8-2019

# Reinforcement Learning for Self Organization and Power Control of Two-Tier Heterogeneous Networks

Roohollah Amiri
*Boise State University*

Mojtaba Ahmadi Almasi
*Boise State University*

Jeffrey G. Andrews
*University of Texas at Austin*

Hani Mehrpouyan
*Boise State University*, hanimehrpouyan@boisestate.edu

# Reinforcement Learning for Self Organization and Power Control of Two-Tier Heterogeneous Networks

Roohollah Amiri, *Student Member, IEEE,* Mojtaba Ahmadi Almasi, *Student Member, IEEE,* Jeffrey G. Andrews, *Fellow, IEEE,*

Hani Mehrpouyan, *Member, IEEE.*

## Abstract

Self-organizing networks (SONs) can help manage the severe interference in dense heterogeneous networks (HetNets). Given their need to automatically configure power and other settings, machine learning is a promising tool for data-driven decision making in SONs. In this paper, a HetNet is modeled as a dense two-tier network with conventional macrocells overlaid with denser small cells (e.g. femto or pico cells). First, a distributed framework based on multi-agent Markov decision process is proposed that models the power optimization problem in the network. Second, we present a systematic approach for designing a reward function based on the optimization problem. Third, we introduce Q-*learning* based distributed power allocation algorithm (Q-DPA) as a self-organizing mechanism that enables ongoing transmit power adaptation as new small cells are added to the network. Further, the sample complexity of the Q-DPA algorithm to achieve $\epsilon$-optimality with high probability is provided. We demonstrate, at density of several thousands femtocells per km$^2$, the required quality of service of a macrocell user can be maintained via the proper selection of independent or cooperative learning and appropriate Markov state models.

## Index Terms

Self-organizing networks, HetNets, Reinforcement learning, Markov decision process.

## I. INTRODUCTION

Self-organization is a key feature as cellular networks densify and become more heterogeneous, through the additional small cells such as pico and femtocells [2]–[6]. Self-organizing networks (SONs) can perform self-configuration, self-optimization and self-healing. These operations can cover basic tasks such as configuration of a newly installed base station (BS), resource management, and fault management in the network [7]. In other words, SONs attempt to minimize human intervention where they use measurements from the network to minimize the cost of installation, configuration and maintenance of the network. In fact SONs bring two main factors in play: *intelligence* and *autonomous adaptability* [2], [3]. Therefore, machine learning techniques can play a major role in processing underutilized sensory data to enhance the performance of SONs [8], [9].

One of the main responsibilities of SONs is to configure the transmit power at various small BSs to manage interference. In fact, a small BS needs to configure its transmit power before joining the network (as self-configuration). Subsequently, it needs to dynamically control its transmit power during its operation in the network (as self-optimization). To address these two issues, we consider a macrocell network overlaid with small cells and focus on autonomous distributed power control, which is a key element of self-organization since it improves network throughput [10]–[14] and minimizes energy usage [15]–[17]. We rely on local measurements, such as signal-to-interference-plus-noise ratio (SINR), and the use of machine learning to develop a SON framework that can continually improve the above performance metrics.

### A. Related Work

In wireless communications, dynamic power control with the use of machine learning has been implemented via reinforcement learning (RL). In this context, RL is an area of machine learning that attempts to optimize a BS's transmit power to achieve a certain goal such as throughput maximization. One of the main advantages of RL with respect to supervised learning methods is its training phase, in which there is no need for correct input/output data. In fact, RL operates by applying the experience that it has gained through interacting with the network [18]. RL methods have been applied in the field of wireless communications in areas such as resource management [19]–[24], energy harvesting [25], and opportunistic spectrum access [26], [27]. A comprehensive review of RL applications in wireless communications can be found in [28].

Q-*learning* is a model-free RL method [29]. The model-free feature of Q-*learning* makes it a proper method for scenarios in which the statistics of the network continuously change. Further, Q-*learning* has low computational complexity and can be implemented by BSs in a distributed manner [1]. Therefore, Q-*learning* can bring *scalability*, *robustness*, and *computational efficiency* to large networks. However, designing a proper reward function which accelerates the learning process and avoids false learning or unlearning phenomena [30] is not trivial. Therefore, to solve an optimization problem, an appropriate reward function for Q-*learning* needs to be determined.

In this regard, the works in [19]–[24] have proposed different reward functions to optimize power allocation between femtocell base stations (FBSs). The method in [19] uses independent Q-*learning* in a cognitive radio system to set the transmit power of secondary BSs in a digital television system. The solution in [19] ensures that the minimum quality of service (QoS) for the primary user is met by applying Q-*learning* and using the SINR as a metric. However, the approach in [19] doesn't take the QoS of the secondary users into considerations. The work in [20] uses cooperative Q-*learning* to maximize the sum transmission rate of the femtocell users while keeping the transmission rate of macrocell users near a certain threshold. Further, the authors in [21] have used the proximity of FBSs to a macrocell user as a factor in the reward function. This results in a fair power allocation scheme in the network. Their proposed reward function keeps the transmission rate of the macrocell user above a certain threshold while maximizing the sum transmission rate of FBSs. However, by not considering a minimum threshold for the FBSs' rates, the approach in [21] fails to support some FBSs as the density of the network (and consequently interference) increases. The authors in [22] model the cross-tier interference management problem as a non-cooperative game between femtocells and the macrocell. In [22], femtocells use the average SINR measurement to enhance their individual performances while maintaining the QoS of the macrocell user. In [23], the authors attempt to improve the transmission rate of cell-edge users while keeping the fairness between the macrocell and the femtocell users by applying a round robin approach. The work in [24] minimizes power usage in a Long Term Evolution (LTE) enterprise femtocell network by applying an exponential reward function without the requirement to achieve fairness amongst the femtocells in the network.

In the above works, the reward functions do not apply to dense networks. That is to say, first, there is no minimum threshold for the achievable rate of the femtocells. Second, the reward

functions are designed to limit the macrocell user rate to its required QoS and not more than that. This property encourages an FBS to use more power to increase its own rate by assuming that the caused interference just affects the macrocell user. However, the neighbor femtocells suffer from this decision and overall the sum rate of the network decreases. Further, they do not provide a generalized framework for modeling a HetNet as a multi-agent RL network or a procedure to design a reward function which meets the QoS requirements of the network. In this paper, we focus on dense networks and try to provide a general solution to the above challenges.

### B. Contributions

We propose a learning framework based on multi-agent Markov decision process (MDP). By considering an FBS as an agent, the proposed framework enables FBSs to join and adapt to a dense network autonomously. Due to unplanned and dense deployment of femtocells, providing the required QoS to all the users in the network becomes an important issue. Therefore, we design a reward function that trains the FBSs to achieve this goal. Furthermore, we introduce a Q-*learning* based distributed power allocation approach (Q-DPA) as an application of the proposed framework. Q-DPA uses the proposed reward function to maximize the transmission rate of femtocells while prioritizing the QoS of the macrocell user. More specifically the contributions of the paper can be summarized as:

1) We propose a framework that is agnostic to the choice of learning method but also connects the required RL analogies to wireless communications. The proposed framework models a multi-agent network with a single MDP that contains the joint action of the all the agents as its action set. Next, we introduce MDP factorization methods to provide a distributed and scalable architecture for the proposed framework. The proposed framework is used to benchmark the performance of different learning rates, Markov state models, or reward functions in two-tier wireless networks.

2) We present a systematic approach for designing a reward function based on the optimization problem and the nature of RL. In fact, due to scarcity of resources in a dense network, we propose some properties for a reward function to maximize sum transmission rate of the network while considering minimum requirements of all users. The procedure is simple and general and the designed reward function is in the shape of low complexity polynomials. Further, the designed reward function results in increasing the achievable sum transmission

rate of the network while consuming considerably less power compared to greedy based algorithms.

3) We propose Q-DPA as an application of the proposed framework to perform distributed power allocation in a dense femtocell network. Q-DPA uses the factorization method to derive independent and cooperative learning from the optimal solution. Q-DPA uses local signal measurements at the femtocells to train the FBSs in order to: (i) maximize the transmission rate of femtocells, (ii) achieve minimum required QoS for all femtocell users with a high probability, and (iii) maintain the QoS of macrocell user in a densely deployed femtocell network. In addition, we determine the minimum number of samples that is required to achieve an $\epsilon$-optimal policy in Q-DPA as its sample complexity.

4) We introduce four different learning configurations based on different combinations of independent/cooperative learning and Markov state models. We conduct extensive simulations to quantify the effect of different learning configurations on the performance of the network. Simulations show that the proposed Q-DPA algorithm can decrease power usage and as a result reduce the interference to the macrocell user.

The paper is organized as follows. In Section II, the system model is presented. Section III introduces the optimization problem and presents the existing challenges in solving this problem. Section IV presents the proposed learning framework which models a two-tier femtocell network with a multi-agent MDP. Section V-A presents the Q-DPA algorithm as an application of the proposed framework. Section VI presents the simulation results while Section VII concludes the paper.

*Notation:* Lower case, boldface lower case, and calligraphic symbols represent scalars, vectors, and sets, respectively. For a real-valued function $Q : \mathcal{Z} \to \mathbb{R}$, $\|Q\|$ denotes the max norm, i.e., $\|Q\| = \max_{z \in \mathcal{Z}} |Q(z)|$. $\mathbb{E}_x[\cdot]$, $\mathbb{E}_x[\cdot|\cdot]$, and $\frac{\partial f}{\partial x}$ denote the expectation, the conditional expectation, and the partial derivation with respect to $x$, respectively. Further, $\Pr(\cdot|\cdot)$ and $|\cdot|$ denote the conditional probability and absolute value operators, respectively.

## II. DOWNLINK SYSTEM MODEL

Consider the downlink of a single cell of a HetNet operating over a set $\mathcal{S} = \{1, ..., S\}$ of $S$ orthogonal subbands. In the cell a single macro base station (MBS) is deployed. The MBS serves one macrocell user equipment (MUE) over each subband while guaranteeing this user a minimum
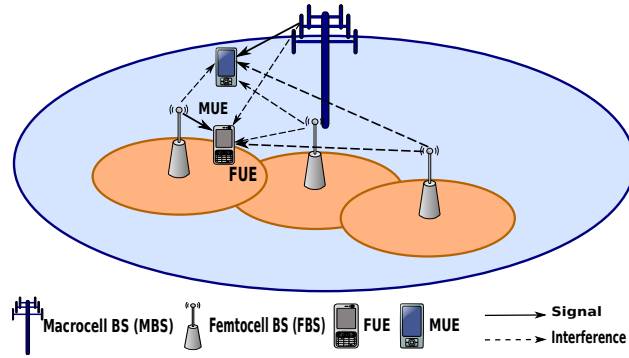
Figure 1. Macrocell and femtocells operating over the same frequency band.

average SINR over each subband which is denoted by $\Gamma_0$. A set of FBSs are deployed in area of coverage of the macrocell. Each FBS selects a random subband and serves one femtocell user equipment (FUE). We assume that overall, on each subband $s \in \mathcal{S}$, a set $\mathcal{K} = \{1, ..., K\}$ of $K$ FBSs are operating. Each FBS guarantees a minimum average SINR denoted by $\Gamma_k$ to its related FUE. We consider a dense network in which the density results in both cross-tier and co-tier interference. Therefore, in order to control the interference-level and provide the users with their required minimum SINR, we focus on power allocation in the downlink of the femtocell network. Uplink results can be obtained in a similar fashion but are not included for brevity. The overall network configuration is presented in Fig. 1. We focus on one subband, meanwhile the proposed solution can be extended to a case in which each FBS supports multiple users on different subbands.

We denote the MBS-MUE pair by the index $0$ and the FBS-FUE pairs by the index $k$ from the set $\mathcal{K}$. In the downlink, the received signal at the MUE operating over subband $s$ includes interference from the femtocells and thermal noise. Hence, the SINR at the MUE operating over subband $s \in \mathcal{S}$, $\gamma_0$, is calculated as

$$\gamma_0 = \frac{p_0|h_{0,0}|^2}{\underbrace{\sum_{k\in\mathcal{K}} p_k|h_{k,0}|^2}_{\text{femtocells' interference}} + N_0}, \tag{1}$$

where $p_0$ denotes the power transmitted by the MBS and $h_{0,0}$ denotes the channel gain from the MBS to the MUE. Further, the power transmitted by the $k$th FBS is denoted by $p_k$ and the channel gain from the $k$th FBS to the MUE is denoted by $h_{k,0}$. Finally, $N_0$ denotes the variance of the additive white Gaussian noise. Similarly, the SINR at the $k$th FUE operating over subband

$s \in \mathcal{S}$, $\gamma_k$, is obtained as

$$\gamma_k = \frac{p_k|h_{k,k}|^2}{\underbrace{p_0|h_{0,k}|^2}_{\text{macrocell's interference}} + \underbrace{\sum_{j \in \mathcal{K} \backslash \{k\}} p_j|h_{j,k}|^2}_{\text{femtocells' interference}} + N_k}, \tag{2}$$

where $h_{k,k}$ denotes the channel gain between the $k$th FBS and the $k$th FUE, $h_{0,k}$ denotes the channel gain between the MBS and the $k$th FUE, $p_j$ denotes the transmit power of the $j$th FBS, $h_{j,k}$ is the channel gain between the $j$th FBS and the $k$th FUE, and $N_k$ is the variance of the additive white Gaussian noise. Finally, the transmission rates, normalized by the transmission bandwidth, at the MUE and the FUE operating over subband $s \in \mathcal{S}$, i.e., $r_0$ and $r_k$, respectively, are expressed as $r_0 = \log_2(1 + \gamma_0)$ and $r_k = \log_2(1 + \gamma_k)$, $k \in \mathcal{K}$.

## III. PROBLEM FORMULATION

Each FBS has the objective of maximizing its transmission rate while ensuring that the SINR of the MUE is above the required threshold, i.e., $\Gamma_0$. Denoting $\mathbf{p} = \{p_1, ..., p_K\}$ as the vector of the transmit powers of the $K$ FBSs operating over the subband $s \in \mathcal{S}$, the power allocation problem is presented as follow

$$\underset{\mathbf{p}}{\text{maximize}} \quad \sum_{k \in \mathcal{K}} \log_2(1 + \gamma_k)$$

$$\text{subject to} \quad 0 \leq p_k \leq p_{max}, \ k \in \mathcal{K}, \tag{3b}$$

$$\gamma_0 \geq \Gamma_0, \tag{3c}$$

$$\gamma_k \geq \Gamma_k, \ k \in \mathcal{K}. \tag{3d}$$

where $p_{max}$ defines the maximum available transmit power at each FBS. The objective (3a) is to maximize the sum transmission rate of the FUEs. Constraint (3b) refers to the power limitation of every FBS. Constraints (3c) and (3d) ensure that the minimum SINR requirement is satisfied for the MUE and the FUEs. The addition of constraint (3d) to the optimization problem is one of the differences between the proposed approach in this paper and that of [19]–[24].

Considering (2), it can be concluded that the optimization in (3) is a non-convex problem for dense networks. This follows from the SINR expression in (2) and the objective function (3). More specifically, the interference term due to the neighboring femtocells in the denominator of (2) ensures that the optimization problem in (3) is not convex [31]. This interference term

may be ignored in low density networks but cannot be ignored in dense networks consisting of a large number of femtocells [32]. However, non-convextiy is not the only challenge of the above problem. In fact, many iterative algorithms are developed to solve the above optimization problem with excellent performance. However, their algorithms contains expensive computations such as matrix inversion and bisection or singular value decomposition in each iteration which makes their real-time implementation challenging [33]. Besides, the $k$th FBS is only aware of its own transmit power, $p_k$, and does not know the transmit powers of the remaining FBSs. Therefore, the idea here is to treat the given problem as a black-box and try to learn the relation between the transmit power and the resulting transmission rate gradually by interacting with the network and simple computations.

To realize self-organization, each FBS should be able to operate autonomously. This means an FBS should be able to connect to the network at anytime and to continuously adapt its transmit power to achieve its objectives. Therefore, our optimization problem requires a self-adaptive solution. The steps for achieving self-adaptation can be summarized as: (i) the FBS measures the interference level at its related FUEs, (ii) determines the maximum transmit power to support its FUEs while not greatly degrading the performance of other users in the network. In the next section, the required framework to solve this problem will be presented.

## IV. THE PROPOSED LEARNING FRAMEWORK

Here, first we model a multi-agent network as an MDP. Then the required definitions, evaluation methods, and factorization of the MDP to develop a distributed learning framework are explained. Subsequently, the femtocell network is modeled as a multi-agent MDP and the proposed learning framework is developed.

### A. Multi-Agent MDP and Policy Evaluation

A single-agent MDP comprises an agent, an environment, an action set, and a state set. The agent can transition between different states by choosing different actions. The trace of actions that is taken by the agent is called its policy. With each transition, the agent will receive a reward from the environment, as a consequence of its action, and will save the discounted summation of rewards as a cumulative reward. The agent will continue its behavior with the goal of maximizing the cumulative reward and the value of cumulative reward evaluates the

chosen policy. The discount property increases the impact of recent rewards and decreases the effect of later ones. If the number of transitions is limited, the non-discounted summation of rewards can be used as well.

A multi-agent MDP consists of a set, $\mathcal{K}$, of $K$ agents. The agents select actions to move between different states of the model to maximize the cumulative reward received by all the agents. Here, we again formulate the network of agents as one MDP, e.g., we define the action set as the joint action set of all the agents. Therefore, the multi-agent MDP framework is defined with a tuple as $(\mathcal{A}, \mathcal{X}, Pr, \mathbf{R})$ with the following definitions.

- $\mathcal{A}$ is the joint set of all the agents' actions. An agent $k$ selects its action $a$ from its action set $\mathcal{A}_k$, i.e., $a_k \in \mathcal{A}_k$. The joint action set is represented as $\mathcal{A} = \mathcal{A}_1 \times \cdots \times \mathcal{A}_K$, with $\mathbf{a} \in \mathcal{A}$ as a single joint action.

- The state of the system is defined with a set of random variables. Each random variable is represented by $X_i$ with $i = 1, ..., n$, and the state set is represented as $\mathcal{X} = \{X_1, X_2, ..., X_n\}$, where $\mathbf{x} \in \mathcal{X}$ denotes a single state of the system. Each random variable reflects a specific feature of the network.

- The transition probability function, $\Pr(\mathbf{x}, \mathbf{a}, \mathbf{x}')$, represents the probability of taking joint action $\mathbf{a}$ at state $\mathbf{x}$ and ending in state $\mathbf{x}'$. In other words, the transition probability function defines the environment which agents are interacting with.

- $\mathbf{R}(\mathbf{x}, \mathbf{a})$ is the reward function such that its value is the received reward by the agents for taking joint action $\mathbf{a}$ at state $\mathbf{x}$.

We define $\pi : \mathcal{X} \to \mathbf{A}$ as the policy function, where $\pi(\mathbf{x})$ is the joint action that is taken at the state $\mathbf{x}$. In order to evaluate the policy $\pi(\mathbf{x})$, a value function $V_\pi(\mathbf{x})$ and an action-value function $\mathbf{Q}_\pi(\mathbf{x}, \mathbf{a})$ are defined. The value of the policy $\pi$ in state $\mathbf{x}' \in \mathcal{X}$ is defined as [18]

$$V_\pi(\mathbf{x}') = \mathbb{E}_\pi \left[ \sum_{t=0}^{\infty} \beta^t \mathbf{R}^{(t+1)} \big| \mathbf{x}^{(0)} = \mathbf{x}' \right], \tag{4}$$

in which $\beta \in (0, 1]$ is a discount factor, $\mathbf{R}^{(t+1)}$ is the received reward at time step $t+1$, and $\mathbf{x}^{(0)}$ is the initial state. The action-value function, $\mathbf{Q}_\pi(\mathbf{x}, \mathbf{a})$, represents the value of the policy $\pi$ for taking joint action $\mathbf{a}$ at state $\mathbf{x}$ and then following policy $\pi$ for subsequent iterations. According to [18], the relation between the value function and the action-value function is given by

$$\mathbf{Q}_\pi(\mathbf{x}, \mathbf{a}) = \mathbf{R}(\mathbf{x}, \mathbf{a}) + \beta \sum_{\mathbf{x}' \in \mathcal{X}} \Pr(\mathbf{x}'|\mathbf{x}, \mathbf{a}) V_\pi(\mathbf{x}'). \tag{5}$$

For the ease of notation, we will use $V$ and $\mathbf{Q}$ for the value function and the action-value function of policy $\pi$, respectively. Further, we use the term *Q-function* to refer to the action-value function. The optimal value of state $\mathbf{x}$ is the maximum value that can be reached by following any policy and starting at this state. An optimal value function $V^*$, which gives an optimal policy $\pi^*$, satisfies the Bellman optimality equation as [18]

$$V^*\left(\mathbf{x}\right) = \max_{\mathbf{a}} \mathbf{Q}^*\left(\mathbf{x}, \mathbf{a}\right), \tag{6}$$

where $\mathbf{Q}^*\left(\mathbf{x}, \mathbf{a}\right)$ is an optimal Q-function under policy $\pi^*$. The general solution for (6) is to start from an arbitrary policy and using the *generalized policy iteration* (GPI) [18] method to iteratively evaluate and improve the chosen policy to achieve an optimal policy. If the agents have *a priori* information of the environment, i.e., $Pr\left(\mathbf{x}, \mathbf{a}, \mathbf{x}'\right)$ is known to the agents, dynamic programming is the solution for (6). However, the environment is unknown in most practical applications. Hence, we rely on reinforcement learning (RL) to derive an optimal Q-function. RL uses temporal-difference to provide a real-time solution for the GPI method [18]. As a result, in Section V-A, we use Q-*learning*, as a specific method of RL, to solve (6).

### B. Factored MDP

To this point, we defined the Q-function over the joint state-action space of all the agents, i.e., $\mathcal{X} \times \mathcal{A}$. We refer to this Q-function as the global Q-function. According to [29], Q-*learning* finds the optimal solution to a single MDP with probability one. However, in large MDPs, due to exponential increase in the size of the joint state-action space with respect to the number of agents, the solution to the problem becomes intractable. To resolve this issue, we use *factored* MDPs as a decomposition technique for large MDPs. The idea in factored MDPs is that many large MDPs are generated by systems with many parts that are weakly interconnected. Each part has its associated state variables and the state space can be factored into subsets accordingly. The definition of the subsets affects the optimality of the solution [34], and investigating the optimal factorization method helps with understanding the optimality of multi-agent RL solutions [35]. In [36] power control of a multi-hop network is modeled as an MDP and the state set is factorized into multiple subsets each referring to a single hop. The authors in [37] show that the subsets can be defined based on the local knowledge of the agents from the environment. Meanwhile, we aim to distribute the power control to the nodes of the network. Therefore, due to the definition

of the problem in Section III and the fact that each FBS is only aware of its own power, we use the assumption in [37] and define the individual action set of the agents, i.e., $\mathcal{A}_k$, as the subsets of the joint action set. Consequently, the resultant Q-function for the $k$th agent is defined as $Q_k(\mathbf{x}_k, a_k)$, in which $a_k \in \mathcal{A}_k$, $\mathbf{x}_k \in \mathcal{X}_k$ is the state vector of the $k$th agent, and $\mathcal{X}_k$, $k \in \mathcal{K}$, are the subsets of the global state set of the system, i.e., $\mathcal{X}$.

In factored MDPs, We assume that the reward function is factored based on the subsets, i.e.,

$$\mathbf{R}(\mathbf{x}, \mathbf{a}) = \sum_{k \in \mathcal{K}} R_k(\mathbf{x}_k, a_k), \tag{7}$$

where, $R_k(\mathbf{x}_k, a_k)$ is the local reward function of the $k$th agent. Moreover, we also assume that the transition probabilities are factored, i.e., for the $k$th subsystem we have

$$\Pr(\mathbf{x}_k'|\mathbf{x}, \mathbf{a}) = \Pr(\mathbf{x}_k'|\mathbf{x}_k, a_k), \ (\mathbf{x}, \mathbf{a}) \in \mathcal{X} \times \mathcal{A}, \ (\mathbf{x}_k, a_k) \in \mathcal{X}_k \times \mathcal{A}_k, \ \mathbf{x}_k' \in \mathcal{X}_k. \tag{8}$$

The value function for the global MDP is given by

$$\mathbf{V}(\mathbf{x}) = \mathbb{E}\left[\sum_{t=0}^{\infty} \beta^t \mathbf{R}^{(t+1)}(\mathbf{x}, \mathbf{a})\right] = \mathbb{E}\left[\sum_{t=0}^{\infty} \beta^t \sum_{k \in \mathcal{K}} R_k^{(t+1)}(\mathbf{x}_k, a_k)\right] = \sum_{k \in \mathcal{K}} V_k(\mathbf{x}_k), \tag{9}$$

where, $V_k(\mathbf{x}_k)$ is the value function of the $k$th agent. Therefore, the derived policy has the value function equal to the linear combination of local value functions. Further, according to (5), for each agent $k \in \mathcal{K}$

$$Q_k(\mathbf{x}_k, a_k) = R_k(\mathbf{x}_k, a_k) + \beta \sum_{\mathbf{x}_k'} \Pr(\mathbf{x}_k'|\mathbf{x}_k, a_k) V_k(\mathbf{x}_k'), \tag{10}$$

and for the global Q-function

$$
\begin{aligned}
\mathbf{Q}(\mathbf{x}, \mathbf{a}) &= \mathbf{R}(\mathbf{x}, \mathbf{a}) + \beta \sum_{\mathbf{x}' \in \mathcal{X}} \Pr(\mathbf{x}'|\mathbf{x}, \mathbf{a}) \mathbf{V}(\mathbf{x}') \\
&= \sum_{k \in \mathcal{K}} R_k(\mathbf{x}_k, a_k) + \beta \sum_{\mathbf{x}' \in \mathcal{X}} \Pr(\mathbf{x}'|\mathbf{x}, \mathbf{a}) \sum_{k \in \mathcal{K}} V_k(\mathbf{x}_k) \\
&= \sum_{k \in \mathcal{K}} R_k(\mathbf{x}_k, a_k) + \beta \sum_{k \in \mathcal{K}} \sum_{\mathbf{x}_k' \in \mathcal{X}_k} \Pr(\mathbf{x}_k'|\mathbf{x}, \mathbf{a}) V_k(\mathbf{x}_k) \\
&= \sum_{k \in \mathcal{K}} R_k(\mathbf{x}_k, a_k) + \beta \sum_{k \in \mathcal{K}} \sum_{\mathbf{x}' \in \mathcal{X}_k} \Pr(\mathbf{x}_k'|\mathbf{x}_k, a_k) V_k(\mathbf{x}_k) = \sum_{k \in \mathcal{K}} Q_k(\mathbf{x}_k, a_k).
\end{aligned}
\tag{11}
$$

Therefore, based on the assumptions in (7) and (8), the global Q-function can be approximated with the linear combination of local Q-functions. Further, (11) results in a distributed and scalable architecture for the framework.
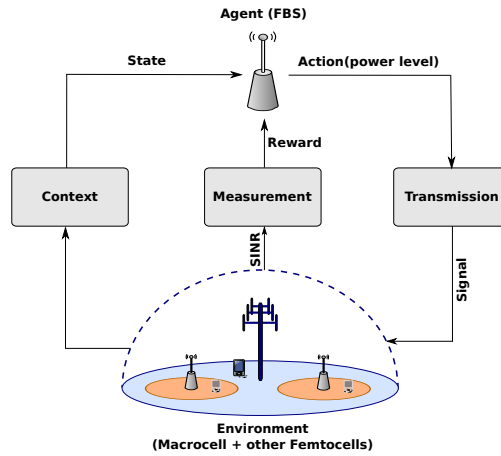
Figure 2. The proposed learning framework: the environment from the point of view of an agent (FBS), and its interaction with the environment in the learning procedure. Context defines the data needed to derive the state of the agent. Measurement refers to calculations needed to derive the reward of the agent.

### C. Femtocell Network as Multi-Agent MDP

In a wireless communication system, the resource management policy is equivalent to the policy function in an MDP. To integrate the femtocell network in a multi-agent MDP, we define the followings according to Fig. 2.

- **Environment**: From the view point of an FBS, the environment is comprised of the macrocell and all other femtocells.

- **Agent**: Each FBS is an independent agent in the MDP. In this paper, the terms of agent and FBS are used interchangeably. An agent has three objectives: (i) improving its sum transmission rate, (ii) guaranteeing the required SINR for its user (i.e., $\Gamma_k$), and (iii) meeting the required SINR for the MUE.

- **Action set** ($\mathcal{A}_k$): The transmit power level is the action of an FBS. The $k$th FBS chooses its transmit power from the set $\mathcal{A}_k$ which covers the space between $p_{\min}$ and $p_{\max}$. $p_{\min}$ and $p_{\max}$ denote the minimum and maximum transmit power of the FBS, respectively. In general, the FBS has no knowledge of the environment and it chooses its actions with the same probability in the training mode. Therefore, equal step sizes of $\Delta p$ are chosen between $p_{min}$ and $p_{max}$ to construct the set $\mathcal{A}_k$.

- **State set** ($\mathcal{X}_k$): State set directly affects the performance of the MUE and the FUEs. To this end, we define four variables to represent the state of the network. The state set variables are defined based on the constraints of the optimization problem in (3). We define the variables

$X_1$ and $X_2$ as indicators of the performance of the FUE and the MUE. On the other hand, the relative location of an FBS with respect to the MUE and the MBS is important and affects the interference power at the MUE caused by the FBS, and the interference power at the FBS causes by the MBS. Therefore, we define $X_3$ as an indicator of the interference imposed on the MUE by the FBS, and $X_4$ as an indicator of interference imposed on the femtocell by the MBS. The state variables are defined as

- $X_1 \in \{0, 1\}$: The value of $X_1$ indicates whether the FBS is supporting its FUE with the required minimum SINR or not. $X_1$ is defined as $X_1 = \mathbb{1}_{\{\gamma_k \geq \Gamma_k\}}$.

- $X_2 \in \{0, 1\}$: The value of $X_2$ indicates whether the MUE is being supported with its required minimum SINR or not. $X_2$ is defined as $X_2 = \mathbb{1}_{\{\gamma_0 \geq \Gamma_0\}}$.

- $X_3 \in \{0, 1, 2, ..., N_1\}$: The value of $X_3$ defines the location of the FBS compared to $N_1$ concentric rings around the MUE. The radius of rings are $d_1, d_2, ..., d_{N_1}$.

- $X_4 \in \{0, 1, 2, ..., N_2\}$: The value of $X_4$ defines the location of the FBS compared to $N_2$ concentric rings around the MBS. The radius of rings are $d'_1, d'_2, ..., d'_{N_2}$.

The $k$th FBS calculates $\gamma_k$ based on the channel equality indicator (CQI) received from its related FUE to assess $X_1$. The MBS is aware of the SINR of the MUE user, i.e., $\gamma_0$, and the relative location of the FBS concerning itself and the MUE. Therefore, the FBS obtains the required information to asses the $X_2$, $X_3$, and $X_4$ variables via backhaul and feedback from the MBS.

Here, we defined the state variables as a function of each FBS's SINR and location. Therefore, in high SINR regime, the state of FBSs can be assumed to be independent of each other. In Section VI, we will examine different possible state sets to investigate the effect of the above state variables on the performance of the network.

## V. Q-DPA, REWARD FUNCTION, AND SAMPLE COMPLEXITY

In this section, we present Q-DPA, which is an application of the proposed framework. Q-DPA details the learning method, the learning rate, and the training procedure. Then, the proposed reward function is defined. Finally, the required sample complexity for the training is derived.

### A. Q-learning Based Distributed Power Allocation (Q-DPA)

To solve the Bellman equation in (6), we use Q-*learning*. The reasoning for choosing the RL method and advantages of Q-*learning* are explained in Sections IV-A and I-A, respectively. The

Q-*learning* update rule to evaluate a policy for the global Q-function can be represented as [29]

$$\mathbf{Q}(\mathbf{x}^{(t)}, \mathbf{a}^{(t)}) \leftarrow \mathbf{Q}(\mathbf{x}^{(t)}, \mathbf{a}^{(t)}) + \alpha^{(t)}(\mathbf{x}, \mathbf{a}) \left( \mathbf{R}^{(t+1)}(\mathbf{x}^{(t)}, \mathbf{a}^{(t)}) + \beta \underbrace{\max_{\mathbf{a}'} \mathbf{Q}(\mathbf{x}^{(t+1)}, \mathbf{a}')}_{(M)} - \mathbf{Q}(\mathbf{x}^{(t)}, \mathbf{a}^{(t)}) \right),$$

(12)

where $\mathbf{a}' \in \mathcal{A}$, $\alpha^{(t)}(\mathbf{x}, \mathbf{a})$ denotes the learning rate at time step $t$, and $\mathbf{x}^{(t)}$ is the new state of the network. The term $M$ is the maximum value of the global Q-function that is available at the new state $\mathbf{x}^{(t+1)}$. After each iteration, the FBSs will receive the delayed reward $\mathbf{R}^{(t+1)}(\mathbf{x}^{(t)}, \mathbf{a}^{(t)})$ and then the global Q-function will be updated according to (12).

In the prior works [19]–[21], [23], [24], a constant learning rate was used for Q-*learning* to solve the required optimization problems. However, according to [38], in finite number of iterations, the performance of Q-*learning* can be improved by applying a decaying learning rate. Therefore, we use the following learning rate

$$\alpha^{(t)}(\mathbf{x}, \mathbf{a}) = \frac{1}{[1 + t(\mathbf{x}, \mathbf{a})]},$$

(13)

in which $t(\mathbf{x}, \mathbf{a})$ refers to the number of times, until time step $t$, that the state-action pair $(\mathbf{x}, \mathbf{a})$ is visited. It is worth mentioning that, by using the above learning rate, we need to keep track of the number of times each state-action pair has been visited during training, which requires more memory. Therefore, at the cost of more memory, a better performance can be achieved.

There are two alternatives available for the training of new FBSs as they join the network, they can use independent learning or cooperative learning. In independent learning, each FBS tries to maximize its own Q-function. In other words, using the factorization method in Section IV-B, the term $M$ in (12) is approximated as

$$M = \max_{\mathbf{a}'} \sum_{k \in \mathcal{K}} Q_k(\mathbf{x}_k^{(t+1)}, a_k') \approx \sum_{k \in \mathcal{K}} \max_{a_k'} Q_k\left(\mathbf{x}_k^{(t+1)}, a_k'\right).$$

(14)

In cooperative learning, the FBSs share their local Q-functions and will assume that the FBSs with the same state make the same decision. Hence, term $M$ is approximated as

$$M = \max_{\mathbf{a}'} \sum_{k \in \mathcal{K}} Q_k(\mathbf{x}_k^{(t+1)}, a_k') \approx \max_{a_k'} \sum_{k \in \mathcal{K}'} Q_k\left(\mathbf{x}_k^{(t+1)}, a_k'\right),$$

(15)

where $\mathcal{K}'$ is the set of FBSs with the same state $\mathbf{x}_k^{(t+1)}$. Cooperative Q-*learning* may result in a higher cumulative reward [39]. However, cooperation will result in the same policy for FBSs with

the same state and additional overhead since the Q-functions between FBSs need to be shared over the backhaul network. The local update rule for the $k$th FBS can be derived from (12) as

$$Q_k(\mathbf{x}_k^{(t)}, a_k^{(t)}) \leftarrow Q_k(\mathbf{x}_k^{(t)}, a_k^{(t)}) + \alpha^{(t)} \left( R^{(t+1)} \left( \mathbf{x}_k^{(t)}, a_k^{(t)} \right) + \beta Q_k \left( \mathbf{x}_k^{(t+1)}, a_k^* \right) - Q_k(\mathbf{x}_k^{(t)}, a_k^{(t)}) \right), \tag{16}$$

where, $R^{(t+1)} \left( \mathbf{x}_k^{(t)}, a_k^{(t)} \right)$ is the reward of the $k$th FBS, and $a_k^*$ is defined as

$$\arg\max_{a_k'} Q_k \left( \mathbf{x}_k^{(t+1)}, a_k' \right), \tag{17}$$

and

$$\arg\max_{a_k'} \sum_{k \in \mathcal{K}'} Q_k \left( \mathbf{x}_k^{(t+1)}, a_k' \right), \tag{18}$$

for independent and cooperative learning, respectively.

In this paper, a tabular representation is used for the Q-function in which the rows of the table refer to the states and the columns refer to the actions of an agent. Generally, for large state spaces, neural networks are more efficient to use as Q-functions, however, part of this work is focused on the effect of state space variables. Therefore, we avoid large number of state variables. On the other hand, we provide exhaustive search solution to investigate the optimality of our solution which is not possible for large state spaces.

The training for an FBS happens over $L$ frames. In the beginning of each frame, the FBS chooses an action, i.e., transmit power. Then, the FBS sends a frame to the intended FUE. The FUE feeds back the required measurements such as CQI so the FBS can estimate the SINR at the FUE, and calculate the reward based on (24). Finally, the FBS updates its Q-table according to (16).

Due to limited number of training frames, each FBS needs to select its actions in a way that covers most of the action space and improves the policy at the same time. Therefore, the FBS chooses the actions with a combination of exploration and exploitation, known as an $e$-greedy exploration. In the $e$-greedy method, the FBS acts greedily with probability $1-e$ (i.e., exploiting) and randomly with probability $e$ (i.e., exploring). In exploitation, the FBS selects an action that has the maximum value in the current state in its own Q-table (independent learning) or in the summation of Q-tables (cooperative learning). In exploring, the FBS selects an action randomly to cover action space and avoid biasing to a local maximum. In [18], it is shown that for a

limited number of iterations the $e$-greedy policy results in a closer final value to the optimal value compared to only exploiting or exploring.

It is worth mentioning that the overhead of sharing Q-tables depends on the definition of the state model $\mathcal{X}_k$ according to Section IV-C. For instance, assuming the largest possible state model as $\mathcal{X}_k = \{X_1, X_2, X_3, X_4\}$. The variables $X_3$ and $X_4$ depend on the location of the FBS and are fixed during training. Therefore, one training FBS uses four rows of its Q-table and just needs the same rows from other FBSs. Hence, if the number of active FBSs is $|\mathcal{K}|$, the number of messages to the FBS in each training frame is $4 \times (|\mathcal{K}| - 1)$, each of size $|\mathcal{A}_k|$.

## B. Proposed Reward Function

The design of the reward function is essential because it directly impacts the objectives of the FBS. Generally, there has not existed a quantitative approach to designing the reward function. Here, we present a systematic approach for deriving the reward function based on the nature of the optimization problem under consideration. Then, we compare the behavior of the designed reward function to the ones in [19]–[21].

The reward function for the $k$th FBS is represented as $R_k$. According to the Section IV-C, the $k$th FBS has knowledge of the minimum required SINR for the MUE, i.e. $\Gamma_0$, and minimum required SINR for its related FUE, i.e. $\Gamma_k$. Also, after taking an action in each step, the $k$th FBS has access to the rate of the MUE, i.e. $r_0$ and the rate of its related FUE, i.e. $r_k$. Therefore, $R_k$ is considered as a function of the above four variables as $R_k : (r_0, r_k, \Gamma_0, \Gamma_k) \rightarrow \mathbb{R}$.

In order to design the appropriate reward function, we need to estimate the progress of the $k$th FBS toward the goals of the optimization problem. Based on the input arguments to the reward function, we define two progress estimators, one for the MUE as $(r_0 - \log_2 (1 + \Gamma_0))$ and one for the $k$th FUE as $(r_k - \log_2 (1 + \Gamma_k))$. To reduce computational complexity, we define the reward function as a polynomial function of the defined progress estimators as

$$R_k (r_0, r_k, \Gamma_0, \Gamma_k) = (r_0 - \log_2 (1 + \Gamma_0))^{k_1} + (r_k - \log_2 (1 + \Gamma_k))^{k_2} + C, \qquad (19)$$

where, $k_1$ and $k_2$ are integers and $C \in \mathbb{R}$ is a constant referred to as the bias of the reward function.

The constant bias, $C$, in the reward function has two effects on the learning algorithm: (i) The final value of the states for a given policy $\pi$, and (ii) the behavior of the agent in the beginning

of the learning process as follows:

1) Effect of bias on the final value of the states: Assume the reward function, $R_1 = f(\cdot)$, and the reward function $R_2 = f(\cdot) + C$, $C \in \mathbb{R}$. We define the value of state $\mathbf{x}$ for a given policy $\pi$ using $R_1$ as $V_1(\mathbf{x})$ and the value of the state $\mathbf{x}$ for the same policy using $R_2$ as $V_2(\mathbf{x})$. According to (4)

$$V_2(\mathbf{x}) = \mathbb{E}_\pi \left[ \sum_{t=0}^{\infty} \beta^t \left( f^{(t+1)}(\cdot) + C \right) \right] = \mathbb{E}_\pi \left[ \sum_{t=0}^{\infty} \beta^t f^{(t+1)}(\cdot) \right] + C \sum_{t=0}^{\infty} \beta^t = V_1(\mathbf{x}) + \frac{C}{1-\beta}. \tag{20}$$

Therefore, bias of the reward function adds the constant value $\frac{C}{1-\beta}$ to the value of the states. However, all the states are affected the same after the convergence of the algorithm.

2) Effect of bias in the beginning of the learning process: This effect is studied using the action-value function of an agent, i.e., the Q-function. Assume that the Q-function of the agent is initialized with zero values and the reward function is defined as $R = f(\cdot) + C$. Further let us consider the first transition of the agent from state $\mathbf{x}'$ to state $\mathbf{x}''$ happens by taking action $a$ at time step $t$, i.e., $\mathbf{x}^{(t)} = \mathbf{x}'$ and $\mathbf{x}^{(t+1)} = \mathbf{x}''$. The update rule at time step $t$ is given by (16)

$$Q(\mathbf{x}', a) \leftarrow Q(\mathbf{x}', a) + \alpha^{(t)}(\mathbf{x}', a) \left( R(\mathbf{x}', a) + \beta \max_{a'} Q(\mathbf{x}'', a') - Q(\mathbf{x}', a) \right)$$
$$\leftarrow \alpha^{(t)}(\mathbf{x}', a) \left( f(\cdot) + \beta \max_{a'} Q(\mathbf{x}'', a') \right) + \underbrace{\alpha^{(t)}(\mathbf{x}', a) C}_{(A)}. \tag{21}$$

According to the above, after the first transition from the state $\mathbf{x}'$ to the state $\mathbf{x}''$, the Q-value for the state $\mathbf{x}'$ is biased by the term (A). If $(A > 0)$, the value of the state $\mathbf{x}'$ increases and if $(A < 0)$, the value of the state $\mathbf{x}'$ decreases. Therefore, the already visited states will be more or less attractive to the agent in the beginning of the learning process as long as the agent has not explored the state-space enough.

The change of behavior of the agent in the learning process can be used to bias the agent towards the desired actions or states. However, in basic Q-learning the agent has no knowledge in prior about the environment. Therefore, we select the bias equal to zero, $C = 0$, and define the reward function as

**Definition 1.** The reward function for the $k$th FBS, $R_k : (r_0, r_k, \Gamma_0, \Gamma_k) \to \mathbb{R}$, is a continuous and differentiable function on $\mathbb{R}^2$ defined as

$$R_k \left( r_0, r_k, \Gamma_0, \Gamma_k \right) = \left( r_0 - \log_2 \left( 1 + \Gamma_0 \right) \right)^{k_1} + \left( r_k - \log_2 \left( 1 + \Gamma_k \right) \right)^{k_2}, \tag{22}$$

where $k_1$ and $k_2$ are integers.

The objective of the FBS is to maximize its transmission rate. On the other hand, high transmission rate for the MUE is a priority for the FBS. Therefore, $R_k$ should have the following property

$$\frac{\partial R_k}{\partial r_i} \geq 0, \; i = 0, k. \tag{23}$$

The above property implies that higher transmission rate for the FBS or the MUE results in higher reward. Hence, considering Definition 1, we design a reward function that motivates the FBSs to increase $r_k$ and $r_0$ as much as possible even more than the required rate as follow

$$R_k = \left( r_0 - \log_2 \left( 1 + \Gamma_0 \right) \right)^{2m-1} + \left( r_k - \log_2 \left( 1 + \Gamma_k \right) \right)^{2m-1}, \tag{24}$$

where $m$ is an integer. The above reward function considers the minimum rate requirements of the FUE and the MUE, while encourages the FBS to increase transmission rate of both.

To further understand the proposed reward function, we discuss reward functions that are used by [19]–[21]. We refer to the designed reward function in [19] as quadratic, in [20] as exponential, and in [21] as proximity reward functions. The quadratic reward function is designed based on a conservative approach. In fact, the FBS is enforced to select actions that result in transmission rate close to the minimum requirement. Therefore, higher or lower rate than the minimum requirement results in a same amount of reward. The behavior of the quadratic reward function can be explained as follow

$$\frac{\partial R_k}{\partial r_i} \times \left( r_i - \log_2 \left( 1 + \Gamma_i \right) \right) \leq 0, \; i = 0, k. \tag{25}$$

The above property implies that if the rate of the FBS or the MUE is higher than the minimum requirement, the actions that increase the rate will decrease the reward. Hence, this property is against increasing sum transmission rate of the network. The exponential and proximity reward functions have the property in (23) for the rate of the FBS, and the property in (25) for the rate of the MUE. In another words, they satisfy the following properties
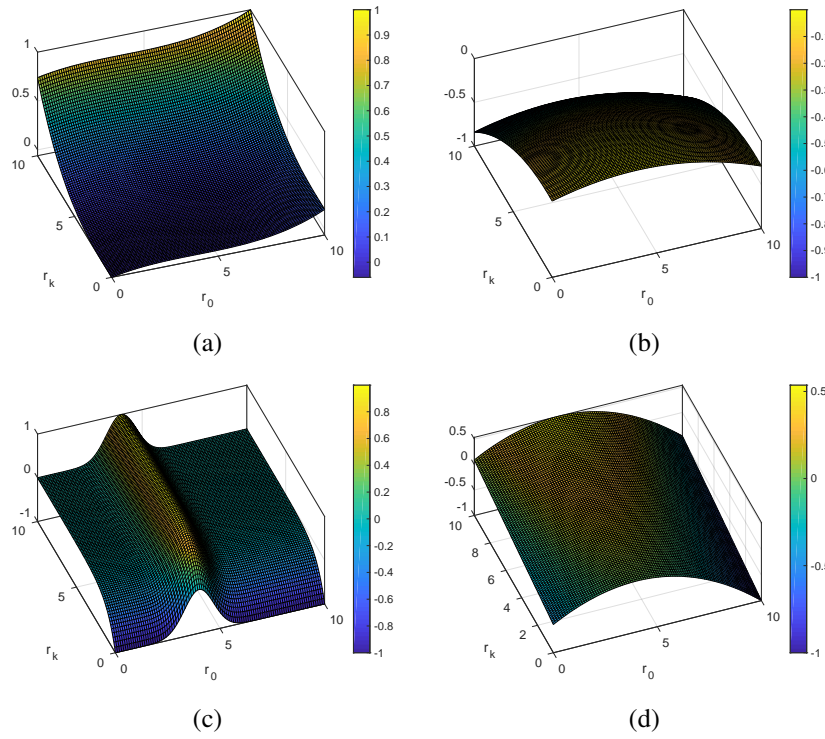
Figure 3. Reward functions: (a) Proposed reward function with $m = 2$, (b) Quadratic reward function with zero maximum at $(4.0, 0.5)$, (c) Exponential reward function, (d) Proximity reward function.

$$\frac{\partial R_k}{\partial r_0} \times (r_0 - \log_2(1 + \Gamma_0)) \leq 0,$$
$$\frac{\partial R_k}{\partial r_k} \geq 0. \tag{26}$$

As the density of the FBSs increases, the above properties result in increasing transmit power to achieve higher individual rate for a FUE while introducing higher interference for the MUE and other neighbor FUEs. In fact, as increasing the FUE rate is rewarded, taking actions that result in increasing the MUE rate decreases the reward. However, the FBS should have the option of decreasing its transmit power to increase the rate of the MUE. This behavior is important since it causes an FBS to produce less interference for its neighboring femtocells. Therefore, we give equal opportunity for increasing the rate of the MUE or the FUE.

The value of reward functions for different FBSs is different, however they have the same behavior. Here, we plot the value of the four reward functions that are discussed above. The plots refers to the proposed (Fig. 3a), quadratic (Fig. 3b), exponential (Fig. 3c), and proximity (Fig. 3d) reward functions. The important information that can be obtained from these plots are the maximal points of the reward functions, behavior of the reward functions around minimum

requirements, and behavior of the reward functions by increasing $r_k$ or $r_0$. The proposed reward function in Fig. 3a shows pushing the FBS to select transmit power levels that increase both $r_k$ and $r_0$, while other reward functions have their maximum around the minimum rate requirements.

## C. Sample Complexity

In each training frame, Q-DPA collects one sample from the environment represented as the state-action pair in the Q-function. Sample complexity is defined as the minimum number of samples that is required to train the Q-function to achieve an $\epsilon$-optimal policy. For $\epsilon > 0$ and $\delta \in (0, 1]$, $\pi$ is an $\epsilon$-optimal policy if [40]

$$\Pr\left(\|Q^* - Q_\pi\| < \epsilon\right) \geq 1 - \delta. \tag{27}$$

The sample complexity depends on the exploration policy that is generating the samples. In Q-DPA, $e$-greedy policy is used as the exploration policy. However, $e$-greedy policy depends on the Q-function of the agent which is being updated. In fact, the distribution of $e$-greedy policy is unknown. Here, we provide a general bound on the sample complexity of Q-*learning*.

**Proposition 1.** Assume $R_{max}$ is the maximum of the reward function for an agent and $Q^{(T)}$ is the action-value for state-action pair $(x, a)$ after $T$ iterations. Then, with probability at least $1 - \delta$, we have

$$\|Q^* - Q^{(T)}\| \leq \frac{2R_{max}}{(1 - \beta)} \left[ \frac{\beta}{T(1 - \beta)} + \sqrt{\frac{2}{T} \ln \frac{2|\mathcal{X}|.|\mathcal{A}|}{\delta}} \right]. \tag{28}$$

*Proof.* See Appendix A. ☐

This proposition proves the stability of Q-*learning* and helps us to provide a minimum number of iterations to achieve $\epsilon > 0$ error with respect to $Q^*$ with probability $1 - \delta$ for each state-action pair. By assuming the right term of the above inequality as $\epsilon$, the following Corollary is concluded.

**Corollary 1.** For any $\epsilon > 0$, after

$$T = \Omega\left(\frac{8R_{max}^2}{\epsilon^2(1 - \beta)^2} \ln \frac{2|\mathcal{X}|.|A_k|}{\delta}\right) \tag{29}$$

number of iterations, $Q^{(T)}$ reaches $\epsilon$-optimality with probability at least $1 - \delta$.

## VI. SIMULATION RESULTS

The objective of this section is to validate the performance of the Q-DPA algorithm with different learning configurations in a dense urban scenario. We first introduce the simulation setup and parameters. Then, we introduce four different learning configurations and we analyze the trade-offs between them. Finally, we investigate the performance of the Q-DPA with different reward functions introduced in Section V-B. For the sake of simplicity, we use the notation IL as independent learning and CL as cooperative learning.

### A. Simulation Setup

We use a dense urban scenario as the setup of the simulation as illustrated in Fig. 4. We consider one macrocell with radius 350 m which supports multiple MUEs. The MBS assigns a subband to each MUE. Each MUE is located within a block of apartments and each block contains two strip of apartments. Each strip has five apartments of size $10 \text{ m} \times 10 \text{ m}$. There is one FBS located in the middle of each apartment which supports an FUE within a 5 m distance. We assume that the FUEs are always inside the apartments. The FBSs are closed-access, therefore, the MUE is not able to connect to any FBS, however, it receives interference from the FBSs working on the same subband as itself. Here, we assume that the MUE and all the FBSs work on the same sub-carriers to consider the worst case scenario (high interference scenario). However, the extension of the simulation to the multi-carrier scenario is straight forward but does not affect our investigations. We assume the block of apartments is located on the edge of the macrocell, i.e., 350 m distance from the MBS, and the MUE is assumed to be in between the two strip of apartments.

In these simulations, in order to initiate the state variables $X_3$ and $X_4$ in Section IV-C, the number of rings around the MBS and the MUE are assumed to be three ($N_1 = N_2 = 3$). Although, as the density increases, more rings with smaller diameters can be used to more clearly distinguish between the FBSs.

It is assumed that the FBSs and the MBS operate at $f = 2.0$ GHz. The MBS allocates 33 dBm as its transmit power, and the FBSs choose their transmit power from a range of 5 dBm to 15 dBm with power steps of 1 dB. In order to model the pathloss, we use the urban dual strip model from 3GPP TR 36.814 [41]. The pathloss model of different links are provided in Table I. In Table I, $R$ is the distance between a transmitter and a receiver in meters, $L_{ow}$ is the
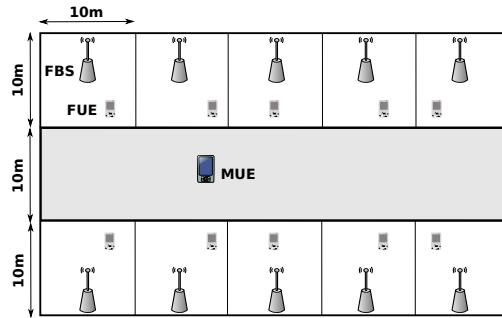
Figure 4. Dense urban scenario with a dual strip apartment block located at distance of 350 m of the MBS; FUEs are randomly located inside each apartment.

Table I. Urban dual strip pathloss model

| Link | PL(dB) |
|------|--------|
| MBS to MUE | $15.3 + 37.6 \log_{10} R$ , |
| MBS to FUE | $15.3 + 37.6 \log_{10} R + L_{ow}$ , |
| FBS to FUE (same apt strip) | $56.76 + 20 \log_{10} R + 0.7 d_{2D,indoor}$ , |
| FBS to FUE (different apt strip) | $max(15.3 + 37.6 \log_{10} R, 38.46 + 20 \log_{10} R) + 18.3 + 0.7 d_{2D,indoor} + L_{ow}.$ |

wall penetration loss which is set to 20 dB [41]. $d_{2D,indoor}$ is the 2-dimensional distance. We assume that the apartments are single floor, therefore, $d_{2D,indoor} \approx R$. The fourth row of the pathloss models is used for the links between the FBSs and the MUE.

The minimum SINR requirements for the MUE and the FUEs are defined based on the required rate needed to support their corresponding user. In our simulations, the minimum required transmission rate to meet the QoS of the MUE is assumed to be 4 (b/s/Hz), i.e., $\log_2(1 + \Gamma_0) = 4$ (b/s/Hz). Moreover, for the FUEs the minimum required rate is set to 0.5 (b/s/Hz), i.e, $\log_2(1 + \Gamma_k) = 0.5$ (b/s/Hz), $k \in \mathcal{K}$. It is worth mentioning that by knowing the media access control (MAC) layer parameters, the values of the required rates can be calculated using [42, Eqs. (20) and (21)].

To perform Q-*learning*, the minimum number of required frames, i.e., $L$, is calculated based on achieving 90% optimality, with probability of at least 0.9, i.e., $\delta = 0.1$. The simulation parameters are given in Table II. The value of the Q-*learning* parameters are selected according to our simulations and references [19]–[24].

The simulation starts with one femtocell. The FBS starts running Q-DPA in Section V-A using IL. After convergence, the next FBS is added to the network. The new FBS runs Q-DPA, while the other FBS is already trained, and will just act greedy to choose its transmit power.

Table II. Simulation Parameters

| Default parameters | Value | State parameters | Value |
|---|---|---|---|
| Frame time | 2 ms | $d_1', d_2', d_3'$ | 50, 150, 400 m |
| UE thermal noise | -174 dBm/Hz | $d_1, d_2, d_3$ | 17.5, 22.5, 45 m |
| Traffic model | Fullbuffer | | |
| **FBS parameters** | **Value** | **Q-DPA parameters** | **Value** |
| $p_{min}$ | 5 dBm | Training period (iterations) $L$ | $T \times \|\mathcal{X}\|.\|\mathcal{A}_k\|$ frames |
| $p_{max}$ | 15 dBm | Learning parameter $\beta$ | 0.9 |
| $\Delta p$ | 1 dBm | Exploratory probability ($e$) | 10% |

After convergence of the second FBS, the next one is added to the network, and so on. We represent all the results versus the number of active femtocells in the system, from one to ten. Considering the size of the apartment block, and the assumption that all femtocells operate on the same frequency range, the density of deployment varies approximately from 600 FBS/$km^2$ to 6000 FBS/$km^2$.

### B. Performance of Q-DPA

Here, we show the simulation results of distributed power allocation with Q-DPA. First, we define two different state sets. The sets are defined as $\mathcal{X}_1 = \{X_1, X_3, X_4\}$ and $\mathcal{X}_2 = \{X_2, X_3, X_4\}$. In both sets, FBSs are aware of their relative location to the MUE and the MBS due to the presence of $X_3$ and $X_4$, respectively. The state set $\mathcal{X}_1$ gives knowledge of the status of the FUE to the FBS, and the state set $\mathcal{X}_2$ provides knowledge of the status of the MUE to the FBS.

In order to understand the effect of independent and cooperative learning, and the effect of different state sets, we use four different learning configurations as: independent learning with each of the two state sets as IL+$\mathcal{X}_1$ and IL+$\mathcal{X}_2$, and cooperative learning with each of the two state sets as CL+$\mathcal{X}_1$ and CL+$\mathcal{X}_2$. The results are compared with *greedy* approach in which each FBS chooses maximum transmit power. The simulation results are shown in three figures as: transmission rate of the MUE (Fig. 5a), sum transmission rate of the FUEs (Fig. 5b), and sum transmit power of the FBSs (Fig. 5c).

According to Fig. 5c, in the greedy algorithm, each FBS uses the maximum available power for transmission. Therefore, the greedy method introduces maximum interference for the MUE and has the lowest MUE transmission rate in Fig. 5a. On the other hand, despite using maximum
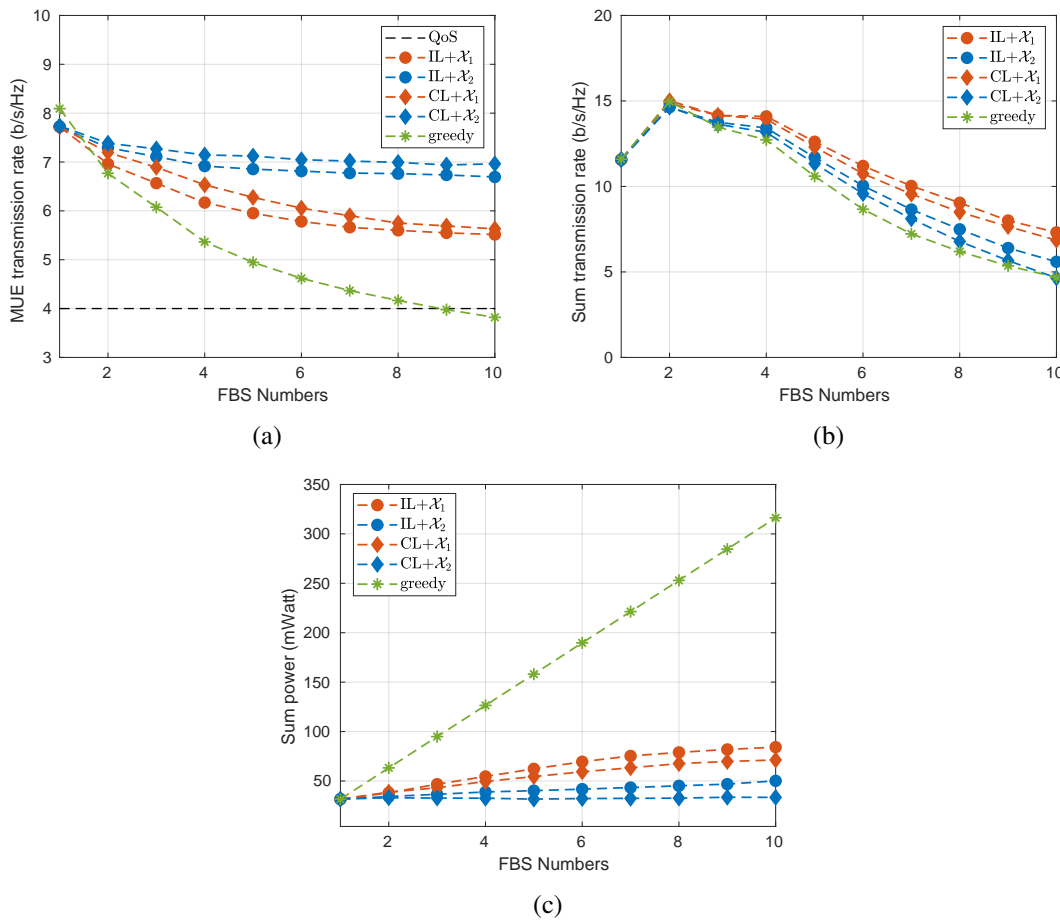
Figure 5. Performance of different learning configurations: (a) transmission rate of the MUE, (b) sum transmission rate of the FUEs, (c) sum transmit power of the FBSs.

Table III. Performance of different learning configurations. $1$ is the best, and $4$ is the worst.

| Learning configuration | $\sum p_k$ | $\sum r_k$ | $r_0$ |
|:---:|:---:|:---:|:---:|
| IL+$\mathcal{X}_1$ | 4 | 1 | 4 |
| CL+$\mathcal{X}_1$ | 3 | 3 | 3 |
| IL+$\mathcal{X}_2$ | 2 | 2 | 2 |
| CL+$\mathcal{X}_2$ | 1 | 4 | 1 |

power, the greedy algorithm does not achieve highest transmission rate for the FUEs either (Fig. 5b). This is again due to the high level of interference.

The state set $\mathcal{X}_2$ provides knowledge of MUE's QoS status to the learning FBSs. Therefore, as we see in Fig. 5a, the performance of IL with $\mathcal{X}_2$ is higher than the ones with $\mathcal{X}_1$. This statement is true for CL too. We can see the reverse of this conclusion in the FUEs' sum transmission

rate in Fig. 5b. The performance of IL with $\mathcal{X}_1$ is higher than IL with $\mathcal{X}_2$. This is because the FBSs are aware of the status of the FUE, therefore, they consider actions that result in the state variable $X_1 = \mathbb{1}_{\{\gamma_k \geq \Gamma_k\}}$ to be $1$. This is true in comparison of the states in CL too. In conclusion, the state set $\mathcal{X}_1$ works in favor of femtocells and the state set $\mathcal{X}_2$ benefits the MUE.

We conclude from the simulation results that IL and CL present different trade-offs. More specifically, IL supports a higher sum transmission rate for the FBSs and a lower transmission rate for the MUE, while CL can support a higher transmission rate for the MUE at the cost of an overall lower sum transmission rate for the FBSs. From a power consumption point of view, IL results in a higher power consumption when compared to that of CL. In general, IL trains an FBS to be selfish compared to CL. IL can be very useful when there is no means of communication between the agents. On the other hand, CL trains an FBS to be more considerate about other FBSs at the cost of communication overhead.

In Table III, we have compared the performance of the four learning configurations. In each column, number $1$ is used as a metric to refer to the highest performance achieved and number $4$ is used to refer to the lowest performance observed. The first column represents the summation of transmit powers of FBSs, the second column indicates the summation of transmission rates of the FUEs, and the third column denotes the transmission rate of the MUE.

## C. Reward Function Performance

Here, we compare the performance of the four reward functions discussed in Section V-B. Since the objective is to maximize the sum transmission rate of the FUEs, according to Table III, we choose the combination IL+$\mathcal{X}_1$ as the learning configuration. The performance of the reward functions are provided as the MUE transmission rate (Fig. 6a), sum transmission rate of the FUEs (Fig. 6b), and sum transmission power of the FBSs (Fig. 6c). In each figure, the solution of the optimization problem with exhaustive search and the performance of greedy method are provided. The exhaustive search provides us with the highest achievable sum transmission rate for the network. The quadratic, exponential, and proximity reward functions result in fast decaying of MUE transmission rate, while the proposed reward function results in a much slower decrease of the rate for the MUE. The proposed reward function manages to achieve a higher sum transmission rate compared to that of the other three reward functions as well. Fig. 6c indicates that the proposed reward function reduces the sum transmitted power at the FBSs which in
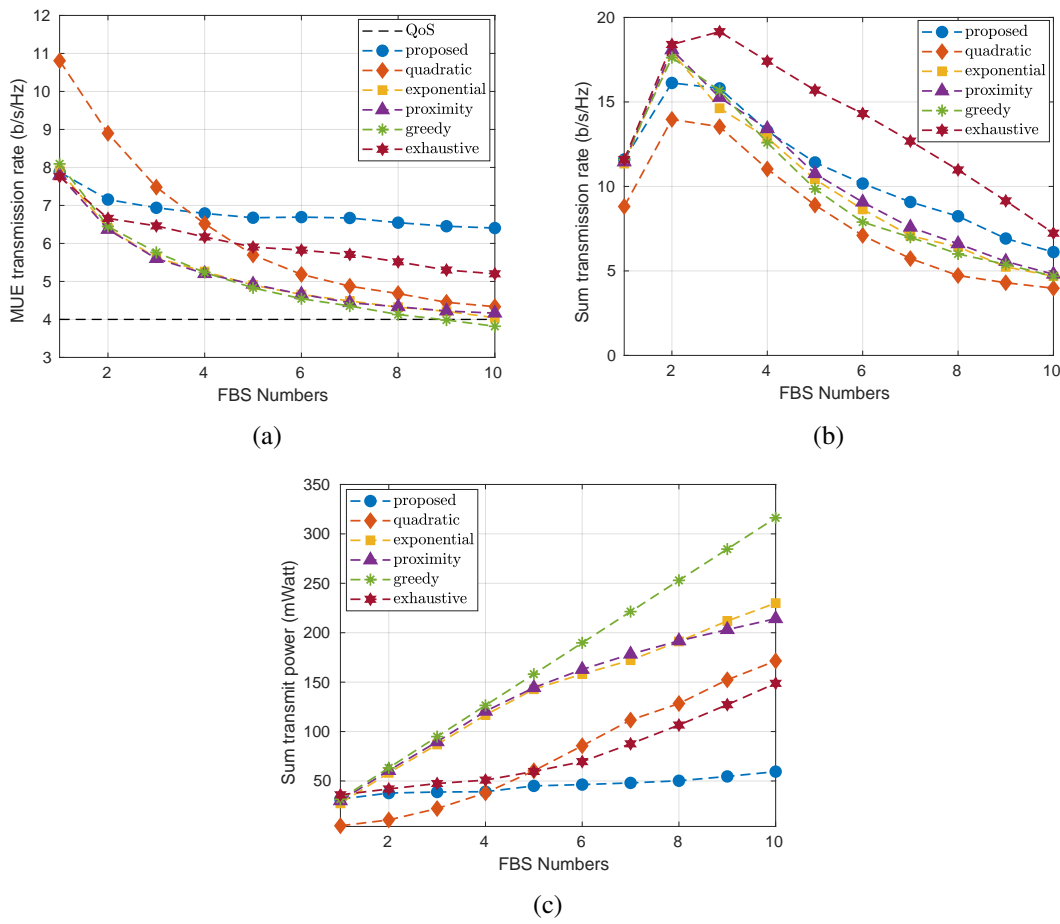
Figure 6. Performance of the proposed reward function compared to quadratic, exponential and proximity reward functions: (a) transmission rate of the MUE, (b) sum transmission rate of the FUEs, (c) sum transmit power of the FBSs.

turn could result in lower levels of interference at the FUEs. In comparison with the exhaustive search solution as the optimal solution, there is a gap of performance. For instance according to Fig. 6c, for eight number of FBSs, the proposed reward function uses an average of 50 mWatt less sum transmit power than the optimal solution. However, as we see in Fig. 6b and Fig. 6a, by using more power, the sum transmission rate can be improved and the transmission rate of the MUE can be decreased to the level of exhaustive solution without violating its minimum required rate. In our future works, we wish to cover this gap by using neural networks as the function approximator of the learning method.

## VII. CONCLUSION AND FUTURE WORK

In this paper, we propose a learning framework for a two-tier femtocell network. The framework enables addition of a new femtocell to the network, while the femtocell trains itself to adapt its transmit power to support its serving user while protecting the macrocell user. On the

other hand, the proposed method as a distributed approach can solve the power optimization problem in dense HetNets, while significantly reducing power usage. The proposed framework is generic and motivates the design of machine learning based SONs for management schemes in femtocell networks. Besides, the framework can be used as a bench test for evaluating the performance of different learning configurations such as Markov state models, reward functions and learning rates. Further, the proposed framework can be applied to other interference-limited networks such as cognitive radio networks as well.

In future work, it would be interesting to consider mmWave-enabled femtocells in the present setup. In fact, the high pathloss and shadowing along with the vulnerability of mmWave directional signals to the blockages impacts the learning outcome [43]. This will in turn affect the subsequent power optimization problem. In addition, as we discussed in simulation section in details, there is a performance gap between the proposed approach and the exhaustive search. Although, the proposed approach results in less computational complexity; we wish to improve and cover this gap by utilizing neural networks as the function approximator of the learning method. In fact, neural networks can handle the large state-action spaces more efficiently. Moreover, another future complementary work to achieve a higher sum data rate and fill the performance gap would be to feed the interference model of the network to the factorization process. This way, a better factorization can be provided for the global Q-function.

## APPENDIX A

### PROOF OF PROPOSITION 1

*Proof.* Assume an MDP represented as $(\mathcal{X}, \mathcal{A}, \Pr(y|x,a), r(x,a))$, a policy $\pi$ with value-function $V_\pi : \mathcal{X} \to \mathbb{R}$ and Q-function $Q_\pi : \mathcal{Z} \to \mathbb{R}$, $\mathcal{Z} = \mathcal{X} \times \mathcal{A}$. Here, $\mathcal{A}$ refers to action space of one agent and $k$ is the iteration index. According to (4), the maximum of the value-function can be fined as $V_{max} = \frac{R_{max}}{1-\beta}$. The Bellman optimality operator is defined as $(\mathrm{T}Q)(x,a) \triangleq r(x,a) + \beta \sum_{y \in \mathcal{X}} \Pr(y|x,a) \max_{b \in \mathcal{A}} Q(y,b)$. $\mathrm{T}Q$ is a contraction operator with factor $\beta$, i.e., $\|\mathrm{T}Q - \mathrm{T}Q'\| \leq \beta \|Q - Q'\|$ and $Q^*$ is a unique fixed-point of $(\mathrm{T}Q)(x,a)$, $\forall (x,a) \in \mathcal{Z}$. Further, for the ease of notation and readability the time step notation is slightly changed as $Q_k$ refers to the action-value function after $k$ iterations.

Assume that the state-action pair $(x,a)$ is visited $k$ times and $\mathcal{F}_k = \{y_1, y_2, ..., y_k\}$ are the visiting next states. At time step $k+1$, the update rule of Q-*learning* is

$$Q_{k+1}(x,a) = (1-\alpha_k)Q_k(x,a) + \alpha_k \mathtt{T}_k Q_k(x,a), \tag{30}$$

where, $\mathtt{T}_k Q_k$ is the empirical Bellman operator defined as $\mathtt{T}_k Q_k(x,a) \triangleq r(x,a) + \beta \max_{b \in \mathcal{A}} Q(y_k, b)$. (From this point, for simplicity, we remove the dependency on $(x,a)$). It is easy to show that $E[\mathtt{T}_k Q_k] = \mathtt{T}Q_k$, therefore, we define $e_k$ as the estimation error of each iteration as $e_k = \mathtt{T}_k Q_k - \mathtt{T}Q_k$. By using $\alpha_k = \frac{1}{k+1}$, the update rule of Q-*learning* can be written as

$$Q_{k+1} = \frac{1}{k+1}(kQ_k + \mathtt{T}Q_k + e_k). \tag{31}$$

Now, in order to prove Proposition 1, we need to state the following lemmas.

**Lemma 1.** For any $k \geq 1$

$$Q_k = \frac{1}{k}\sum_{i=0}^{k-1} \mathtt{T}_i Q_i = \frac{1}{k}\left(\sum_{i=0}^{k-1}\mathtt{T}Q_i + \sum_{i=0}^{k-1} e_i\right). \tag{32}$$

*Proof.* We prove this lemma by induction. The lemma holds for $k=1$ as $Q_1 = \mathtt{T}_0 Q_0 = \mathtt{T}Q_0 + e_0$. We now show that if the result holds for $k$, then it also holds for $k+1$. From (31) we have

$$Q_{k+1} = \frac{k}{k+1}Q_k + \frac{1}{k+1}(\mathtt{T}Q_k + e_k) = \frac{k}{k+1}\frac{1}{k}\left(\sum_{i=0}^{k-1}\mathtt{T}Q_i + \sum_{i=0}^{k-1}e_i\right) + \frac{1}{k+1}(\mathtt{T}Q_k + e_k)$$

$$= \frac{1}{k+1}\left(\sum_{i=0}^{k}\mathtt{T}Q_i + \sum_{i=0}^{k}e_i\right).$$

Thus (32) holds for $k \geq 1$ by induction. $\qquad\square$

**Lemma 2.** Assume that initial action-value function, $Q_0$, is uniformly bounded by $V_{max}$. Then, for all $k \geq 1$ we have $\|Q_k\| \leq V_{max}$ and $\|Q^* - Q_k\| \leq 2V_{max}$.

*Proof.* We first prove that $\|Q_k\| \leq V_{max}$ by induction. The inequality holds for $k=1$ as

$$\|Q_1\| = \|\mathtt{T}_0 Q_0\| = \|r + \beta \max Q_0\| \leq \|r\| + \beta \|Q_0\| \leq R_{max} + \beta V_{max} = V_{max}.$$

Now, we assume that for $1 \leq i \leq k$, $\|Q_k\| \leq V_{max}$ holds. First, $\|\mathtt{T}_k Q_k\| = \|r + \beta \max Q_k\| \leq \|r\| + \beta \|\max Q_k\| \leq R_{max} + \beta V_{max} = V_{max}$. Second, from Lemma 1 we have

$$\|Q_{k+1}\| = \frac{1}{k+1}\left\|\sum_{i=0}^{k}\mathtt{T}_i Q_i\right\| \leq \frac{1}{k+1}\sum_{i=0}^{k}\|\mathtt{T}_i Q_i\| \leq V_{max}.$$

Therefore, the inequality holds for $k \geq 1$ by induction. Now the bound on $\|Q^* - Q_k\|$ follows $\|Q^* - Q_k\| \leq \|Q^*\| + \|Q_k\| \leq 2V_{max}$. $\qquad\square$

**Lemma 3.** Assume that initial action-value function, $Q_0$, is uniformly bounded by $V_{max}$, then, for any $k \geq 1$

$$\|Q^* - Q_k\| \leq \frac{2\beta V_{max}}{k(1-\beta)} + \frac{1}{k}\left\|\sum_{i=0}^{k-1} e_i\right\|. \tag{33}$$

*Proof.* From Lemma 1, we have

$$Q^* - Q_k = Q^* - \frac{1}{k}\left(\sum_{i=0}^{k-1} \mathrm{T}Q_i + \sum_{i=0}^{k-1} e_i\right) = \frac{1}{k}\sum_{i=0}^{k-1}(\mathrm{T}Q^* - \mathrm{T}Q_i) - \frac{1}{k}\sum_{i=0}^{k-1} e_i.$$

Therefore, we can write

$$\|Q^* - Q_k\| \leq \frac{1}{k}\left\|\sum_{i=0}^{k-1}(\mathrm{T}Q^* - \mathrm{T}Q_i)\right\| + \frac{1}{k}\left\|\sum_{i=0}^{k-1} e_i\right\| \leq \frac{1}{k}\sum_{i=0}^{k-1}\|\mathrm{T}Q^* - \mathrm{T}Q_i\| + \frac{1}{k}\left\|\sum_{i=0}^{k-1} e_i\right\|$$

$$\leq \frac{\beta}{k}\sum_{i=0}^{k-1}\|Q^* - Q_i\| + \frac{1}{k}\left\|\sum_{i=0}^{k-1} e_i\right\|.$$

and according to [44], $\|Q^* - Q_i\| \leq \beta^i\|Q^* - Q_0\|$. Hence, using Lemma 2, we can write

$$\|Q^* - Q_k\| \leq \frac{\beta}{k}\sum_{i=0}^{k-1} 2\beta^i V_{max} + \frac{1}{k}\left\|\sum_{i=0}^{k-1} e_i\right\| \leq \frac{2\beta V_{max}}{k(1-\beta)} + \frac{1}{k}\left\|\sum_{i=0}^{k-1} e_i\right\|.$$

$\square$

Now, we prove Proposition 1 by using the above result in Lemma 3. To this aim, we need to provide a bound on the norm of the summation of errors in the inequality of Lemma 3. First, we can write

$$\frac{1}{k}\left\|\sum_{i=0}^{k-1} e_i\right\| = \frac{1}{k}\max_{(x,a)\in\mathcal{Z}}\left|\sum_{i=0}^{k-1} e_i\right|.$$

For the estimation error sequence $\{e_0, e_1, \cdots, e_k\}$, we have the property that $\mathbb{E}[e_k|\mathcal{F}_{k-1}] = 0$ which means that the error sequence is a martingale difference sequence with respect to $\mathcal{F}_k$. Therefore, according to Hoeffding-Azuma inequality [45] for a martingale difference sequence of $\{e_0, e_1, \cdots, e_{k-1}\}$ which is bounded by $2V_{max}$, for any $t > 0$, we can write

$$\Pr\left(\left|\sum_{i=0}^{k-1} e_i\right| > t\right) \leq 2\exp\left(\frac{-t^2}{8kV_{max}^2}\right).$$

Therefore, by a union bound over the state-action space, we have

$$\Pr\left(\left\|\sum_{i=0}^{k-1} e_i\right\| > t\right) \leq 2|\mathcal{X}|.|\mathcal{A}|\exp\left(\frac{-t^2}{8kV_{max}^2}\right) = \delta,$$

and then,

$$\Pr \left( \frac{1}{k} \left\| \sum_{i=0}^{k-1} e_i \right\| \le V_{max} \sqrt{\frac{8}{k} \ln \frac{2|\mathcal{X}|.|\mathcal{A}|}{\delta}} \right) \ge 1 - \delta.$$

Hence, with probability at least $1 - \delta$ we can say

$$\|Q^* - Q_k\| \le \frac{2R_{max}}{(1-\beta)} \left[ \frac{\beta}{k(1-\beta)} + \sqrt{\frac{2}{k} \ln \frac{2|\mathcal{X}|.|\mathcal{A}|}{\delta}} \right].$$

Consequently, the result in Proposition 1 is proved. $\square$

## REFERENCES

[1] R. Amiri, H. Mehrpouyan, L. Fridman, R. K. Mallik, A. Nallanathan, and D. Matolak, "A machine learning approach for power allocation in HetNets considering QoS," *in Proc. IEEE ICC*, pp. 1–7, May 2018.

[2] O. G. Aliu, A. Imran, M. A. Imran, and B. Evans, "A survey of self organisation in future cellular networks," *IEEE Commun. Surv. Tutor.*, vol. 15, no. 1, pp. 336–361, First Quarter 2013.

[3] J. Moysen and L. Giupponi, "From 4G to 5G: Self-organized network management meets machine learning," *CoRR*, vol. abs/1707.09300, 2017. [Online]. Available: http://arxiv.org/abs/1707.09300

[4] J. G. Andrews, S. Buzzi, W. Choi, S. V. Hanly, A. Lozano, A. C. K. Soong, and J. C. Zhang, "What will 5G be?" *IEEE J. Select. Areas Commun.*, vol. 32, no. 6, pp. 1065–1082, June 2014.

[5] M. Peng, D. Liang, Y. Wei, J. Li, and H. Chen, "Self-configuration and self-optimization in LTE-advanced heterogeneous networks," *IEEE Commun. Mag.*, vol. 51, no. 5, pp. 36–45, May 2013.

[6] M. Agiwal, A. Roy, and N. Saxena, "Next generation 5G wireless networks: A comprehensive survey," *IEEE Commun. Surv. Tutor.*, vol. 18, no. 3, pp. 1617–1655, Thirdquarter 2016.

[7] P. V. Klaine, M. A. Imran, O. Onireti, and R. D. Souza, "A survey of machine learning techniques applied to self-organizing cellular networks," *IEEE Commun. Surv. Tutor.*, vol. 19, no. 4, pp. 2392–2431, Fourthquarter 2017.

[8] A. Imran, A. Zoha, and A. Abu-Dayya, "Challenges in 5G: how to empower SON with big data for enabling 5G," *IEEE Network*, vol. 28, no. 6, pp. 27–33, Nov 2014.

[9] R. Li, Z. Zhao, X. Zhou, G. Ding, Y. Chen, Z. Wang, and H. Zhang, "Intelligent 5G: When cellular networks meet artificial intelligence," *IEEE Wirel. Commun.*, vol. 24, no. 5, pp. 175–183, Oct 2017.

[10] V. Chandrasekhar, J. G. Andrews, T. Muharemovic, Z. Shen, and A. Gatherer, "Power control in two-tier femtocell networks," *IEEE Trans. Wireless Commun.*, vol. 8, no. 8, pp. 4316–4328, Aug 2009.

[11] Z. Lu, T. Bansal, and P. Sinha, "Achieving user-level fairness in open-access femtocell-based architecture," *IEEE Trans. Mobile Comput.*, vol. 12, no. 10, pp. 1943–1954, Oct 2013.

[12] H. Claussen, "Performance of macro- and co-channel femtocells in a hierarchical cell structure," *IEEE 18th Int. Symp. Pers. Indoor Mobile Radio Commun.*, pp. 1–5, Sep 2007.

[13] R. Amiri and H. Mehrpouyan, "Self-organizing mm-wave networks: A power allocation scheme based on machine learning," *in Proc. IEEE GSMM*, pp. 1–4, May 2018.

[14] Y. Sinan Nasir and D. Guo, "Deep reinforcement learning for distributed dynamic power allocation in wireless networks," *ArXiv e-prints*, Aug. 2018.

[15] D. Lopez-Perez, X. Chu, A. V. Vasilakos, and H. Claussen, "Power minimization based resource allocation for interference mitigation in OFDMA femtocell networks," *IEEE J. Select. Areas Commun.*, vol. 32, no. 2, pp. 333–344, Feb 2014.

[16] M. Yousefvand, T. Han, N. Ansari, and A. Khreishah, "Distributed energy-spectrum trading in green cognitive radio cellular networks," *IEEE Trans. Green Commun.*, vol. 1, no. 3, pp. 253–263, Sep 2017.

[17] H. Yazdani and A. Vosoughi, "On cognitive radio systems with directional antennas and imperfect spectrum sensing," *in Proc. IEEE ICASSP*, pp. 3589–3593, March 2017.

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

[19] A. Galindo-Serrano and L. Giupponi, "Distributed Q-learning for aggregated interference control in cognitive radio networks," *IEEE Trans. Veh. Technol.*, vol. 59, no. 4, pp. 1823–1834, May 2010.

[20] H. Saad, A. Mohamed, and T. ElBatt, "Distributed cooperative Q-learning for power allocation in cognitive femtocell networks," *in Proc. IEEE Veh. Technol. Conf.*, pp. 1–5, Sep 2012.

[21] J. R. Tefft and N. J. Kirsch, "A proximity-based Q-learning reward function for femtocell networks," *in Proc. IEEE Veh. Technol. Conf.*, pp. 1–5, Sep 2013.

[22] M. Bennis, S. M. Perlaza, P. Blasco, Z. Han, and H. V. Poor, "Self-organization in small cell networks: A reinforcement learning approach," *IEEE Trans. Wireless Commun.*, vol. 12, no. 7, pp. 3202–3212, July 2013.

[23] B. Wen, Z. Gao, L. Huang, Y. Tang, and H. Cai, "A Q-learning-based downlink resource scheduling method for capacity optimization in LTE femtocells," *in Proc. IEEE. Int. Comp. Sci. and Edu.*, pp. 625–628, Aug 2014.

[24] Z. Gao, B. Wen, L. Huang, C. Chen, and Z. Su, "Q-learning-based power control for LTE enterprise femtocell networks," *IEEE Syst. J.*, vol. 11, no. 4, pp. 2699–2707, Dec 2017.

[25] M. Miozzo, L. Giupponi, M. Rossi, and P. Dini, "Distributed Q-learning for energy harvesting heterogeneous networks," *in Proc. IEEE. ICCW*, pp. 2006–2011, June 2015.

[26] B. Hamdaoui, P. Venkatraman, and M. Guizani, "Opportunistic exploitation of bandwidth resources through reinforcement learning," *in Proc. IEEE GLOBECOM*, pp. 1–6, Nov 2009.

[27] G. Alnwaimi, S. Vahid, and K. Moessner, "Dynamic heterogeneous learning games for opportunistic access in LTE-based macro/femtocell deployments," *IEEE Trans. Wireless Commun.*, vol. 14, no. 4, pp. 2294–2308, April 2015.

[28] K.-L. A. Yau, P. Komisarczuk, and P. D. Teal, "Reinforcement learning for context awareness and intelligence in wireless networks: Review, new features and open issues," *J. Netw. Comput. Appli.*, vol. 35, no. 1, pp. 253 – 267, Jan 2012.

[29] C. J. C. H. Watkins and P. Dayan, "Q-learning," *Machine Learning*, vol. 8, no. 3, pp. 279–292, May 1992.

[30] L. Matignon, G. J. Laurent, and N. Le Fort-Piat, "Reward function and initial values: Better choices for accelerated goal-directed reinforcement learning," *in Proc. ICANN*, pp. 840–849, 2006.

[31] Z.-Q. Luo and W. Yu, "An introduction to convex optimization for communications and signal processing," *IEEE J. Select. Areas Commun.*, vol. 24, no. 8, pp. 1426–1438, Aug 2006.

[32] S. Niknam and B. Natarajan, "On the regimes in millimeter wave networks: Noise-limited or interference-limited?" *in Proc. IEEE ICCW*, pp. 1–6, May 2018.

[33] 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 Trans. Signal Processing*, vol. 66, no. 20, pp. 5438–5453, Oct 2018.

[34] C. Boutilier, T. L. Dean, and S. Hanks, "Decision-theoretic planning: Structural assumptions and computational leverage," *J. Art. Intel. Research*, vol. 11, pp. 1–94, July 1999.

[35] R. Amiri, H. Mehrpouyan, D. Matolak, and M. Elkashlan, "Joint Power Allocation in Interference-Limited Networks via Distributed Coordinated Learning," *in Proc. IEEE Veh. Technol. Conf.*, to be published. [Online]. Available: https://arxiv.org/abs/1806.02449

[36] Z. Lin and M. van der Schaar, "Autonomic and distributed joint routing and power control for delay-sensitive applications in multi-hop wireless networks," vol. 10, no. 1, pp. 102–113, Jan 2011.

[37] C. Guestrin, M. G. Lagoudakis, and R. Parr, "Coordinated reinforcement learning," *in Proc. ICML*, pp. 227–234, July 2002.

[38] E. Even-Dar and Y. Mansour, "Learning rates for Q-learning," *J. Mach. Learn. Research*, vol. 5, pp. 1–25, Dec 2004.

[39] L. Busoniu, R. B. ŝka, and B. D. Schutter, "A comprehensive survey of multiagent reinforcement learning," *IEEE Trans. Syst., Man, Cybern. C*, vol. 38, no. 2, pp. 156–172, March 2008.

[40] M. J. Kearns and S. P. Singh, "Finite-sample convergence rates for Q-learning and indirect algorithms," *NIPS*, vol. 11, pp. 996–1002, 1999.

[41] 3GPP, "Evolved Universal Terrestrial Radio Access (E-UTRA); Further advancements for E-UTRA physical layer aspects," 3rd Generation Partnership Project (3GPP), Technical Specification (TS) 36.814, 03 2010, version 9.0.0.

[42] C. C. Zarakovitis, Q. Ni, D. E. Skordoulis, and M. G. Hadjinicolaou, "Power-efficient cross-layer design for OFDMA systems with heterogeneous QoS, imperfect CSI, and outage considerations," *IEEE Trans. Veh. Technol.*, vol. 61, no. 2, pp. 781–798, Feb 2012.

[43] S. Niknam, R. Barazideh, and B. Natarajan, "Cross-layer Interference Modeling for 5G MmWave Networks in the Presence of Blockage," *ArXiv e-prints*, Jul. 2018.

[44] A. L. Strehl, L. Li, and M. L. Littman, "Reinforcement learning in finite MDPs: PAC analysis," *J. Mach. Learn. Res.*, vol. 10, pp. 2413–2444, Dec. 2009.

[45] W. Hoeffding, "Probability inequalities for sums of bounded random variables," *J. Am. Stat. Assoc*, vol. 58, no. 301, pp. 13–30, 1963.