Self-Optimization of Coverage and Capacity in LTE Networks Based on Central Control and Decentralized Fuzzy Q-Learning

Jingyu Li, Jie Zeng, Xin Su, Wei Luo, and Jing Wang

Tsinghua National Laboratory for Information Science and Technology, Tsinghua University, Beijing 100084, China

Correspondence should be addressed to Jie Zeng, zengjie@tsinghua.edu.cn

Received 4 May 2012; Accepted 18 July 2012

Academic Editor: Yiqing Zhou

Copyright © 2012 Jingyu Li et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

To reduce capital expenditures (CAPEX) and operational expenditures (OPEX) in network operations, self-organizing network (SON) has been introduced as a key part of long-term-evolution (LTE) system. Self-optimization of coverage and capacity is one of the most important tasks in the context of SON. This paper proposes a central control mechanism that utilizes the fuzzy Q-learning algorithm in a decentralized fashion for this task. In our proposed approach, each eNB is a learning agent that tries to optimize its antenna downtilt automatically using information from its own and its neighboring cells, and the initialization and the termination of the optimization processes of all agents are in the control of the central entity. The simulation results verify that our proposed approach can achieve remarkable performance enhancement as well as fast convergence, indicating that it is able to meet different levels of demands defined by 3GPP for coverage and capacity optimization.

1. Introduction

In order to achieve higher network performance and flexibility, while reducing capital expenditures and operational expenditures, 3GPP (Third Generation Partnership Project) introduces the concept of SON in [1]. In the context of SON, LTE systems have self-organizing capabilities of automating the configuration and optimization of the wireless network by introducing functionalities of self-configuring, self-optimizing, and self-healing [2, 3].

The research and development of online algorithms are vital parts in self-optimization process due to the challenges existing in realistic networks. Usually there is no definite mapping function from the input vector and adjustment parameters to the optimization objective, and thus it is difficult to solve the optimization problem directly. Besides, the input space and the parameter space can be very large, thereby full search algorithms are computationally prohibited. Furthermore, the information about the input may be incomplete and partly incorrect, which may cause negative impact on decision making.

The simulated annealing algorithm may be one of the candidates for solving the above optimization problem. It is a probabilistic heuristic searching solution to combinatorial optimization problems by simulating the physical process of annealing where a substance is gradually cooled to reach a minimum-energy state [4]. It has been widely applied in the network planning and optimization. For example, Siomina et al. [5] adopts the simulated annealing as an automated optimization engine with the aim of minimizing CPICH power by adjusting antenna parameters. However, it achieves optimization of service coverage in the offline mode with hypothetical network settings, and it does not run in the SON entity, a realistic node with self-optimizing functionalities in the network. The idea of simulated annealing algorithms is applied to the coverage and capacity optimization task in the context of SON by Cai et al. [6]. In his study, the optimization is achieved by controlling downlink transmit power with the Annealed Gibbs Sampling method in a decentralized SON manner. His approach relies on the accurate information about the interferences the evolved Node B (eNB) will bring to its neighboring cells at its all transmit power levels. However, as for adjusting antenna parameters, such as downtilt angles, this application of simulated annealing algorithms is limited, since such information cannot be predicted due to the ill-defined influence...
of changing downtilt angles on users’ received signal-to-interference-and-noise ratio (SINR). Admittedly, simulated annealing algorithms can overcome these difficulties if they are utilized in a centralized way, in which the central entity runs the algorithm to search for the global optimal solution. Nevertheless, the change interval (i.e., the time needed between two successive iterations) has to be relatively large due to long operating time for collecting the global information.

In this paper, we apply Q-Learning technique in a decentralized manner for the joint optimization of coverage and capacity task. Q-Learning is a practical form of Reinforcement Learning (RL) which is an important subfield of machine learning. RL is a type of learning involving an agent learning behavior that achieves a goal by directly interacting with its uncertain environment and by properly utilizing past experience derived from previous actions [7]. And we combine the fuzzy rule with the Q-Learning to deal with the realistic problem whose input and output variables are continuous. In addition, we introduce the central control mechanism which is responsible for the initialization and the termination of the optimization process of every learning agent deployed in every eNB.

The rest of the paper is outlined as follows. In Section 2, we describe the SON concept including its basic framework, use cases, and architectures. In Section 3, we discuss about the implementation of our approach with the hybrid architecture for the coverage and capacity optimization task. In Section 4, we present the details of the learning technique applied into our approach with the decentralized multiagent manner. In Sections 5 and 6, we show the simulation setting and the corresponding results. In Section 7, we conclude the paper.

2. SON Concept

2.1. Basic Framework and Use Cases. Aiming at reducing CAPEX and OPEX, self-organizing functions have highly automatic features requiring minimum manual intervention. It consists of self-configuring, self-optimizing, and self-healing functionalities. Figure 1 depicts a basic framework for these functionalities [8].

Self-configuration provides the capabilities for newly deployed eNBs to finish the configuration with automatic installation procedures for obtaining the basic configuration information to operate the system. This process works in preoperational state, which is known as the state from when the eNB is powered up and has backbone connectivity until the RF transmitter is switched on. The use cases for self-configuration include automatic configuration of physical cell identity, neighbor-list configuration, and coverage/capacity-related parameters [2].

Self-optimization offers the benefits of the dynamic optimization in the operational state, which is known as the state where the RF interface is switched on, as shown in Figure 1. This functionality reduces the workload for site visit and analysis of network performance manually, resulting in reduction of OPEX. The main use cases for self-optimization are coverage and capacity optimization, energy savings, mobility robustness optimization, mobility load balancing optimization, RACH optimization and interference reduction [2].

The introduction of self-healing process is to solve or mitigate the faults that could be solved automatically by activating proper recovery actions in the operational state. It includes not only the automatic detection as well as localization of the failure, but also the recovery and the compensation actions. The use cases of self-healing are cell outage detection, cell-outage compensation, cell outage recovery, and return-from-cell-outage compensation [3].

2.2. SON Architecture. Regarding the SON entity allocation, there are three basic ways to implement the SON architecture, namely centralized SON, decentralized SON and hybrid SON, as shown in Figure 2 [9].

In the centralized architecture, the SON entity (i.e., the optimization algorithms) resides on the network management system (NMS) or a central SON server that manages all the eNBs. With the global information from all the eNBs, centralized architecture can readily approach the global optimum. However, the updates of parameters are rather slow, and the optimization process would cost much time because of the large operating interval to collect all the information from all the eNBs who also need to receive the measurements from their UEs (user equipments). Therefore, the applicability of many effective algorithms with slow response time is restricted, that is why the purely centralized architecture does not gain much support from the vendors.

In the decentralized architecture, each eNB has a SON entity, which means each eNB runs the optimization process and gives the decision autonomously based on its UEs’ feedback information as well as other information from some other eNBs via the X2 interface. This decentralized architecture allows for ease of deployment and optimization on faster time scale. However, with such decentralized manner, there may be challenges on the stability of the network as well as the overall optimization performance.

The hybrid architecture, as its name indicates, is a combination of the centralized and the decentralized SON, which means part of a SON entity is located in the eNB, while the other is deployed in the NMS. Thus, it provides a tradeoff between the two architectures. For example, the selection of eNBs participating in the optimization, the initial parameter setting, and the objective setting is done in the central NMS, while the core optimization algorithm runs within the involved eNB.

Each architecture has its own advantages and disadvantages on implementation. Which is the most suitable approach depends on the requirement of the use case, the optimization objective, the infrastructure deployment, and other specific conditions.

3. Joint Optimization of Coverage and Capacity under Hybrid Architecture

Optimizing coverage and capacity of the network is one of the typical operational tasks of SON. The automating of
this task can minimize human intervention in the network by discovering the coverage and capacity problems through the eNB and UE measurements and by optimizing RF parameters through intelligent algorithms automatically. Its objectives include coverage optimization, capacity optimization, and joint optimization of coverage and capacity. The challenges of joint optimization of coverage and capacity include the tradeoff between coverage and capacity, the SON architecture implementation, the selection of indicators to observe and parameters to optimize, and so forth.

3.1. The Hybrid Architecture. For this task, the coverage and/or capacity problem is usually regional, and thus, only one or two cells are usually involved in the optimization; even if more than two cells are involved, antenna-parameter adjustment causes only local impact, which means only neighbors’ information is required to ensure stability and convergence. Hence, it is possible to run the algorithm in a decentralized manner to achieve the optimization. In addition, requiring only the information from its own UEs and from its neighboring cells, such manner can provide fast timescale of operation, as discussed in Section 2.2.

On the other hand, the capacity and coverage problem detection, the selection of eNBs, when to start/stop their optimization, and initial parameters of the algorithm need...
to be chosen in a central entity that can obtain the global information.

Based on the above considerations, we use hybrid architecture in our approach, which means we implement the control functionality including optimization activation/deactivation and initial parameter setting in a central entity, and deploy the optimization algorithm in every eNB in a decentralized manner.

3.2. Indicators. For self-optimization tasks, many indicators for observation, such as UE signaling reports, pilot-strength measurements and traffic counters, are recommended in [2].

The spectral efficiency (SE), referring to the information rate that can be transmitted per unit bandwidth, is a frequently used metric for evaluating system performance. We can use the CDF of 50% tile of the spectral efficiency distribution to denote the system capacity [10]. For the system coverage, considering that the users on the cell edge experience significantly poorer performance and greater probability of outage than the others in the cell, their performance metric can be reflected by the CDF of 5%-tile of the spectral efficiency [11]. Given that the spectral efficiency can be easily predicted by the RSRQ information that each eNB collects from its active users, RSRQ is a good choice as the indicator for the joint optimization of coverage and capacity.

To balance coverage and capacity, we define a joint performance metric (JPM) as the weighted sum of the spectral efficiency CDF of 50% tile and 5% tile by a fixed factor $\lambda$ ($0 \leq \lambda \leq 1$)

$$JPM = (1 - \lambda)SE_{50\%} + \lambda SE_{5\%}. \quad (1)$$

In addition, to take into account the impact of interference from neighboring cells, the optimization processes should be performed jointly. Hence, we use the key performance indicator (KPI) defined in formula (2) as the optimization objective of each eNB (cell).

$$KPI_i = w_i \cdot JPM_i + \frac{(1 - w_i)}{|N(i)|} \sum_{j \in N(i)} JPM_j, \quad (2)$$

where $JPM_i$ is the joint performance metric for cell $j$, $w_i$ is the weight for cell $i$, and $N(i)$ is the set of its neighbors. Each cell calculates its KPI with its own and its neighbors’ JPM by formula (2).

3.3. Outputs. The outputs, which are the parameters to adjust during the optimization process, could be the transmit power, antenna azimuth, antenna downtilt, and so forth.

In the process of self-optimization, each eNB may tend to increase transmit power in order to improve its capacity and coverage, but it would cause serious interference to its neighboring cells, thereby making little improvement for the overall system performance. In addition, antenna azimuth is one of the antenna configuration parameters which have great impacts on the cell overlap, and adjusting it can reduce cell overlap significantly, thereby improving the performance of the edge users. However, adjusting antenna azimuth has to be done mechanically, which would require costly site visits and significant time consumption. Comparatively, antenna downtilt is a better candidate for self-optimization of coverage and capacity. On one hand, signal power levels from the home cell can be improved with interference to the neighboring cells being effectively reduced by increasing the downtilt angle; on the other hand, coverage problems on the cell edge can be alleviated by decreasing the downtilt angle. Thus, we can make a tradeoff between the coverage and capacity by adjusting the downtilt angle. And fortunately, the advances in electrical downtilting enable the automated optimization task without any costly site visits by utilizing remote electrical downtilt (RET) controllers [12]. Therefore, we choose antenna downtilt as the parameter to be optimized in this paper.

3.4. Process Flow. The flow chart of the decentralized optimization approach is shown in Figure 3. After mapping RSRQ to SE, the two statistics, that is, $SE_{50\%}$ and $SE_{5\%}$, are both obtained. With its own and its neighbors’ SE statistics, JPM is calculated by formula (1), and KPI is calculated by formula (2). The optimization algorithm tries to search for a better downtilt configuration with a certain learning technique based on the current KPI as well as some previous KPIs.

4. Learning Technique

Q-learning [7] involves an agent with the learning behavior of achieving a goal by interacting with its uncertain environment (i.e., performs an action and receives rewards from the outside) and by learning from the past experience. In the Q-learning algorithm, there is a set of states and a set of actions. Each time the agent selects one action from the action set according to a quality function which is the quality metric of a state-action pair. After performing the action, the agent moves into a next state, and obtains a reward from the environment, with which it updates its quality function. The goal of the agent is to maximize the total reward.

Fuzzy logic is usually introduced in RL algorithms as an approximate approach when realistic problems have input and output variables with a large or infinite number of possible values (e.g., continuous downtilt) [13].

In this paper, we propose the Fuzzy Q-Learning (FQL) algorithm which combines the fuzzy logic with QL algorithm in the decentralized manner where multiple agents aim at achieving their own optimal goal by interacting with the environment. And the factor considering the performance of the neighboring cells is introduced in the goal.

4.1. FQL Algorithm. Let $X$ be the state space, and $A$ be the action space. $r(x,a)$ denotes the reward received at the state $x \in X$ with the action $a \in A$ performed. The objective of an agent is to find an optimal policy $\pi^*(x)$ for the state $x$ to maximize the utility function $R$ which is defined as a long-term sum of discounted rewards.

$$R = \mathbb{E}_\pi \left[ \sum_{t=0}^{\infty} \gamma^t r(x_t, a_t) \right], \quad (3)$$
where \( x_t \) and \( a_t \) denote the state and the action of the agent at step \( t (t = 0, 1, 2, \ldots) \), respectively, and \( y \) is the discount factor.

The solution of this maximization problem uses the quality function \( Q^\pi(x, a) \) (\( x \in X, a \in A \)) that is defined as the expected sum of discounted rewards from the initial state \( x_0 \) under policy \( \pi \) as follows:

\[
Q^\pi(x_0, a_0) = E_{\pi} \left[ \sum_{t=0}^{\infty} y^t r(x_t, a_t) \mid x = x_0 \right].
\]  

The low discount factor means immediate rewards are optimized, while the high one counts future rewards more strongly.

The Q-learning algorithm solves this with the Temporal Difference (TD) scheme by updating the quality function iteratively, as indicated in the following equation:

\[
Q_{t+1}(x_t, a_t) = Q_t(x_t, a_t) + \xi \left( r_{t+1} + y \max_{a'} Q_t(x_{t+1}, a') - Q_t(x_t, a_t) \right),
\]  

where \( \xi \) is the learning rate. With higher learning rate, the agent would learn faster, taking shorter time to achieve the optimal, but it might violate the convergence condition if the learning rate is too high.

The introduction of fuzzy inference systems (FIS) enables the possibility to treat continuous state and action spaces. An FIS is described by fuzzy rules as follows [13]:

\[
\begin{align*}
\text{if } x \text{ is } S_i & \quad \text{then } y = a[i,1] \quad \text{with } q[i,1], \\
\text{or } y = a[i,2] & \quad \text{with } q[i,2], \\
& \quad \ldots \ldots \\
\text{or } y = a[i,J] & \quad \text{with } q[i,J],
\end{align*}
\]

where \( S_i \) is the modal value corresponding to rule \( i (i = 1, 2, \ldots, J) \), and \( (a[i,j])_{j=1}^J \) are potential actions whose quality values \( (q[i,j])_{j=1}^J \) are initialized to zero.

For every rule \( i \), let \( k(i) \in [1 : J] \) be the subscript of the rule action chosen by an exploration/exploitation policy (EEP) using \( \epsilon \)-greedy method as follows:

\[
k(i) = \begin{cases} 
\text{random}(j), & \text{with prob. } 1 - \epsilon, \\
\arg \max_{j \in [1 : J]} q(i, j), & \text{with prob. } \epsilon,
\end{cases}
\]

where \( \epsilon \) is the probability of taking a nongreedy action, less than but close to 1. It is a tradeoff factor between exploration and exploitation.

The inferred action \( a(x) \) for input vector \( x \) and its quality are given by

\[
a(x) = \sum_{i=1}^{J} a[i,k(i)] \alpha_i(x),
\]

\[
Q(x, a(x)) = \sum_{i=1}^{J} q[i,k(i)] \alpha_i(x),
\]

where \( \alpha_i(x) \), the degree of truth in the FIS for rule \( i \), is defined by certain membership function which must satisfy \( \alpha_i(S_i) = 1 \) and \( \alpha_j(S_i) = 0 \) for \( j \neq i \).

At time step \( t \), after performing action \( a(x_t) \), the state becomes \( x_{t+1} \). The value of this state is defined as

\[
V_t(x_{t+1}) = \sum_{i=1}^{J} \max_k q[i,k] \alpha_i(x_{t+1}).
\]

The incremental quantity \( \Delta Q \) can be calculated by

\[
\Delta Q = r_t + yV_t(x_{t+1}) - Q(x_t, a(x_t)),
\]

and the elementary quality \( q[i,j] \) can be updated by the quantity

\[
\Delta q[i,j] = \begin{cases} 
\xi \cdot \Delta Q \cdot \alpha_i(x_{t+1}), & \text{if } j = k[i], \\
0, & \text{otherwise},
\end{cases}
\]

The detail of the FQL algorithm is presented in Table 1.
(1) $t = 0$, $q[i,j] = 0$ for all $i \in [1:I]$ and $j \in [1:J]$, and observe the state $x_t$.
(2) For every rule $i$, determine $a_i(x_t)$ according to the membership function $i$.
(3) For every rule $i$, select $k[i]$ with an EEP (formula (7)).
(4) Calculate the inferred action $a(x_t)$ (formula (8)).
(5) Calculate the corresponding $Q(x_t, a(x))$ (formula (9)).
(6) Perform the action $a(x_t)$, receive the reward $r_{t+1}$, and observe the next state $x_{t+1}$.
(7) For every rule $i$, calculate $a_i(x_{t+1})$.
(8) Calculate $V_i(x_{t+1})$ (formula (10)).
(9) Calculate $\Delta Q$ (formula (11)).
(10) Update $q[i,j]$: $q[i,j] \leftarrow q[i,j] + \Delta q[i,j]$ (formula (12)).
(11) $t \leftarrow t + 1$, go to (3).

### Table 1: FQL algorithm.

| Rule | Action $a_i(x_t)$ | Reward $r_{t+1}$ | Next State $x_{t+1}$ | $q[i,j]$ Update |
|------|-------------------|-------------------|----------------------|----------------|
| 1    |                    |                   |                      | + \Delta q[i,j] |
| 2    |                    |                   |                      |                |
| 3    |                    |                   |                      |                |
| 4    |                    |                   |                      |                |

### 4.2. Decentralized Self-Optimization Approach on the FQL.

In our decentralized self-optimization approach, the multilearning agents are the eNBs involved in the optimization process and each eNB runs its own process based on the FQL. In such a multiagent setting where each agent learns independently, we can regard the other agents as part of the environment. Although for such case the convergence to the optimal point could not be proved rigorously, this multi-agent learning approach has been shown to converge in multiple applications [14].

We adopt the current downtilt of each eNB as the state $x$ of the FQL for each agent, which is fuzzified by 5 membership functions (one function denotes one rule, i.e., $I = 5$), as depicted in Figure 4. And the action $a$ is the angle modification applied to the current downtilt with the following settings:

$$J = 5, \quad a[i,1] = -2, \quad a[i,2] = -1, \quad a[i,3] = 0,$$

$$a[i,4] = 1, \quad a[i,5] = 2 \quad (i = 1, 2, 3, 4, 5).$$

The change of the KPI is chosen as the reward function (i.e., $r_t = KPL - KPL_{t-1}$). In brief, by adjusting the antenna downtilt and receiving the feedback of reward, each eNB tries to maximize the KPI with the learning technique.

### 5. Simulation Setting

The simulation results presented in this section are derived from a dynamic LTE-based system-level simulator developed with Matlab tool in the light of [15]. The simulation scenario is based on a hexagonal network deployment with a number of eNBs and UEs randomly distributed in the network area.

In the downlink, the received signal power of each UE in its serving cell is influenced by the thermal noise and interferences from its neighboring cells. The received SINR is calculated as

$$\text{SINR}_u = \frac{P_{tb,u}}{\sigma^2 + \sum_{k \neq b} P_{k,b}},$$

where $P_t$ is the transmit power (in linear unit) of the cell $i$ (assuming the cell $b$ is the serving cell for user $u$); $g_{tu}$ is the path loss (in linear unit) from the cell $i$ to the user $u$; $\sigma^2$ is the variance of receiver thermal noise modeled as AWGN; the spectral efficiency for each UE is mapped from the received SINR as follows:

$$\text{SE}_u = \Omega(\text{SINR}_u),$$

where $\Omega$ is a step function mapping SINR to the spectral efficiency, which is derived from LTE link-level simulation results and illustrated in Figure 5 [16].

The key parameters for scenario configuration are listed in Table 2, many of which are from 3GPP reports [17]. Three use cases defined by 3GPP [9] are considered in our simulations. In the use case shown in Figure 6(a), there are coverage holes among the cells, and the RF parameters are to be adjusted to expand the coverage area of each cell. In the use case shown in Figure 6(b), LTE systems are deployed with islands of coverage, but due to poor RF planning, the deployed network may fall short of the designed footprint. As a result, the whole coverage is to
be enlarged by adjusting the RF parameters of the eNBs. In the last case shown in Figure 6(c), a new site is added into the existing coverage. In such case, the new site and the surrounding eNBs need to automatically adjust their RF parameters to minimize interference in the area, while maintaining the service coverage.

In this paper, each site has three sectors, and we call sector as cell. Each eNB controls one cell, and it is an SON entity that runs the FQL algorithm. The central entity is in control of all the decentralized entities, deciding when to start/stop the FQL algorithms running in each SON entity, as well as the setting of the FQL algorithm parameters listed in Table 3.

Considering the total delay to be at the level of 100 ms, we set the time-step size as 200 ms in this simulation. And at each time step, only one agent is activated, in order to ensure the validity of the approximation that the other agents are considered as part of the environment. For comparison, we find the optimal solution by searching all possible downtilt angles (from 4.0° to 20.0°, with 0.5° as the step) for all eNBs in the three use cases respectively.

6. Results and Discussion

In practical application, when to stop the optimization process should be decided by the central entity who compares the average of all the JPMs reported by the involved eNBs with preset threshold, for example, the initial one minus an expected gain from the optimization process. But in order to illustrate the convergence of the FQL algorithm, we run all the simulations for 2000 steps in the three cases.

Figure 7(a) shows the convergence curves for the proposed approach and the global optimal value searched offline in the Isolated Island Case; and Figure 7(b) presents the spectral efficiency distribution before and after the optimization process in this case. Figures 8 and 9 indicate the results in the coverage hole case and the newly added site case, respectively.

From Figure 7(a), we can see that in the isolated island case, our proposed approach obtains remarkable gain (about 34%), and it is close to the global optimal value which has about 38% gain. Figure 7(b) illustrates that the system coverage (i.e., 5% tile of the spectral efficiency) and capacity
Table 2: Scenario configuration.

| Parameter                          | Value                                      |
|-----------------------------------|--------------------------------------------|
| Use case                          | Isolated island                           |
| Cell layout                       | 7 sites (3 sectors)                        |
| Wrap around                       | No                                         |
| Users active per sector           | 40                                         |
| Intersite distance                | 500 m                                      |
| Carrier frequency                 | 2.0 GHz                                    |
| Minimum distance to base station  | 35 m                                       |
| Propagation loss                  | $-128.1 - 37.6 \log_{10}(d)$, $d$ in km   |
| Shadowing standard deviation      | 8 dB                                       |
| Shadowing correlation distance    | 50 m                                       |
| Shadowing correlation             | intersite: 0.5; intersector: 1.0           |
| Penetration loss                  | 20 dB                                      |
| Receiver noise figure             | 7 dB                                       |
| Thermal noise density             | $-174$ dBm/Hz                              |
| Channel bandwidth                 | 10 MHz                                     |
| BS maximum transmit power         | 46 dBm                                     |
| BS antenna gain (boresight)       | 14 dBi                                      |
| BS antenna beamwidth @3 dB        | 70 deg                                     |
| BS antenna front-to-back ratio    | 20 dB                                      |
| BS antenna height                 | 32 m                                       |
| UE antenna gain                   | 0 dBi                                      |
| User movement                     | Random walk: 3 km/h                        |
| Traffic model                     | Full buffer                                |
| Scheduling algorithm              | Round robin                                |

Figure 8: Performance in the coverage hole case: (a) the convergency of the average JPM of the involved eNBs; (b) the CDF of the spectral efficiency.

(i.e., 50% tile of the spectral efficiency) are improved by roughly 67% and 27% from the initial coverage and capacity performance, respectively.

In the coverage hole case as shown in Figure 8, the final JPM obtained by our proposed approach is very close to the global optimal as well, and the system coverage improvement is about 380%, which is much more remarkable than the capacity improvement. It is because in this use case, the coverage performance is rather poor at the beginning due to the large coverage hole at the cell edges, and in the optimization task the coverage is greatly emphasized with the JPM factor $\lambda = 0.8$. 
Figure 9: Performance in the newly added site case: (a) the convergency of the average JPM of the involved eNBs; (b) the CDF of the spectral efficiency.

In the newly added site case, the coverage problem is relatively not so serious while the capacity problem may be notable due to the newly added site which would cause great impact on the central users in the cells close to the new site, so we emphasize more about the capacity (with $\lambda = 0.3$). As depicted in Figure 9(b), our approach achieves 13% capacity improvement and the similar coverage improvement.

In sum, the results in all these cases demonstrate that the proposed approach is able to achieve high-performance gain in terms of coverage and capacity, which is close to the optimal. And the tradeoff between coverage and capacity can be easily balanced by setting the factor $\lambda$. Besides, from the results we claim that 1000~1500 steps are enough for the convergence in these three cases, since there is little improvement of the average JPM after 1500 steps in Figures 7(a) and 9(a), and after 1000 steps in Figure 8(a). Yet the actual time for reaching the optimal is determined by the time demanded for the eNB to perform the adjustment, and the time for the user equipment to feedback measurement. Considering the total latency is at the level of 0.1 second, we set the time step to be 0.2 second in the simulations, and thus the total time for 1000~1500 steps is 200~300 seconds. Hence, the convergence rate of the proposed approach can meet the need of practical applications.

7. Conclusion

This paper presented an online approach for optimizing coverage and capacity autonomously in LTE networks, which is based on the central control mechanism, and the decentralized fuzzy Q-learning algorithm. All learning agents are in the control of the central entity, and try to optimize its antenna downtilt automatically using information from its own and its neighboring cells.

From the simulation results obtained in different use cases, we can draw the conclusion that our proposed approach not only achieves remarkable performance gain in terms of coverage and capacity, but also has good characters of convergence rate and stability in the multi-agent system. In addition to coverage and capacity optimization, this automatic approach with the ability to learn from the changing environment may also provide other self-optimizing capabilities for LTE self-organizing networks.

Acknowledgment

This work was supported by 973 Project (no. 2012CB 31600), Beijing Natural Science Foundation Funded Project (no. 4110001), National S&T Major Project (no. 2010ZX 03005-003 and no. 2011ZX 03003-002-01).

References

[1] 3GPP TS 32.500, “Self-organizing networks (SON); Concepts and requirements (Release 9),” V9.0.0, December 2009.
[2] 3GPP TR 36.902, “E-UTRA; Self-configuring and self-optimizing network (SON) use cases and solutions (Release 9),” V9.3.1, March 2011.

[3] 3GPP TS 32.541, “SON; Self-healing concepts and requirements (Release 10),” V2.0.0, March 2011.

[4] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, “Optimization by simulated annealing,” Science, vol. 220, no. 4598, pp. 671–680, 1983.

[5] I. Siomina, P. Várbrand, and Y. Di, “Automated optimization of service coverage and base station antenna configuration in UMTS networks,” IEEE Wireless Communications, vol. 13, no. 6, pp. 16–25, 2006.

[6] T. Cai, G. P. Koudouridis, C. Qvarfordt, J. Johansson, and P. Legg, “Coverage and capacity optimization in E-UTRAN based on central coordination and distributed Gibbs sampling,” in Proceedings of the 71st Vehicular Technology Conference (VTC'10), pp. 1–5, May 2010.

[7] R. S. Sutton and A. G. Barto, Reinforcement Learning: An Introduction, MIT Press, 1998.

[8] 3GPP TS 36.300, “E-UTRA and E-UTRAN; Overall description; Stage 2 (Release 9),” V9.9.0, December 2011.

[9] 3G Americas, “The benefit of SON in LTE: self-optimization and self-organizing networks,” July 2011.

[10] ITU-Recommendation, “Requirements related to technical performance for IMT-Advanced radio interface(s),” Rec. ITU-R M.2134, November 2008.

[11] IEEE 802.16m-07/004r4, “IEEE 802.16m Evaluation Methodology Document (EMD),” November 2008.

[12] W. C. Y. Lee, Mobile Communications Engineering, McGraw-Hill, 1998.

[13] P. Y. Glorennec, Reinforcement Learning: An overview, European Symposium on Intelligent Techniques, Aachen, Germany, 2000.

[14] L. Panait and S. Luke, “Cooperative multi-agent learning: the state of the art,” Autonomous Agents and Multi-Agent Systems, vol. 11, no. 3, pp. 387–434, 2005.

[15] ITU-Recommendation, “Guidelines for evaluation of radio transmission technologies for IMT-2000,” Rec. ITU-R M.1225, 1998.

[16] 3GPP TSG-RAN WG1 R1-071987, “LTE downlink performance verification for frame structure type 2,” Telephone Conference, April 2007.

[17] 3GPP TR 36.814, “E-UTRA; Further advancements for E-UTRA physical layer aspects (Release 9),” V9.0.0, March 2010.
