Power Allocation in Multi-cell Networks Using Deep Reinforcement Learning

Yong Zhang, Canping Kang, Tengteng Ma, Yinglei Teng, Da Guo

School of Electronic Engineering, Beijing University of Posts and Telecommunications, Beijing, P.R. China
Email: {Yongzhang, canping_kang, mtt, lilytengtt, guoda}@bupt.edu.cn

***Abstract*—In this paper, multi-cell power allocation approach is researched. Different from the traditional optimization decomposition method, Deep Reinforcement Learning (DRL) method is employed to solve the power allocation issue which is an NP-hard problem. The objective of our work is to maximize the overall capacity of the entire network in the scenario where the base stations are randomly and densely distributed. We propose a wireless resource mapping method and a deep neural network for multi-cell power allocation named as Deep-Q-Full-Connected-Network (DQFCNet). Compared with the water-filling power allocation and Q-learning method, DQFCNet can achieve a higher overall capacity. Furthermore, the simulation results show that DQFCNet has significant improvement in convergence speed and stability.**

Keywords: Deep Learning, Reinforcement Learning, Power allocation, resource allocation, multi-cell networks, power control

I. Introduction

Power and wireless resource allocation problem has been studied in Long-Term Evolution (LTE) system and 5G mobile communication system for the last decade. With the rapid development of wireless communication, the network is becoming larger in scale. Especially in Ultra Dense Network (UDN), the number of Base Stations (BSs) increases dramatically. The problem of the optimal allocation of power and wireless resource becomes particularly acute. Inappropriate allocation solutions reduce the network spectral efficiency.

The resource allocation in multi-cell is an NP-hard problem [1]. Currently, a lot of researchers investigate the resource allocation problem in single-cell environments, including the heterogeneous network [2]-[4]. However, the inter-cell interference increases the complexity of the problem. Higher transmit power of one BS could improve its capacity but reduce the capacity of the neighboring cells. No BS can unilaterally modify its transmit power to improve the sum rate of the entire network. The main approach towards self-organizing can be divided into two main groups: learning and optimization [5].

In the past years, the optimization becomes the most popular means which present in a lot of literature [6]-[9]. Reference [6]-[8] propose joint resource allocation and power control scheme for maximum energy-efficient. As the most fundamental issue of communication networks, we are more concerned with the overall capacity of the entire network. Regardless of optimizing the power allocation mechanism, Jia Shi, et al. designed the subcarrier allocation algorithms to mitigate Inter-Cell Interference (ICI) and enhance the spectral efficiency [9] in hexagonal LTE/LTE-A networks. Reference [10] proposes a cooperative ICI coordination technique in the same network, where adjacent eNodeBs negotiate with each other to determine the transmit power according to the users' satisfaction function. The solution is simple but the authors don't verify the solution's convergence in a large-scale network. Reference [1] studies the coordinated scheduling and power control method to maximize weighted sum-rate in hexagonal networks. Reference [11] decomposes this issue into graph-partitioning-based subproblems and solves it using optimal and heuristic approaches.

Using the learning methods is a novel way to solve the joint power and resource allocation issue in multi-cell networks [12]. Some work has been done on resource allocation in multi-cell networks. Wang, et al. investigated the transmit power issue associated with indoor enterprise closed-access small BS network [13]. Usama, et al. suggested a multi-parameter Q-Learning algorithm to minimize the ICI and maximize the total Signal to Interference plus Noise Ratio (SINR) using sub-band power factor and edge-to-center boundary [14]. However, maximizing the SINR of the entire network does not mean the maximum overall capacity of the entire network.

Inspired by the success of deep reinforcement learning (DRL) [15], a power allocation approach is proposed in multi-cell networks using DRL. We introduce a deep reinforcement learning into wireless networks based on the characteristics of the wireless networks. First of all, considering the huge state space of wireless networks, the transmit power is discretized and the influence of discrete granularity is analyzed in term of the convergence and stability of the system. Secondly, the dynamic state adding strategy is proposed to reduce the computational complexity. Finally, the deep neural network and objective function are introduced, and speed

978-1-5386-6358-5/18/$31.00 ©2018 IEEE

up the convergence. It is verified by numerical simulation that our algorithm can improve the spectrum efficiency of the network.

II. SYSTEM MODEL AND PROBLEM FORMULATION

A. System Model

The system is considered as a downlink multi-cell OFDM cellular with the densely deployed BSs. The system uses the reuse-1 model [16], i.e., each BS can use all the available bandwidth. The network model is the same as [11] which is shown in Fig.1. A central controller can collect the information of the entire network including SINR and transmit power. The number of users associated with BS n at time t is $UEs_t^n \in \{M_t^1, \dots M_t^n, ..., M_t^N\}$ ($\sum_{n=1}^{N} M_t^n = M$),where M is the total number of users. Random walk model is adopted in our work. The moving speed of the user at time t is $V_t^m (0 \le V_t^m \le Vmax)$,and the moving angle is $D_t^m (0 \le D_t^m \le 2\pi)$. The geographical distribution of BSs follows the Poisson point process model. Each BS BS_n^f fully reuses the K orthogonal subcarrier. When a user connects to the BS, the BS is activated and the transmit power is p_t^n. Each user can only connect to one BS at time t. Each subcarrier can only be assigned to one user. The six-path fading channel model is used for evaluation. The choice of channel model does not affect our proposal's performance.

B. Problem Formulation

Let $\eta_t^{(n,k,m)}$ denotes the received SINR of the n-th BS served user m on the k-th subcarrier at time t, where $k \in \{1, ..., K\}$ and is given by

$$\eta_t^{(n,k,m)} = a_t^{(n,m)} \alpha_t^{(n,k,m)} \frac{g_{n,m}^{(k)} p_t^{(n,k)}}{\sum\limits_{n' \neq n} g_{n',m}^{(k)} p_t^{(n',k)} + \sigma^2} \quad (1)$$

where $g_{n,m}^{(k)}$ and $g_{n',m}^{(k)}$ denote channel gains of the n-th and n'-th BSs to user m on the k-th subcarrier respectively. And $p_t^{(n,k)}$ and $p_t^{(n',k)}$ denote the total transmit power of the n' BSs on the k-th subcarrier respectively. $\alpha_t^{(n,k,m)}$ denotes whether BS n allocates subcarrier k to user m $\alpha_t^{(n,k,m)} \in [0, 1]$, and σ^2 represents the power of Gaussian white noise. $a_t^{(n,m)}$ indicates whether user m is connected to BS n:

$$a_t^{(n,m)} = \begin{cases} 1, & \mathrm{m} \in \mathrm{M_i} \\ 0, & \mathrm{m} \notin \mathrm{M_i} \end{cases} \quad (2)$$

The system performance is analyzed in terms of the overall capacity measured in (bps/Hz). The capacity achieved by BS_n^f at its associated user on subcarrier k at time t is given by:

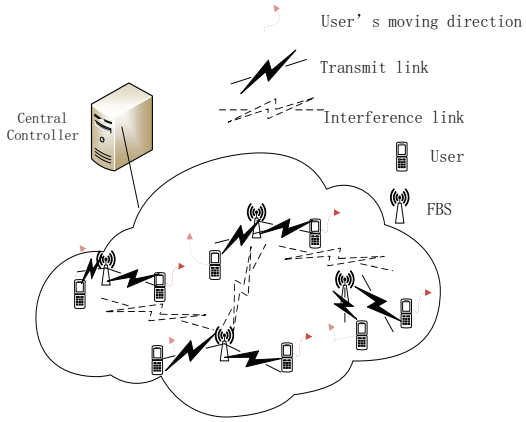

Fig. 1. System model.

$$C_t^{(n,k)} = \frac{B}{K} \log_2 (1 + \sum_{m=1}^{M} \eta_t^{(n,k,m)}) \quad (3)$$

The overall capacity can be defined as:

$$R_t = \sum_{n=1}^{N} \sum_{k=1}^{K} C_t^{(n,k)} \quad (4)$$

Our objective is to increase the overall capacity of the entire network by adjusting the transmit power of BSs on subcarrier $p_t^n = [p_t^{(n,1)} ... p_t^{(n,k)} ... p_t^{(n,K)}]$ based on nearly optimal subcarrier allocation. The optimization problem can be formulated as follows:

$$\begin{cases} \arg\max R_t \\ s.t. C1 : p_t^{(n,k)} \ge p_{\min}, \forall n, k \\ \quad C2 : \sum\limits_{n,k} p_t^{(n,k)} \le p_{\max}, \forall n, k \end{cases} \quad (5)$$

where $p_{\max}$ is the maximum transmit power of the BS, and $p_{\min}$ is the minimum transmit power of the subcarrier.

In the network initialization phase, the users complete the association with the base stations according to the maximum SINR criterion. The interference between users and base stations mainly comes from inter-cell interference, and there is no interference between users in the intra-cell. The link rate is affected by the signal strength from the attached base station to the user, and the inter-cell interference strength. Our goal is to adjust the transmission power of base stations to increase the overall capacity of the entire network. Considering the fairness of users, we allocate the downlink subcarrier of one base station to all attached users equally. The power on the subcarrier is initially allocated according to the water-fill algorithm.

The above problem is a multi-objective non-convex optimization problem. The traditional method to solve this problem is heuristic search algorithm. However, most of these algorithms are very inefficient that have

long running time and cannot be adjusted online in real time. In order to solve this problem, in the next chapter, we will introduce a deep reinforcement learning algorithm to solve this problem.

III. DEEP-Q-FULL-CONNECTED-NETWORK SOLUTION

A. Reinforcement Learning - Multi-Agent Q Learning

The scenario of distributed cognitive BSs can be mathematically formulated using stochastic games [17], where the learning process of each BS can be represented by the quintuple $\{N, S, A, P, R(s, \vec{a})\}$ where:

- $N = \{1, 2, \ldots, N_f\}$: is the set of agents(i.e. BSs).
- $S = \{S_1, S_2, ..., S_m\}$: is the set of possible states the system can occupy, where m is the number of possible states.
- $A = \{a_1, a_2, ..., a_l\}$: is the set of possible actions
- P: is the probabilistic transfer function, which represents the probability of the system shifting from one state to another.
- $R(s, \vec{a})$: defines the reward function that determines the reward to the agent n when the joint action $\vec{a}$ is performed in state $s \in S$.

Q-learning, as a model-free reinforcement learning algorithm based on value function, has a natural advantage in solving the dynamic wireless networks environment problem. In the Q-learning algorithm, each agent converges its own behavioral value function through continuous iterative learning. This behavioral value function is generally represented by a table $Q(s_m, a_l)$, where $a_l \in A, s_m \in S$. Thus, the size of this table is $m \times 1$. The Q-value $Q(s_m, a_l)$ represents the expected value of the cumulative reward of action a_l at state s_m over an infinite time horizon, and is given by:

$$Q_\pi(s, a) = E_\pi[\sum_{k=0}^{\infty} R_{t+1} + \gamma Q(S_{t+1}, A_{t+1}) | S_t = s, A_t = a] \tag{6}$$

where $R_{(t+1)}$ denotes the instant reward after action a is taken in the state s. $0 \leq \gamma \leq 1$ is the discount coefficient, and represents the weight of the empirical value.

The Q-value is updated as follows:

$$\begin{aligned} Q_n^{t+1}(s_m^n, a_l^n) := (1-\alpha) Q_n^t(s_m^n, a_l^n) + \\ \alpha(R_n^t + \gamma \max_{a' \in A} Q_n^t(s_{m'}^n, a')) \end{aligned} \tag{7}$$

where $0 \leq \alpha \leq 1$ denotes the learning rate.

B. power allocation algorithm and Q learning mapping

A proposed Q-learning algorithm based on power allocation is a multi-agent algorithm based on continuously interacting with the environment. The agents, states, actions, and the reward function are defined as follows:

- Agent: BS n, $1 \leq n \leq N_f$.
- States: $s_t^{n,k} = \{M_t^n, p_t^n\}$, where M_t^n represents the number of users connected to the BS n at time t, and p_t^n represents the power of the n-th BS at time t. In order to reduce the complexity of the algorithm and the state space of the network, the power of the BS is discretized. The discrete rules are as follows:

$$p_t^n = \tau, (P_{\max}^f - A_\tau) \leq \sum_{k=0}^{K} p_t^{n,k} < (P_{\max}^f - A_{\tau+1}) \tag{8}$$

where $\tau \in \{0, 1, 2, 3, 4, 5\}$,$\mathrm{A}_0 = \mathrm{p}_{\max}^f$, $\mathrm{A}_6 = 0$, and A1 A5 are arbitrarily selected thresholds.
- Action: $a_n = \{k_n, \vec{p}_t^{(n,k)}\}$ where k_n denotes the k-th subcarrier of the n-th BS. $\vec{p}_t^{(n,k)}$ denotes the adjustment value of the power on the k-th subcarrier of the n-th BS, and $\vec{p}_t^{(n,k)} \in \{-|\vec{p}_t^{(n,k)}|, 0, +|\vec{p}_t^{(n,k)}|\}$ is determined by the change of the total throughput C_t^h of the h users around the current user after taking action. If C_t^h increases, then $\vec{p}_t^{(n,k)} = +|\vec{p}_t^{(n,k)}|$, and vice versa.
- Reward: The reward of the n-th BS on the k-th subcarrier is defined as [18]:

$$r_t^{n,k} = \begin{cases} 1 - e^{-(C_t^{(n,k)})}, & \sum_{k=1}^{K} p_t^{(n,k)} \leq P_{\max}^f \\ -1, & Otherwise \end{cases} \tag{9}$$

Due to the dynamic nature of the environment, the state space of the environment is relatively large, and the size of the Q-value table for each agent is not same. If we determined a large size Q-value table with fixed size, the computational complexity will be much higher. Therefore, a dynamic state adding method is introduced, i.e., when there is a new state, this state will be added to the state sets automatically. No need to customize a Q table for each agent is the advantage of the dynamic method. Also, it can improve the search efficiency of Q tables and enhance storage space utilization. But it also means that the Q table takes a long time to converge because the network needs to re-converge again when a new state occurs. Therefore, this paper proposes a DRL algorithm to speed up the convergence of the Q-value table.

C. DQFCNet

DeepMind proposed the concept of [15]. Compared with Q-learning introduced earlier, the update of the value function is the parameter of the neural network instead of the Q-value table. The update method adopts the gradient descent algorithm. The value function is updated as follows:

$$\begin{aligned} \theta_{\mathrm{t+1}} = \theta_{\mathrm{t}} + \alpha[r + \gamma \max_{a'} Q(s', a'; \theta^-) \\ - Q(s, a; \theta)] \nabla Q(s, a; \theta) \end{aligned} \tag{10}$$

where $r + \gamma \max_{a'} Q(s', a'; \theta^-)$ is the Temporal Difference (TD) target. $Q(s, a; \theta)$ is the network object by value function approximation. And $\nabla Q(s, a; \theta)$ is the gradient.

In many games, such as Go, DQN has achieved good results. The game environment is based on noise-free,

complete information state. However, the wireless networks environment has noise, incomplete information, and soft boundaries. It is difficult to map this information into DQN. Therefore, a simplified version of DQN (DQFCNet) is designed to accelerate the convergence of the behavioral value function by discretizing the number of environmental states and using a deep full-connection neural network.

First of all, a deep fully connected neural network is proposed as shown in Fig. 2, which contains four layers and the input layer data is

$$inputs_t = [M_t^1, \dots M_t^n, ..., M_t^N, {p_t}^{(1,1)}, \dots, {p_t}^{(n,k)}, \dots, {p_t}^{(N,K)}, {csi_t}^{(a_{n,1},k)}, \dots {csi_t}^{a_{n,m},k}, ..., {csi_t}^{(a_{n,M},K)}] \quad (11)$$

where ${csi_t}^{a_{n,m},k}$ represents the channel state information of the m-th user on the k-th subcarrier of the n-th BS. The middle two hidden layers are mainly to increase the nonlinearity of the network optimization and improve the fitting ability of the network. The output layer data is:

$$outputs = [|\vec{p}_t^{(1,1)}|, ..., |\vec{p}_t^{(n,k)}|, ..., |\vec{p}_t^{(N,K)}|] \quad (12)$$

where $|\vec{p}_t^{(n,k)}|$ represents the total power adjustment value of the n-th BS on the k-th subcarrier.

The deep neural network adopts dropout technology [19], which increases network generalization ability, reduces network variance, and prevents overfitting. In order to speed up the training of the network, the AdamOptimizers [20] is used in the back propagation of the network. The loss function of the deep neural network is designed as follows:

$$Loss = \frac{1}{s}\sum_{i=0}^{s}(q_z - o_z)^2 + 1/mean(o_z) + c||\theta||_2 \quad (13)$$

Formula(11), q_z represents the Q learning adjustment strategy. o_z represents the output of the neural network, i.e., the total power adjustment value of each BS on each subcarrier. c is a penalty factor. θ represents the weight parameter. The first part of the loss function is to make the output of the neural network approximate the Q-value table as much as possible. The second part is the reciprocal of the mean value of the output value. When the loss function is minimized, the network will try to improve the average power adjustment of the BSs, Which can effectively improve the overall capacity of the entire network. The third part is an l2-norm, whose role is to make some restrictions on the parameter θ, so as to prevent the network from overfitting.

During the training process, the neural network model f is evaluated. If the network adjustment capability of f is stronger than the current best model f^*, the current model f is used as the best model f^* to enhance Q learning.

To accelerate the convergence process and improve the spectrum efficiency of the entire network, the output of deep neural networks is introduced to fasten the action selection (Eq. 13).

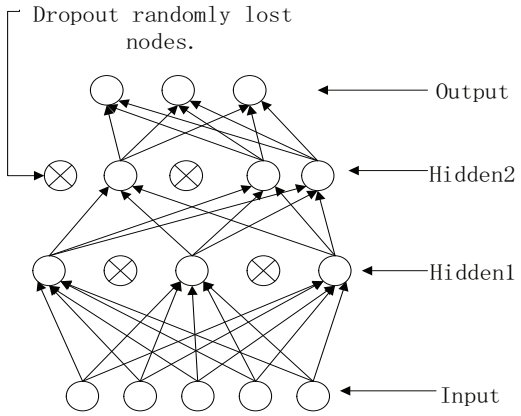

Fig. 2. The deep neural network model.

$$\mathrm{a}_n^k = \arg\max_{a\in A}((1-\beta)*\sum_{1<n'<N_f} Q_{n'}^k(s^{n',k},:)+\beta*f_n^*) \quad (14)$$

where f_n^* represents the adjustment strategy of the current best deep neural network model on the n-th BS, and it plays a role in strategy enhancement. Where β is used to adjust the emphasis on the output of the neural network. In the initial stage of network training, we should pay more attention to the output of the neural network, so that β should be relatively large. While agents continue to learn, the weight of the Q-value table should become larger.

D. Complexity Analysis

The computational complexity of DQFCNet is compared with the algorithm proposed in paper [11]. In [11], Branch_and_Cut algorithm is used to find the upper bound of the overall capacity of the entire network in the multi-cell scenario. The main computational overhead is generated by Branch_and_Cut algorithm. Assuming that the state of the environment is represented by a vector of length m. The computational complexity of this method is $O2^{\mathrm{m}}$. In the deep neural network proposed in this paper, the activation function uses relu, and after the model is trained, the complexity of the algorithm is: $O(m*(n1*n2+n2*n3+n3*n4))$, where $n1$, $n2$, $n3$,$n4$ are the number of neurons in each layer. As can be seen from the above analysis, DQFCNet greatly reduces the complexity of the algorithm.

IV. SIMULATION AND EVALUATION

The entire cellular network environment includes N_f BSs. Each user connects to the BS with highest SINR. All BSs share the spectrum bandwidth. In the simulation, we set the noise power σ^2=10^{-7}, and the power regulation amplitude of each BS on each subcarrier, where A1-A5=[25,20,15,10,5]. Some other parameters settings are shown in Table I and β is updated as Table II . And

TABLE I
SIMULATION PARAMETERS

Parameters	Values
Bandwidth B	10 MHz
The number of BSs	32
The number of UEs	200
Radius r	15 m
Initial power p	0-5 dbm
Network Size	100*100 m^2
$p_{\max}$	30 dbm
$p_{\min}$	-100 dbm
$v_{\max}$	1m/s
h	5
α	0.5
γ	0.9
ε	0.2
Dropout rate	0.8

TABLE II
β **UPDATE**

Parameters	Values
Iteration step (0-30)	0.8
Iteration step (30-60)	0.6
Iteration step (60-100)	0.3
Iteration step (100-)	0.1

at each step, the network updates the topology once due to the node mobility.

The number of neurons per layer is configured as [14880,10000,5000,2048] in the deep neural network. The number of training parameters is 209,057,048.

Fig. 3 shows that the frequency efficiency of the three modes after the models converge. For each iteration, the user moves and the network topology is updated. Fig.3 indicates that DQFCNet can achieve highest frequency efficiency and more stable than Q-learning and water-filling algorithm. And we can be seen that DQFCNet algorithm is approaching the upper bound of the algorithm mentioned in [11].

The convergence of the network is illustrated in Fig.4. The Q-learning fluctuates greatly because of the node mobility. Q-learning needs to re-compute and converge when the network topology changes. In the dynamic scenario, although DQFCNet will also fluctuate, it relatively stable compared with Q-learning. At the same time, with the strengthening of the strategy of deep neural network, DQFCNet improves the spectral efficiency a lot.

Fig.5 shows the convergence contrast diagram of power discretization into three levels. As it shows, under different discrete granularity, the convergence speed of the whole network is very small. But it can obviously be found that the whole network is more stable and the spectral efficiency higher when discrete granularity is 6. If the power discrete into 15, the number of ambient states is too large. After the model has converged, the system is stable with little fluctuation. When the dispersion is 3, after the model convergence, it fluctuates greatly because the discrete granularity is too coarse to represent the real change of the environment.

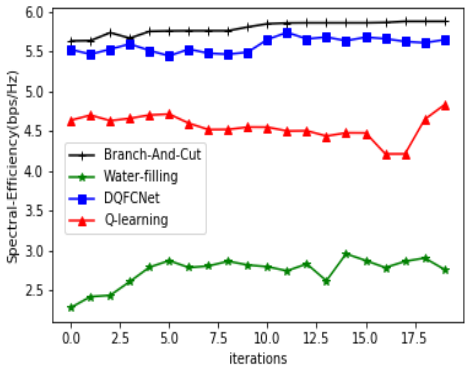

Fig. 3. Comparison of Spectral Efficient (the model has converged).

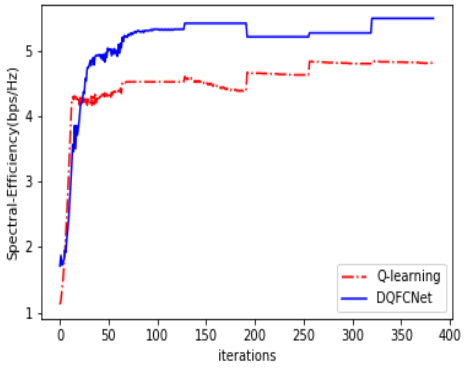

Fig. 4. Comparison of convergence speed.

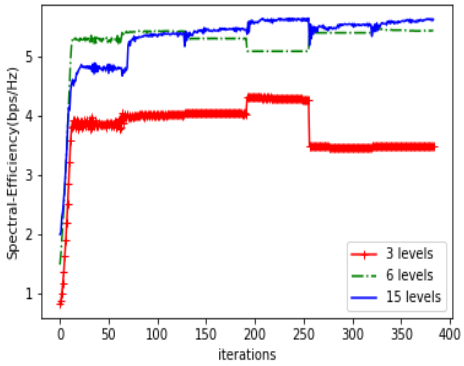

Fig. 5. Comparison of convergence on the different discrete granularity.

V. Conclusion

The multi-cell power allocation problem is a non-convex problem and has been a research difficulty in

the wireless communication field. The optimization decomposition is the traditional solution which has high computation complexity. Inspired by the research results in the field of deep learning, we proposed the DQFCNet method to apply deep reinforcement learning to multi-cell power allocation and achieved good system capacity gains. In addition, due to the use of deep neural networks, the convergence speed and stability of DQFCNet have been improved.

The success of deep learning is instructive for the researches of multi-cell wireless network optimization. It should be pointed out that although this paper aims to maximize the overall capacity of the entire network, our proposal can also be extended to other technical indicators, such as system energy efficiency, Quality of Experience, and so on.

ACKNOWLEDGMENT

This work is supported by National Natural Science Foundation of China under Grant No. 61771072.

REFERENCES

[1] Zhang, Honghai, et al. "Weighted Sum-Rate Maximization in Multi-Cell Networks via Coordinated Scheduling and Discrete Power Control." IEEE Journal on Selected Areas in Communications 29.6(2011):1214-1224.

[2] Yang, Kai, et al. "Energy-Efficient Downlink Resource Allocation in Heterogeneous OFDMA Networks." IEEE Transactions on Vehicular Technology 66.6(2017):5086-5098.

[3] Yang, Zhaohui, et al. "User Association, Resource Allocation and Power Control in Load-Coupled Heterogeneous Networks." GLOBECOM Workshops IEEE, 2017:1-7.

[4] Zou, Shangzhang, et al. "Joint Power and Resource Allocation for Non-Uniform Topologies in Heterogeneous Networks." Vehicular Technology Conference IEEE, 2016:1-5.

[5] Aliu, Osianoh Glenn, et al. "A Survey of Self Organisation in Future Cellular Networks." IEEE Communications Surveys & Tutorials 15.1(2013):336-361.

[6] Wang, Xiaoming, et al. "Energy-Efficient Resource Allocation in Coordinated Downlink Multicell OFDMA Systems." IEEE Transactions on Vehicular Technology 65.3(2016):1395-1408.

[7] Lahoud, Samer, et al. "Energy-Efficient Joint Scheduling and Power Control in Multi-Cell Wireless Networks." IEEE Journal on Selected Areas in Communications 34.12(2016):3409-3426.

[8] Yang, Kai, et al. "Energy-Efficient Downlink Resource Allocation in Heterogeneous OFDMA Networks." IEEE Transactions on Vehicular Technology 66.6(2017):5086-5098.

[9] Shi, Jia, L. L. Yang, and Q. Ni. "Novel Intercell Interference Mitigation Algorithms for Multicell OFDMA Systems With Limited Base Station Cooperation." IEEE Transactions on Vehicular Technology 66.1(2017):406-420.

[10] Yassin, Mohamad, et al. "Cooperative resource management and power allocation for multiuser OFDMA networks." Iet Communications 11.16(2017):2552-2559.

[11] Pateromichelakis, Emmanouil, et al. "Graph-Based Multicell Scheduling in OFDMA-Based Small Cell Networks." IEEE Access 2(2014):897-908.

[12] Zhang, Chaoyun, P. Patras, and H. Haddadi. "Deep Learning in Mobile and Wireless Networking: A Survey." (2018).

[13] Wang, Zhiyang, et al. "Learn to adapt: Self-optimizing small cell transmit power with correlated bandit learning." IEEE International Conference on Communications IEEE, 2017:1-6.

[14] Sallakh, Usama, S. S. Mwanje, and A. Mitschele-Thiel. "Multi-parameter Q-Learning for downlink Inter-Cell Interference Coordination in LTE SON." Computers and Communication IEEE, 2014:1-6.

[15] Mnih, V, et al. "Human-level control through deep reinforcement learning. " Nature 518.7540(2015):529.

[16] Mazarico, Juan I, et al. "Detection of synchronization signals in reuse-1 LTE networks." Wireless Days IEEE, 2009:1-5.

[17] Burkov, Andriy, and B. Chaib-Draa. "Labeled initialized adaptive play Q-learning for stochastic games." Draa (2010).

[18] Saad, H, A. Mohamed, and T. Elbatt. "A cooperative Q-learning approach for distributed resource allocation in multi-user femtocell networks." Wireless Communications and NETWORKING Conference IEEE, 2014:1490-1495.

[19] Srivastava, Nitish, et al. "Dropout: a simple way to prevent neural networks from overfitting." Journal of Machine Learning Research 15.1(2014):1929-1958.

[20] Kingma, Diederik P, and J. Ba. "Adam: A Method for Stochastic Optimization." Computer Science (2014).