
| System bandwidth | 100 MHz |
| System carrier frequency | 28 GHz |
| Number of subcarriers in an RB | 12 |
| Subcarrier spacing | 60 kHz |
| TTI duration | 0.25 ms |
| Number of OFDM symbols per TTI | 14 |
| Simulation duration | 1 s |
| Number of simulation rounds | 100 |
| Cell radius | 200 m |
| Total transmit power | 35 dBm |
| Noise figure | 9 dB |
| Noise spectral density | -174 dBm/Hz |
| Shadowing standard deviation | 3.1 dB |
| Number of UEs | 20 |
| Number of clusters | 2 |
| Number of UEs selected per clusters | 2 |

In the simulations, we consider a warm-up phase during which $\epsilon$ values vary linearly from 100% to 5% over time. A high exploration rate at the beginning provides better knowledge about the action space and avoids getting stuck at local optima. We also assume that UEs start with their throughput requirement fulfilled. This phase allows the system to get into a configuration that we consider typical of normal long-run conditions.

Three performance metrics are considered herein: the number of TTIs (iterations) to reach and stay above the target satisfaction, the system satisfaction level itself, and the system throughput. The system satisfaction was previously defined in Section VI-C. The system throughput is the sum of the throughput of all UEs.

In the sequel, we evaluate the system satisfaction over the TTIs for different required throughput's of service 1. We recall that the required throughput of service 2 is 200 kbps higher than that of service 1. As it is shown, the number of TTIs needed to reach the target satisfaction $\mu$ increases with the throughput requirement. For low (100 kbps), medium (400 kbps), and high (600 kbps) throughput requirements, 15, 30, and 70 TTIs are needed to achieve the target satisfaction, respectively. Therefore, the SBCB algorithm is capable of learning how to schedule the UEs aiming at achieving the target satisfaction in a small number of TTIs. Later, we will see that the SBCB algorithm also increases system throughput.

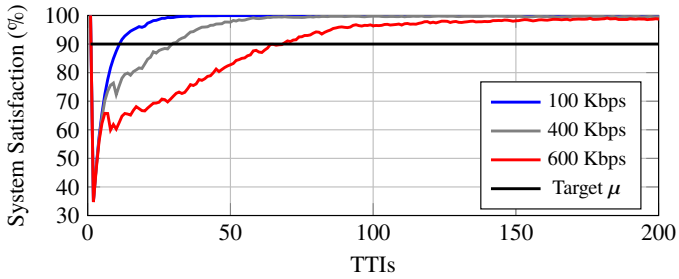


Figure 2. System satisfaction over the TTIs for different required throughput.

In Figure 3, the system satisfaction for SBCB and PF QoS algorithms are shown for increasing values of required throughput of service 1, where the bright region is the 90% confidence interval for the obtained results. As expected, satisfaction levels decrease when the required rate increases. However, differently from PF QoS, the SBCB algorithm is capable of maintaining higher satisfaction (above $\mu$) and closer to 100% even for high required throughput. The performance loss of the PF QoS is because it does not take into account any information about the interference among scheduled UEs, increasing the probability of scheduling in the same RB UEs with high channel correlation. SBCB learns about channel correlation through rewards and uses the same CSI as PF QoS.

In Figure 4, SBCB and PF QoS algorithms are compared in terms of system throughput for different values of the required throughput for service 1. As can be seen, the system throughput decreases as the required throughput increases, so that there is a trade-off between satisfaction and system throughput to which the algorithms are subjected. The SBCB solution

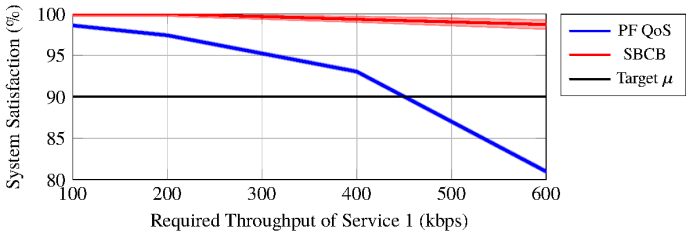


Figure 3. System satisfaction versus the required throughput of service 1.

prioritizes the QoS provisioning aiming at maintaining high satisfaction. Therefore, it loses more performance on system throughput than on satisfaction. The baseline algorithm can maintain almost the same throughput. However,its satisfaction decreases drastically compared to SBCB. Anyway, SBCB provides a gain in system throughput up to 22.5% compared to PF QoS.

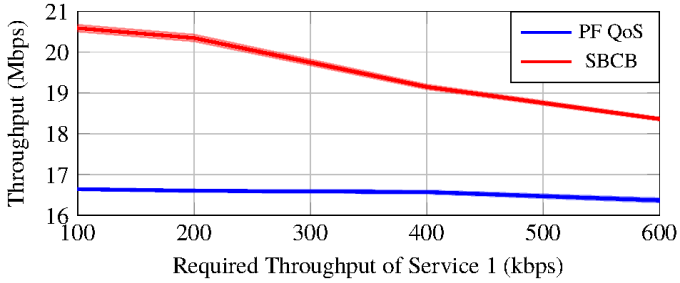


Figure 4. System throughput versus required throughput of service 1.

## VIII. Conclusions

In this work, we evaluated a novel scheduling based on CB aiming to maximize the system throughput with QoS constraints in a multiple service hybrid precoding massive MIMO scenario. The proposed solution avoids computing digital precoders for every group of scheduled UEs, reduces the required CSI feed back, and keep information of the bests groups of scheduled UEs up-to-date over time. We show that the low correlation among clusters makes it possible to create virtual agents, drastically reducing the search space for UE scheduling, polling, and digital precoding. Numerical results show that the proposed solution outperforms the baseline algorithm in terms of system throughput and UE satisfaction. Furthermore, the proposed solution converges rapidly and achieves the target satisfaction in a small number of TTIs. The study of other CB strategies and consideration of multiples RBs are the perspective of this work.

## IX. Acknowledgments

The authors thank FUNCAP, CAPES and CNPq for their financial and scholarship support. This work is supported by CAPES/PROBRAL Proc. nº 88887.144009/2017-00 and CAPES/PRINT Proc. nº 88887.311965/2018-00. It was also supported by DAAD with funds from the Federal Ministry of Education and Research (BMBF) and performed in the context of the DFG Collaborative Research Center (CRC) 1053 MAKI - subprojects C1 and B3.

## References

[1] X. Sun, X. Gao, G. Y. Li, and W. Han, "Agglomerative User Clustering and Cluster Scheduling for FDD Massive MIMO Systems," *IEEE Access*, vol. 7, pp. 86 522–86 533, 2019.

[2] R. W. Heath, N. González-Prelcic, S. Rangan, W. Roh, and A. M. Sayeed, "An Overview of Signal Processing Techniques for Millimeter Wave MIMO Systems," *IEEE Journal of Selected Topics in Signal Processing*, vol. 10, no. 3, pp. 436–453, April 2016.

[3] S. Maghsudi and E. Hossain, "Multi-Armed Bandits with Application to 5G Small Cells," *IEEE Wireless Communications*, vol. 23, no. 3, pp. 64–73, June 2016.

[4] R. S. Sutton and A. G. Barto, *Introduction to Reinforcement Learning*, 1st ed. Cambridge, MA, USA: MIT Press, 1998.

[5] J. Nam, A. Adhikary, J.-Y. Ahn, and G. Caire, "Joint Spatial Division and Multiplexing: Opportunistic Beamforming, User Grouping and Simplified Downlink Scheduling," *IEEE Journal of Selected Topics in Signal Processing*, vol. 8, no. 5, pp. 876–890, Oct. 2014.

[6] G. Bu and J. Jiang, "Reinforcement Learning-Based User Scheduling and Resource Allocation for Massive MU-MIMO System," in *2019 IEEE/CIC International Conference on Communications in China (ICCC)*, Aug 2019, pp. 641–646.

[7] Y. Yang, Y. Li, K. Li, S. Zhao, R. Chen, J. Wang, and S. Ci, "DECCO: Deep-Learning Enabled Coverage and Capacity Optimization for Massive MIMO Systems," *IEEE Access*, vol. 6, pp. 23 361–23 371, 2018.

[8] A. Maatouk, S. E. Hajri, M. Assaad, H. Sari, and S. Sezginer, "Graph Theory Based Approach to Users Grouping and Downlink Scheduling in FDD Massive MIMO," in *2018 IEEE International Conference on Communications (ICC)*, May 2018, pp. 1–7.

[9] W. V. F. Mauricio, D. C. Araujo, F. H. C. Neto, F. R. M. Lima, and T. F. Maciel, "A Low Complexity Solution for Resource Allocation and SDMA Grouping in Massive MIMO Systems," in *2018 15th International Symposium on Wireless Communication Systems (ISWCS)*, Aug 2018, pp. 1–6.

[10] H. Xu, T. Zhao, S. Zhu, D. Lv, and J. Zhao, "Agglomerative Group Scheduling for MmWave Massive MIMO under Hybrid Beamforming Architecture," in *2018 IEEE 18th International Conference on Communication Technology (ICCT)*, Oct 2018, pp. 347–351.

[11] D. C. Araújo, E. Karipidis, A. L. F. de Almeida, and J. C. M. Mota, "Hybrid Beamforming Design with Finite-resolution Phase-shifters for Frequency Selective Massive MIMO Channels," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, March 2017, pp. 6498–6502.

[12] V. Kumar and N. B. Mehta, "Modeling and Analysis of Differential CQI Feedback in 4G/5G OFDM Cellular Systems," *IEEE Transactions on Wireless Communications*, vol. 18, no. 4, pp. 2361–2373, April 2019.

[13] G. W. Milligan and M. C. Cooper, "An Examination of Procedures for Determining the Number of Clusters in a Data Set," *Psychometrika*, vol. 50, no. 2, pp. 159–179, Jun 1985. [Online]. Available: https://doi.org/10.1007/BF02294245

[14] E. Castaneda, A. Silva, A. Gameiro, and M. Kountouris, "An Overview on Resource Allocation Techniques for Multi-User MIMO Systems," *IEEE Communications Surveys and Tutorials*, vol. 19, no. 1, pp. 239–284, 2017.

[15] R. P. Antonioli, E. B. Rodrigues, T. F. Maciel, D. A. Sousa, and F. R. P. Cavalcanti, "Adaptive Resource Allocation Framework for User Satisfaction Maximization in Multi-Service Wireless Networks," *Telecommunication Systems*, vol. 68, no. 2, pp. 259–275, Jun 2018. [Online]. Available: https://doi.org/10.1007/s11235-017-0391-3

[16] O. Caelen and G. Bontempi, "Improving the Exploration Strategy in Bandit Algorithms," in *Learning and Intelligent Optimization*, V. Maniezzo, R. Battiti, and J.-P. Watson, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 56–68.

[17] 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," *IEEE Communications Surveys Tutorials*, vol. 15, no. 2, pp. 678–700, Second 2013.

[18] S. Jaeckel, L. Raschkowski, K. Börner, and L. Thiele, "Quadriga: A 3-D Multi-cell Channel Model with Time Evolution for Enabling Virtual Field Trials," *IEEE Transactions on Antennas and Propagation*, vol. 62, no. 6, pp. 3242–3256, June 2014.