Interference Mitigation in Femtocell Network Using Q-learning

Anonymous Author(s)
Affiliation
Address
email

Abstract

The idea of femtocell have attracted lots of interest from the research community of wireless communication. Femtocells aim to increase the capacity and coverage area of the current cellular network by helping to reduce its actual cell size. However, new design challenges arise by randomly deploying the femtocells over the cellular network, hence a heterogeneous network. One of the main problems is the so-called co-tier and cross-tier interference caused by the new femtocell network layer, which is operating in the same frequency spectrum as the cellular network. Various techniques have been proposed to deal with the interference management issue in heterogeneous networks. This paper investigates the use of reinforcement learning algorithms to solve interference problems in two tier heterogeneous networks. We assume the femtocell user equipments (FUEs) and macro user equipments (MUEs) to be selfish and we try to guarantee the Quality of Service (QoS) of all users equally if possible. We formulate the power adaptation process of the FUEs and MUE to be a discrete multi agent Markov decision problem and solve it by using the well-known Q-learning algorithm. In the MDP process, each agent adapts its own transmission power by learning from the environment. Numerical results show that the distributed decision process will converge to an equilibrium and make the system more efficient.

1 Introduction

The next generation network will be composed of several layers of networks with different service ranges. The current cellular network promises to provide network access anywhere and any time. But in practice every mobile user experienced some dead spots of the network. In future, femtocells with a coverage circle of several meters in radius, will be randomly and massively deployed over the traditional cellular network. The aim of the femtocells is to provide indoor mobile users a better wireless connection and save their limited battery energy. At the same time, femtocells deployment increases the overall network capacity, coverage area and throughput by reducing the distance between transmitters and receivers. However, cross-tier and co-tier signal interferences arise with the random deployment of femtocells because all the agents operate in the same frequency spectrum. Interference management is recognised as one of the key challenges in literature and it has attracted a lot of research efforts over the past few years. Centralized and distributed approaches are the two mainstreams in the past literature. Generally, the centralized approach requires frequent and heavy information exchange between the central controller and each mobile agent. With the random and large scale deployment of femtocells, centralized control is more difficult to be realized in practice. Therefore, more and more researchers focus on designing distributed and efficient interference management schemes for femtocell network.

In the literature, there are three access modes for macrocells and femtocells to share or rather compete spectrum, namely open, closed and hybrid access mode. In open access mode, the MUE can

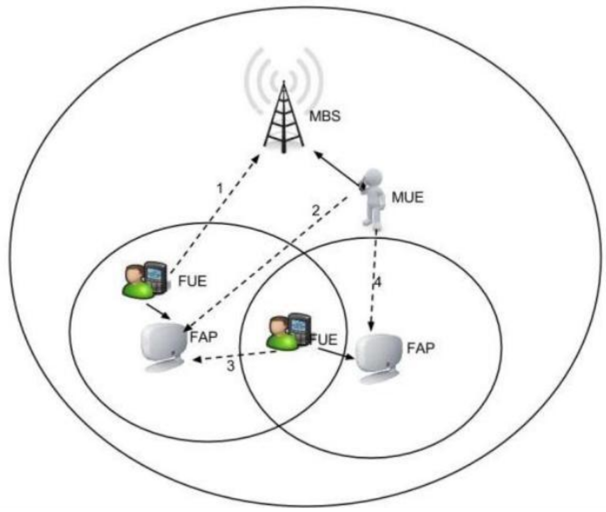

Figure 1: Illustration of a simplified femtocell network and the interference cases. The solid arrows are the designated communication links. The dashed arrows indicate interferences to the base stations in the uplink. Link 1,2,4 are cross-tier interferences and link 3 is co-tier interference.

connect to any Base Station (BS) it prefers. In closed access mode, the MUE is only allowed to connect to the designated Macro Base Station (MBS), it is not allowed to connect any Femtocell Access Point (FAP). In hybrid access mode, the FAPs will share some spectrum resources with the nearby MUEs based on some pre-designed sharing policy. One of the challenges is to design a good hybrid access scheme to balance between the QoS of the out coming MUEs and the local FUEs which belong to the designated FAP. A detailed analysis and comparison between open, closed and hybrid access is given in [12] and [13]. In this paper, the work is based on the closed access mode.

The two basic approaches to mitigate or avoid interferences in femtocell network are power control and channel allocation. A general coverage of the various previous interference mitigation methods can be found in [1]. A list of various interference scenarios and classification is also provided. In this paper we will focus on the learning methods proposed over the past few years. To the best of the writer's knowledge, there have been several papers applying reinforcement learning techniques to solve the interference problem over the past two years. It is natural and more popular now to assume the mobile users, both FUEs and MUEs, and the FAPs are selfish agents. They compete the limited spectrum resources and try to maximize their own data rate and save their own limited power. This is a well defined mixed task stochastic game as explained in [6]. Multi agent reinforcement learning algorithms are frequently employed to solve this kind of problem. In [2], Q-learning is used to solve the optimal downlink power allocation of the FAPs. The FAPs can select among a finite set of transmission power levels to maintain interference to the MUEs at a desired value. In [3], being aware of the slow convergence problem in Q-learning algorithm, a Q-learning initialization method is proposed to deal with the convergence speed. Both [2] and [3] assume there is an initial training phase to learn the optimal policy for power allocation, then the FAPs follow the learned policy to allocate power in all the channels. In [4], Q-learning is used to select the transmission channels for the FAPs to avoid interference to the MUEs. In [8], the authors assume open access mode and they try to solve the cell selection, also called handover problem by using Q-learning. In [9], the authors proposed a new reward function based on [2] and argued that the design of reward function will affect the convergence speed. A cooperative learning objective based on information communication among the learning agents is also provided in the paper. In [5], Q-learning, fictitious play and replication dynamics are compared to each other in terms of convergence speed. The authors concluded that better overall performance and faster convergence are achieved at the expense of more information exchange among learning agents.

In this paper, we study the power control problem of the FUEs in the uplink channel. All the papers mentioned above talk about how the FAPs allocate power to different downlink channels to either reduce or avoid interference to the MUEs. The power allocation policy is learned in the initial learning phase, once learned, the FAPs will follow the policy. However, in practice, the environment is changing all the time, the optimal policy will also change with the environment. In the uplink case, each FUE keeps adapting its own transmission power in consideration of its own channel condition, interference received from others and interference to others. In the closed access mode, it is not always possible to satisfy the target data rates for all users, especially when the co-channel user density is above some threshold. In the uplink case, the FUEs keeps interacting with the environment to maximize its own throughput. So the best policy is not a fixed set of levels of power, but a good power adaptation policy.

This paper is organized as follows. Section II sets up the system model used in the paper. Section III introduces the Q-learning algorithm and its application to our problem. Numerical results is analysed in section IV. And finally we give conclusions and future roadmap in section V.

2 System Model

In this paper, we consider a network with one MBS and N_f FAPs. The MUEs are randomly distributed in the cellular network and in the coverage area of a FAP, the FUEs associated with the FAP are uniformly distributed. A simplified version of the network is illustrated in Figure 1. We assume an OFDMA system. But we do not consider the channel selection problem, we deal with the power control problem in each single sub-channel separately. At each time slot, in each FAP, only one FUE is allowed to transmit. So within each FAP, it is orthogonal TDMA scheduling.

Denote the maximum transmission power of the FUE as P^f_{max} and the maximum transmission power of the MUE as P^m_{max}. Denote the target signal to interference noise ratio (SINR) of the MUE as γ^m_T and the target SINR of the FUE as γ^f_T. The target SINR is the minimum SINR required by the mobile user for reliable data transmission. Mobile users will intelligently adjust transmission power by learning from the environment to satisfy the SINR requirement and save energy at the same time.

The instantaneous SINR of FUE i, which is associated with a designated FAP, in a certain sub-channel can be written as

$$\gamma_i^f = \frac{p_i^f g_i^f}{\sigma^2 + \sum_{j\in I_m} p_j^m g_j^m + \sum_{j\in I_f} p_j^f g_j^f} \tag{1}$$

where p^f_k and p^m_k denote the transmission power of FUE k and MUE k respectively. g^f_k denotes the channel gain between FUE k and the designated FAP. g^m_k denotes the channel gain between MUE k and the designated FAP. σ^2 is the channel noise power. I_f is the set of all the interfering MUEs transmitting in the same sub-channel. I_m is the set of all the interfering FUEs in the same sub-channel.

Similarly, the SINR of MUE i, which associated with the MBS in a certain sub-channel can be written as

$$\gamma_i^m = \frac{p_i^m h_i^m}{\sigma^2 + \sum_{j\in I_m} p_j^m h_j^m + \sum_{j\in I_f} p_j^f h_j^f} \tag{2}$$

where all the channel gains h are between the corresponding mobile user and the MBS. The throughput of the channel can be calculated as follows:

$$c = log_2(1+\gamma) \tag{3}$$

3 Application of Reinforcement Learning

3.1 Reinforcement learning

In this part, we will introduce the Q-learning model. Q-learning is one of the many algorithms to solve the discrete Markov decision problem (MDP). In discrete MDP, the system is modelled as a Markov chain and the system state jumps randomly from one state to another state in discrete time steps [11]. Formally, a finite state and action spaces single agent MDP can be defined as a tuple (S, A, f, r), where S is a finite set of environment states, A is a finite set of agent actions, $f : S \times A \times S \to [0, 1]$ is the state transition probability function, and $r : S \times A \times S \to \mathbf{R}$ is the reward function [6]. Denote s_k and a_k as the state of the system and action of the agent at discrete time step k respectively. The immediate reward received by the agent after taking action a_k and the resulting state transition from s_k to s_{k+1} can be written as $r_{k+1} = r(s_k, a_k, s_{k+1})$. A decision policy is defined as $\pi : S \rightarrow A$, which tells the agent which action to choose given the current state. The aim of the agent is to find an optimal policy π^* to maximize some performance metric.

The expected discounted reward given the policy π and initial state s is given by:

$$V^{\pi}(s) = E_f \left\{ \sum_{j=0}^{\infty} \gamma^j r(s_j, \pi(s_j), s_{j+1}) | s_0 = s \right\} \tag{4}$$

where $\gamma \in [0, 1)$ is called the discount factor. It is used to bound the summation and can be interpreted as we have more uncertainty in the future reward. $\pi(s_j)$ denotes the action taken at state s_j. $V^{\pi}(s)$ is also called the value function of state s given policy π. Notice that the expectation is taken over the probabilistic transition function. In deterministic model, the transition probability function is fixed to a specific transition function and the expectation can be saved. Another way to write the value function of state s is

$$V^{\pi}(s) = E_f \left\{ r(s, \pi(s), s') \right\} + \gamma \sum_{s' \in S} f(s, \pi(s), s') V^{\pi}(s') \tag{5}$$

This is also called Bellman equation. The first term is the expected immediate reward we get after taking action $\pi(s)$ in state s. The second term is the expected sum of the discounted rewards starting in state s', where s' is the next state after state s and follows the distribution given by the transition probability function f.

In Q-learning, the action-value function, also the Q-value is defined as the expected return of a state-action pair given some policy π: $Q^{\pi}(s,a) = E\left\{\sum_{j=0}^{\infty} \gamma^j r_{k+j+1} | s_k = s, a_k = a, \pi\right\}$. The optimal Q-value is defined as $Q^*(s,a) = max_{\pi} Q^{\pi}(s,a)$ Given all the above definitions, the Bellman optimality equation can be written as

$$Q^*(s,a) = E\{r(s,a,s')\} + max_{b \in A} \sum_{s' \in S} \gamma f(s, \pi(s), s') Q^*(s', b) \tag{6}$$

To compute the optimal Q-value for each state-action pair, Q-learning algorithm estimates the optimal Q-value by an iteration approximation procedure. The update equation is

$$Q(s,a) = (1-\mu)Q(s,a) + \mu[r(s,a,s') + \gamma max_{b \in A} Q(s', b)] \tag{7}$$

where $\mu \in (0, 1]$ is the learning rate. The learning rate is typically time varying and decreases with time. From the update equation we can see that the Q-learning method is model-free. It does not require any prior knowledge about the state transition probability or reward function. Both of them are acquired on-line in simulation. One of the conditions to guarantee the convergence of Q-learning is the agent has to keep trying all actions in all states with non-zero probability [6]. To satisfy this condition, the ϵ-greedy exploration procedure is incorporated into the Q-learning algorithm. That is, in each iteration, the agent chooses a random action with probability $\epsilon \in (0, 1)$ and chooses the greedy action that will maximizes the Q-value with probability $1 - \epsilon$.

3.2 Problem Formulation

In this paper, we assume there is no information exchange among the learning agents. However, we assume the FUEs and MUE will get their SINR information from the corresponding BSs instantaneously through some feedback channel. The learning agents, actions, states and reward functions are designed and explained as follows:

- Agents: The learning agents are the FUEs and the MUE associated with the only MBS in the considered channel. If there are N_f FAPs, there are $N_f + 1$ learning agents in the system. They adapt their transmission powers by learning from the environment independently. However, their action will inevitably affect each other's channel condition, i.e. the SINR parameter as given in equation (1) and (2) . These agents will learn to reach the optimal equilibrium if there exists one for any given simulation scenario. We assume equal importance of all the users in the system in this paper. As we are discussing the uplink communication channel, the main interference is actually from MUE to nearby FUE and neighbour FUEs' co-tier interference, thus we do not give the MUE any priority over any FUE. Because this is a closed access mode system, when the number of learning agents increases, some users may not be able to satisfy their target SINRs or data rates.

- Actions: There are three actions for each agent in almost any state. To increase the transmission power, to keep it or to decrease it. $A_{i,t} = A = \{0$: decrease tranmission power; 1 : keep current transmission power; 2 : increase transmission power.$\forall i\}$. Two boundary cases are: *1*) when the agent tries to decrease its power to negative value, we floor its transmission power by zero. That is, the agent chooses to keep silent in certain kinds of states. *2*) when the agent tries to increase its power above the maximum transmission power, we ceil its power by P^f_{max} or P^m_{max}. These two boundary special treatments are necessary as validated in simulations. If we allow the agents to choose any transmission power they prefer, the power competition will increase with iterations and it never falls back to the normal state. This is due to the selfish nature of the learning agents. No agent will start first to decrease its own transmission power to suffer the lower transmission rates. The step size of the power change will be adjusted in the simulation and has significant effect on the convergence and behaviour of the final learning curve. Contrary to most existing paper, in which a fixed set of transmission power levels are pre-determined to be selected by all the agents, our scheme allows more freedom in the agent's action. This design also results in a different pattern of convergence compared to the existing papers and this will be further explained in the simulation section.
- States: The state of learning agent i at time t is represented as a tuple of three indicators: $s_{i,t}$ = $\{I^i_{\gamma,t}, I^i_{p,t}, a^i_{t-1}\}$. Here a^i_{t-1} is the action performed by agent i at discrete time step or iteration $t-1$, the last time action. $I^i_{\gamma,t}$ and $I^i_{p,t}$ are defined as follows:

$$I^i_{\gamma,t} = \begin{cases} 1 & \text{if } \gamma_i \geq \gamma^i_T \\ 0 & \text{if } \gamma_i < \gamma^i_T \end{cases} \tag{8}$$

$$I^i_{p,t} = \begin{cases} 1 & \text{if } p_i \geq P^i_{max} \\ 0 & \text{if } p_i < P^i_{max} \end{cases} \tag{9}$$

  Since there is no information exchange among learning agents, an agent can only monitor and learn from its own past actions, transmission power and SINR statistics. Initially, I planned to monitor a five element tuple, including the last time SINR and last time transmission power besides the given three elements. But due to time limit, I make it simplified now. Basically, the more states and actions, the more need to worry about the convergence issue in simulation.
- Rewards: Finally the reward function is defined as follows:

$$r_{i,t} = \begin{cases} ln(\frac{c_{i,t}}{c_T}) + \exp(-\frac{p_{i,t}}{P^i_{max}}) & \text{if } 0 \leq p_{i,t} \leq P^i_{max} \\ -3 & \text{if otherwise} \end{cases} \tag{10}$$

  The reward function is designed in order to encourage maximum date rates and relatively efficient transmission power. All the other states will be punished. The design of reward function is quite flexible and parameter can be tuned in simulation. The Q-learning algorithm used in the simulation is given in algorithm 1.

4 Numerical Results

In this section we present some numerical results from our simulation. The learning rate is $\alpha = 90/(100+iteration)$. Some learning parameters are given in the figure caption. When you read the figures shown below, you may wonder why the channel capacity curve does not converge to a strict line. It instead oscillates in a small interval. This is due to my design of the agent action. The agent learns to increase or decrease or remain the transmission power to keep the equilibrium with the outside environment. The agent is not fixed to a certain power level as in existing papers. It learns to dynamically adjust its power and track the environment, i.e. the other agents. Actually, these agents learn by tracking the others and their aim is to track the environment to adjust transmission power.

Algorithm 1 Distributed power control

```
procedure Q-LEARNING
    Initialize all Q-values to 0. Q_i(a, s) = 0, ∀ a ∈ A, s ∈ S, i.
    Initialize the states s_{i,0} = (γ_{i,0}, p_{i,0}, a_{i,0}), ∀i
    while iteration do
        for all i do
            if rand() < ε then
                choose a random action available in current state s_{i,t}
            else
                choose the greedy action a_{i,t} = arg max_{a_{i,t}} Q(s_{i,t}, a_{i,t})
            end if
        end for
        Perform selected action and obtain next state S = {s_{i,t+1}, ∀i}
        for all i do
            update Q_i(s_{i,t}, a) := (1 − α)Q(s_{i,t}, a) + α[r + γ max_b Q(s_{i,t+1}, b)]
        end for
    end while
end procedure
```

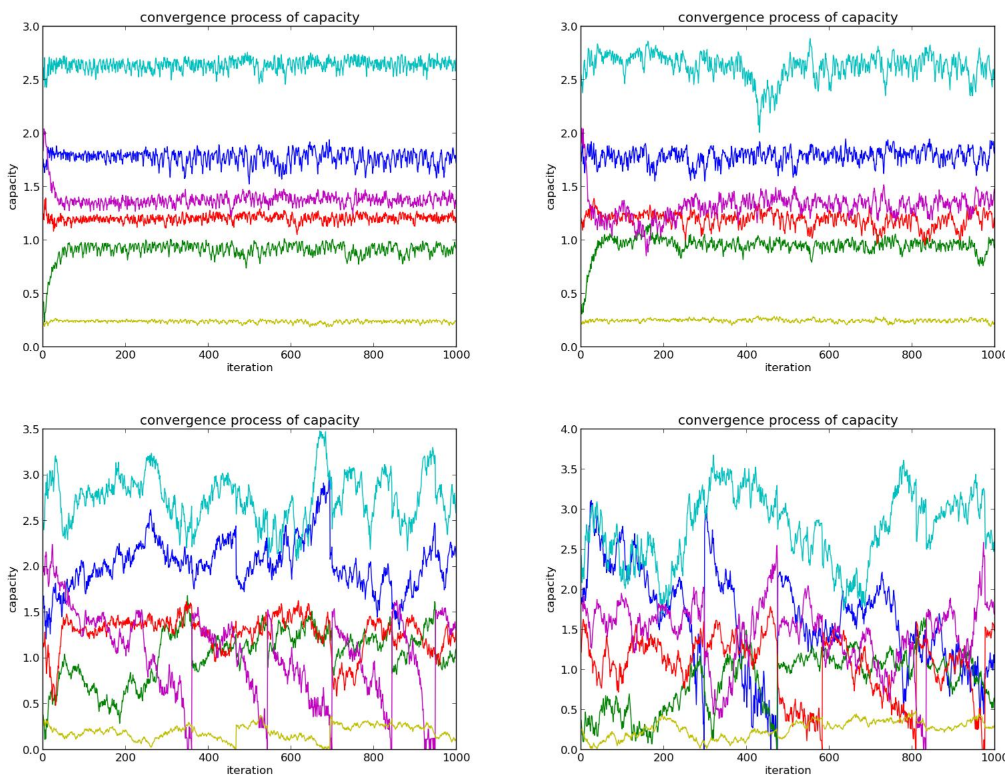

Figure 2: Illustration of the effect of exploration rate ϵ. As ϵ increases, it takes more iterations to converge. 1000 iterations, 6 learning agents, $\epsilon = 0.1, 0.2, 0, 7, 0, 9$ from left to right, from top to down side.

So finally, when they learned the policy, their transmission power, as a result the capacity will still oscillate in a small interval, instead of keeping a constant power level.Figure (2) shows that as the exploration parameter increases, the learning takes a longer time.

Figure (3) illustrates a common phenomenon in the closed access femtocell network. All the other curves except the blue curve have converged to some policy. Now the blue agent is a dominant

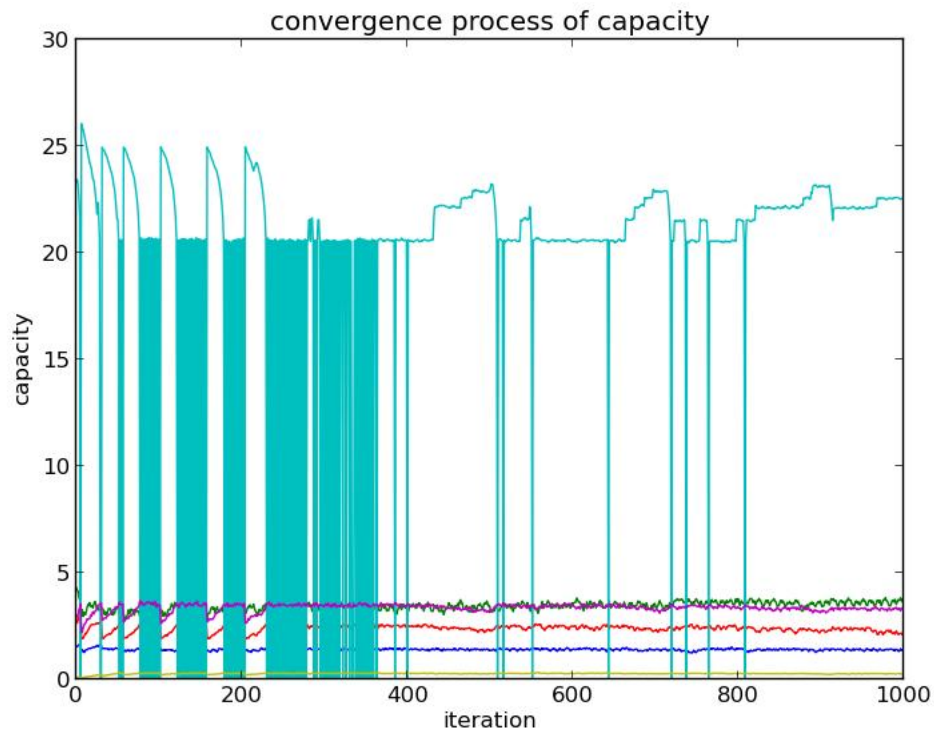

Figure 3: Who is playing and who is learning?

interference source in the network. By observation, on can find that the other curves are tracking the high variation blue curve. Whenever the blue curve goes down, the others' throughputs are increased. But since the agent is selfish, it has no incentive to cooperate with others in our formulation. It has a good channel condition and the other agents have less interference to it. So the blue agent can choose any action at any state. All the other users have to maintain a relatively high transmission power to combat the interference from the blue agent. The blue agent is the MUE. In this case, the femtocell network users suffer because of the nearby MUE.

5 Conclusions

In this paper, we introduced Q-learning algorithm to solve the power control problem in uplink closed access femtocell network. The simulation results show that most of the time, the FUEs will suffer from the nearby interfering MUE. Although the FUEs can learn to converge to an equilibrium, the system is far from optimal. Future work will investigate the use of learning algorithm in open access and hybrid access femtocell network. One direction is to learn a joint cell selection and power control policy.

References

[1] Nazmus Saquib et al., Interference Management in OFDMA Femtocell Networks: Issues and Approaches, Wireless Communications, IEEE 19, no. 3 (2012): 8695.

[2]A. Galindo-Serrano and L. Giupponi, Distributed Q-learning for interference control in OFDMA-based femtocell networks, presented at the Vehicular Technology Conference (VTC 2010-Spring), 2010 IEEE 71st, 2010, pp. 15.

[3]M. Simsek, A. Czylwik, A. Galindo-Serrano, and L. Giupponi, Improved decentralized Q-learning algorithm for interference reduction in LTE-femtocells, presented at the Wireless Advanced (WiAd), 2011, 2011, pp. 138143.

[4]M. Bennis and D. Niyato, A Q-learning based approach to interference avoidance in self-organized femtocell networks, presented at the GLOBECOM Workshops (GC Wkshps), 2010 IEEE, 2010, pp. 706710.

[5]M. Bennis, S. Guruacharya, and D. Niyato, Distributed Learning Strategies for Interference Mitigation in Femtocell Networks, presented at the Global Telecommunications Conference (GLOBECOM 2011), 2011 IEEE, 2011, pp. 15.

[6]L. Busoniu, R. Babuska, and B. De Schutter, A comprehensive survey of multiagent reinforcement learning, Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on, vol. 38, no. 2, pp. 156172, 2008.

[7]M. Nazir, M. Bennis, K. Ghaboosi, A. B. MacKenzie, and M. Latva-aho, Learning based mechanisms for interference mitigation in self-organized femtocell networks, presented at the Signals, Systems and Computers (ASILOMAR), 2010 Conference Record of the Forty Fourth Asilomar Conference on, 2010, pp. 18861890.

[8]C. Dhahri and T. Ohtsuki, Learning-Based Cell Selection Method for Femtocell Networks, presented at the Vehicular Technology Conference (VTC Spring), 2012 IEEE 75th, 2012, pp. 15.

[9]H. Saad, A. Mohamed, and T. ElBatt, Distributed Cooperative Q-learning for Power Allocation in Cognitive Femtocell Networks, arXiv preprint arXiv:1203.3935, 2012.

[10]A. Gosavi, A Tutorial for Reinforcement Learning, The State University of New York at Buffalo, 2011.

[11]A. Gosavi, Reinforcement learning: a tutorial survey and recent advances, INFORMS Journal on Computing, vol. 21, no. 2, pp. 178192, 2009.

[12]H.-S. Jo, P. Xia, and J. G. Andrews, Open, closed, and shared access femtocells in the downlink, EURASIP Journal on Wireless Communications and Networking, vol. 2012, no. 1, pp. 116, 2012.

[13]P. Xia, V. Chandrasekhar, and J. G. Andrews, Open vs. closed access femtocells in the uplink, Wireless Communications, IEEE Transactions on, vol. 9, no. 12, pp. 37983809, 2010.