Deep Reinforcement Learning for Resource Allocation in Massive MIMO

Liang Chen[1,2,3], Fanglei Sun[1], Kai Li[1], Ruiqing Chen[1,2,3], Yang Yang[1],(Fellow, IEEE), Jun Wang[4]
[1]ShanghaiTech University, Shanghai, China
[2]Shanghai Institute of Microsystem and Information Technology, Shanghai, China
[3]University of Chinese Academy of Sciences, Beijing, China
[4]University College London, London, UK

***Abstract*—As the extensive application of massive multiple-input multiple-output (MIMO) in 5G and beyond 5G (B5G) networks, multi-user (MU) MIMO scheduling faces big challenges on performance enhancement with effective interference coordination and computational complexity reduction. Plenty of deep learning and reinforcement learning for wireless resource scheduling are proposed to solve the above issues via a well trained network, instead of executing iteration search on each scheduling period. However, the dimension of the channel state information and the size of user combination set may increase exponentially in massive MIMO system, which makes the neural network over complicated and causes severe convergent issues. In this paper, a novel Actor-Critic framework is developed to overcome the above existing issues for the single-cell downlink multi-user scheduling issue in massive MIMO system. Pointer network is investigated as the policy network in our proposed algorithm, which transfers the complicated selection issue among user combinations to a user sequential selection issue based on conditional probability. Simulation results show that the performance of our method is very close to that of the greedy algorithm with much less computational complexity. Moreover, our proposal is robust and effective with the increase of the number of antennas and users.**

***Index Terms*—Massive MIMO, single-cell downlink MU-MIMO scheduling, pointer network, advantage Actor Critic**

I. INTRODUCTION

Massive MIMO has been one of the key technologies for 5G and B5G networks, due to its potential for high capacity, increased diversity, and interference suppression [1]. Simultaneous communication with multiple users creates multi-user interference and degrades the throughput performance. As the above issues get worse in massive MIMO systems, in the traditional wireless communication field, greedy search based joint of scheduling and precoding algorithms has been studied to reduce the scheduling complexity in [2] and [3]. Although greedy search can significantly reduced the selection complexity compared with exhaustive search, the capacity evaluations based on either linear or non-linear precoding update on the greedy iterations still costs high computational and time resource. As the number of antennas increases in massive MIMO system, this problem becomes more serious.

In recent years, with the rapid development of artificial intelligence techniques, Deep Learning (DL) techniques have achieved impressive results in different fields, such as image processing and natural language processing. In order to improve the resource allocation performance in MU-MIMO system, DL based scheduling algorithms have been explored in [4] and [5], however most of them try to directly find the relationship between the downlink channel of all users with the scheduling result among all possible user combinations. These methods make the DL network really complicated and hardly to be adapted in particularly massive MIMO systems. As we know, Reinforcement Learning (RL) technique is capable of maximizing the cumulative rewards by sequential decision making, and deep learning is good at automatic feature engineering. Due to deep reinforcement learning (DRL) technique has the above two advantages, it becomes a hot research topic for wireless scheduling issues. In [6], the authors proposed an optimal resource allocation algorithm based on DRL, however, all user combinations are regarded as the action space in their proposal, which makes the size of the action space increase exponentially with the number of users. In order to solve action dimensional disaster problem, in [7] and [8], the same value-based multi-agent DRL algorithm are proposed to solve different resource allocation problems. In [9], they investigate the joint design of transmit beamforming at the base station (BS) and phase shifts at the reflecting reconfigurable intelligent surface (RIS) to maximize the sum rate of multi-user downlink MIMO systems utilizing DRL.

Based on the above analysis and our previous work on policy-based DRL scheduling algorithm [5], in this paper we try to explore a new Actor-Critic [10] framework to overcome the above issues for the single-cell downlink scheduling issue in masssive MIMO system. Multi-user scheduling problem can eventually be abstracted into a combinatorial optimization (CO) problem. Pointer network (PN) has been proposed to solve the problem efficiently through supervised learning method [11]. It wisely transfers the complicated selection issue among user combinations to a user sequential selection issue based on conditional probability, and therefore making the scheduling issues not depend on complicated neural networks with big action set any more. However, in many application scenarios, getting high-quality labeled data is expensive and may be infeasible. Thus, in [12], the authors combined DRL and PN to solve the traveling salesman problem (TSP). In [13], the authors replaced the encoder of the PN by element-

This paper is supported by the project of cooperation with Huawei Noah's Ark Lab

wise projections to solve vehicle routing problem (VPR). And in [14], the authors solved the same problem by using Transformer architecture [15] as encoder. In [16], the authors had integrates a large number of literatures that use DRL to solve CO problems.

Motivated by this fact, pointer network is investigated as policy network in our proposed Actor-Critic framework due to its simple structure. Spectral efficiency (SE) is considered as the optimization target. Based on our well trained RL mode, computational complexity caused by greedy search in a big-size action set on each scheduling period can be eliminated. Simulation results show that the performance of the proposed RL-based model is very close to that of the greedy algorithm in testing set. Moreover, simulation results also show that our proposal is robust and effective with the increase of the number of antennas and users.

The main contributions of this paper are summarized as follows:

- we proposed a mapping function by DRL method: $\pi_\theta(\mathscr{H}) \rightarrow \mathscr{S}$, $\mathscr{H}, \mathscr{S}$ are the channel space and optimal scheduled sequence space, respectively. When optimal policy is obtained by training set, it can be directly deployed to the BS of downlink scheduling in the subsequent transmit time interval (TTI), we could obtain a user scheduled sequence without using sophisticate mathematical optimization techniques and multiple iterations to find a optimal user sequence.
- The simulation result show that the performance of our proposed algorithm is very close to that of the greedy scheduling algorithm, however, the running time of our proposed algorithm is ten times faster than that of greedy algorithm, which can effectively reduce system energy consumption and has the ability to be deployed online.

II. SYSTEM MODEL AND PROBLEM FORMULATION

In this section, we assume BS is equipped with M_t transmit antennas. There are L users in a cell, K users are scheduled each time, and each user is equipped with a single antenna. The channel vector from BS to an arbitrary user k is denoted as $\boldsymbol{h_k} \in \mathbb{C}^{1\times M_t}$.

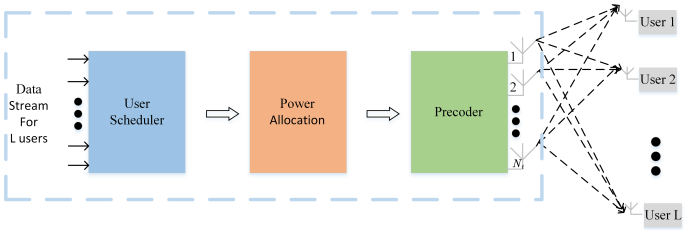


Fig. 1. System model of massive MIMO

The receive signal at each user can be written as:

$$y_k = \sqrt{P_k}\boldsymbol{h_k}\boldsymbol{u_k}x_k + \sum_{i\neq k}\sqrt{P_i}\boldsymbol{h_k}\boldsymbol{u_i}x_i + w_k., \qquad (1)$$

where y_k is the received vector of user k, P_k is the power allocated to user k. Assuming that P_t is the maximum transmit power of the BS, then $\sum_k P_k \leq P_t$. Channel noise $\boldsymbol{w_k} \sim \mathcal{CN}(0, N_0)$,

The signal-to-interference plus noise ratio (SINR) of the arbitrary user k is given by:

$$\text{SINR}_k = \frac{|\boldsymbol{h_k}\boldsymbol{u_k}|^2 P_k}{N_0 W + \sum_{j\neq k} P_j |\boldsymbol{h_k}\boldsymbol{u_j}|^2}, \qquad (2)$$

Where W is the the communication bandwidth. And the instant SE of user k is:

$$R_k = \log(1 + \text{SINR}_k), \qquad (3)$$

Subsequently, we could deduce the instant spectral efficiency of the system as:

$$R_{sum} = \sum_{i\in[K]} \log(1 + \text{SINR}_k). \qquad (4)$$

Our objective is to find out the optimal scheduled sequence $N = [n_1, n_2, \cdots, n_K]$ that can maximize the spectral efficiency of the cell. In our model, suppose the BS could obtain the instantaneous cell channel state information (CSI). When scheduled sequence is given, we could select the row vector of channel matrix based on the scheduled sequence, and form a new matrix $\tilde{\boldsymbol{H}} = \boldsymbol{H}_N$. Then, we could determine the precoding matrix by different schemes e.g, Zero Forcing (ZF). Since we can not increase the power of the signal after precoding operation, therefore, we assume $||\boldsymbol{u_j}|| = 1$.

III. DEEP REINFORCEMENT LEARNING BASED ALGORITHM DESIGN

Fig. 2 shows the overall structure of our proposed framework and the brief implementation of each module. More details are given in the following descriptions.

A. Problem reformulation

In this part, the DRL setting are derived from the optimization equation in Section II. Since RL follows Markov decision process (MDP) which involves a state set, an action set and state transition probabilities. For single-cell downlink MU-MIMO scheduling, in this paper, we try to propose an Actor-Critic framework, where the explicit expression of state transition probability can not be ignored. Based on the optimization problem, we define the state, action and reward function as follows:

- ***State*** **:** The state s_t is determined by the channel matrix $\boldsymbol{H} \in \mathbb{C}^{L\times M_t}$. Since the neural network can only take real rather complex numbers as input, thus, the real part and imaginary part will be separated as independent port, and the input dimension of state will be $\mathbb{R}^{L\times(M_t\times 2)}$
- ***Action*** **:** In our system, at each scheduling period, base station will decide which users will be scheduled and which users will not. Therefore, the action space $\mathcal{A}$ is a subset of $\{1, 2, \cdots, L\}$ and $|\mathcal{A}| = 2^{|L|}$
- ***Reward*** **:** The reward is determined as the cell spectral efficiency by equation (4) when the instantaneous channel matrix $\boldsymbol{H}$ and the scheduled sequence N are given.

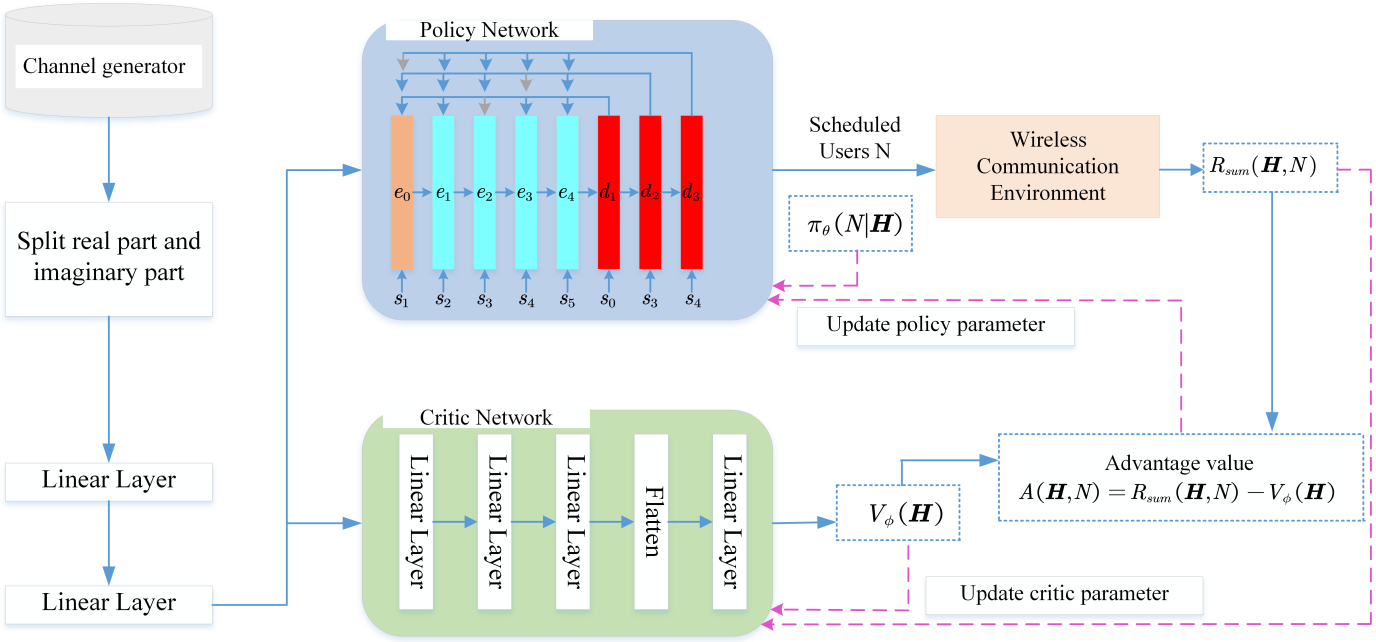


Fig. 2. Algorithm structure

B. Policy network

Based on the above analysis, the action space is too large to use value-based algorithms, (e.g, Deep Q-Learning algorithm). If value-based algorithm is adopted, the output ports of Q value network will be $x^{|L|}$, which is intractable in practice.

Our reinforcement learning algorithm is based on Actor-Critic framework. Pointer network are introduced in our Actor-Critic framework as policy network. Since it could use the chain rule in Pointer network, the joint probability of an action when state is given could be estimated by the following equation:

$$p(n_1, n_2, \cdots, n_K|H, \theta) = \prod_{i=1}^{K} p(n_i|n_1, \cdots, n_{i-1}, \boldsymbol{H}; \theta). \tag{5}$$

The pointer network has an encoder and decoder. The decoder network could transform the state matrix into a latent memory states $\{e_1, e_2 \cdots, e_L\}$. The initial latent state e_0 is unknown, but we could obtain it by learning process. Combining e_0 and latent memory states, we could obtain the encoder hidden state matrix $E \in \mathbb{R}^{h\times(L+1)}$, h is the dimension of hidden state. As for decoder part, the initial latent state will be the last latent memory state of encoder, and the initial input of decoder will be the row mean of input state matrix. In the i-th selection process, the hidden vector is $d_i \in \mathbb{R}^{h\times 1}$, and the weighted vector which could be calculated at the output time i is:

$$\omega_i = \text{softmax}(v^T \tanh(W_1 E + W_2 d_i)), \tag{6}$$

where $W_1, W_2 \in \mathbb{R}^{w\times h}, v \in \mathbb{R}^{w\times 1}$ are learnable parameters of the model. When we have the weight vector ω_i, the output will be the index of maximum value of weight vector: $\arg\max_j(\omega_i^j)$. Thus, the probability of $\pi_\theta(a|s)$ could be decomposed into:

$$\pi_\theta(a|s) = \prod_{i=1}^{\ell} \omega_i^j. \tag{7}$$

The flag of decision stops at the ℓ-th decision, the probability value of ω_ℓ^0 is the largest one.

Combined with our objective function, the loss function of policy network is:

$$J_\theta = -\mathbb{E}_{\boldsymbol{H}\sim\mathcal{D}}[\sum_{N\in\mathcal{A}} Q(N, \boldsymbol{H})\pi_\theta(N|\boldsymbol{H})], \tag{8}$$

where $\mathcal{D}$ is an unknown distribution, and from Bellman equation [17]: $Q(\boldsymbol{H}, N) = R_{sum}(\boldsymbol{H}, N) + \gamma V(\boldsymbol{H}')$. In our scenarios, the trajectory ends directly after the current state decision is completed, thus, $V(\boldsymbol{H}') = 0$. And the gradient of J_θ is formulated using the REINFORCE algorithm [18]:

$$\nabla_\theta J_\theta = -\mathbb{E}_{\boldsymbol{H}\sim\mathcal{D}}[\sum_{N\in\mathcal{A}} (Q(N, \boldsymbol{H}) - b)\nabla_\theta \log \pi_\theta(N|\boldsymbol{H})], \tag{9}$$

where b denotes a baseline function that does not depend on policy θ and could reduce the variance of the estimated gradient. In real implementation, we could update our network by Advantage Actor-Critic (A2C) algorithm, we set $b = V(s)$. Then we could approximate the above gradient by Monte Carlo method:

$$\theta^{t+1} = \theta^t - \frac{\mu_a}{B}\sum_{i=1}^{B}[R_{sum}(\boldsymbol{H}_i, N_i) - V_\phi(\boldsymbol{H}_i)\nabla_\theta \log \pi_\theta(N_i|\boldsymbol{H}_i)], \tag{10}$$

where μ_a is the learning rate of policy network, and B is the batch size.

Algorithm 1 Proposed algorithm

Initialize policy network parameters θ^0
Initialize Critic network parameters ϕ^0
for $t = 1 \rightarrow Epoch$ **do**
    Sample channel matrix $\{\boldsymbol{H}_1, \boldsymbol{H_2}, \cdots, \boldsymbol{H_B}\}$
    Calculate action N_i and probability of $\pi(N_i|\boldsymbol{H}_i; \theta^t)$
    Calculate baseline $V(\boldsymbol{H}_i; \phi^t)$
    Calculate gradient of policy network based (9)
    Calculate loss of Critic network based (12)
    Update policy network by (10)
    Update Critic network by gradient descent
end for

C. Value network

The structure of our value network is a multi-layer perception parameterized by ϕ. Based on Bellman optimality equation [17]:

$$V_{\phi_*}(\boldsymbol{H}) = \max_{N \in \mathcal{A}} Q_{\phi_*}(\boldsymbol{H}, N) = \max_{N \in \mathcal{A}} R_{sum}(\boldsymbol{H}, N). \quad (11)$$

When policy network converge to the optimal policy, then $V_{\phi_*}(\boldsymbol{H}) = R(\boldsymbol{H}, \pi_\theta(\boldsymbol{H}))$. Thus the Critic is trained with stochastic gradient descent on a mean squared error objective between its predictions and the actual spectral efficiency sampled by the most recent policy:

$$L(\phi) = \frac{1}{B} \sum_{i=1}^{B} |V_\phi(\boldsymbol{H}_i) - R(\boldsymbol{H}_i, N_i)|^2. \quad (12)$$

IV. EXPERIMENT RESULTS

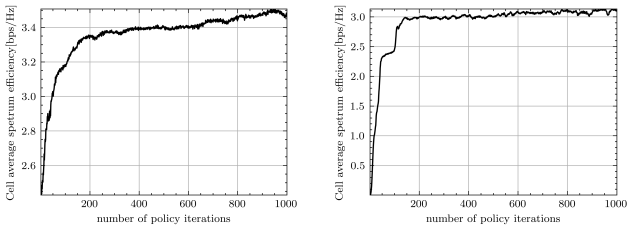
In this section, we provide the simulation results to illustrate the performance of DRL method and the greedy search method. Channel state information, e.g, channel matrices $\boldsymbol{H}$ is obtained from our 5G wireless simulation platform based on the 3GPP 3D-UMa channel model [19] with ray-tracing data as input. The simulation parameters are listed in Table I:

TABLE I
PARAMETERS OF SYSTEM SIMULATION

Channel model	3D-UMa
Number of antennas of BS	$M_t = 16$
Number of antennas of a user	$N_r = 1$
Transmit power of BS	$P_t = 0.25w$
Noise power	$N_0 = -195dB$
Bandwidth	$W = 10kHz$
User number	$L = 20, 30$
Batch size	$B = 256$
Actor learning rate	$\alpha_a = 1e^{-4}$
Critic learning rate	$\alpha_c = 1e^{-2}$
Number of training epochs	2000
Number of scheduling periods when training	$T = 800$

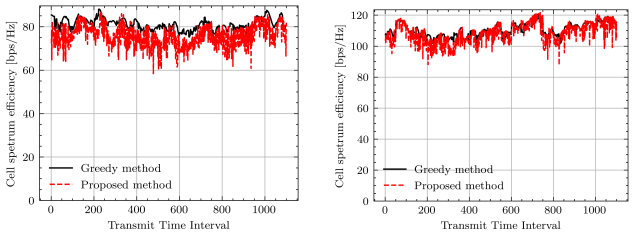
In each iteration, we define the performance indicator under policy π_θ as:

$$\eta(\theta) = \frac{1}{L} \frac{1}{T} \sum_{i=1}^{T} R_{sum}(\boldsymbol{H}_i, N_i). \quad (13)$$



(a) user number $K = 20$ (b) user number $K = 30$

Fig. 3. Learning curve between expected return and policy iteration step



(a) user number $K = 20$ (b) user number $K = 30$

Fig. 4. Performance comparison between DRL method and the greedy method

In our experiments, procedure is run on the server with 16 cores Xeon(R) Silver 4110 CPU. The ZF preceding scheme is adopted here and the power is equally allocated among all scheduled users: $P_i = \frac{P_t}{K}$. The GRU encoder and decoder is adopted by our policy network, ReLU activation function is adopted in the linear layer. A 4-layer perception architecture is adopted by our Critic network. The learning curves in Fig. 3 show the variation of the average cell spectral efficiency obtained by Equation (13) with the increase of the number of epochs for two sampled cells, which may experience dramatically different transmission environment with different user numbers. We trained the network two thousand epochs, which took about 8 hours, and the policy network was finally able to converge.

In testing stage, we generated additional $T = 1100$ TTI channel matrices in the same cell. As for our proposed RL method, the scheduling sequences could obtained when the channel matrices are directly sent to the pretrained policy neural network. Greedy algorithm is adopted by us as the comparison algorithm, which needs to iterate many times in each TTI to optimize a scheduling sequence. Fig. 4 shows the performance comparison of different TTI under greedy algorithm and pretrained policy neural network. In addition, we calculate the average system user SE over the whole TTI channel matrices and count the total running time under different algorithms. The detailed performance comparison are listed in Table II.

Although the performance of our reinforcement learning algorithm can not exceed the greedy algorithm, the running time of our algorithm are much faster than the greedy method in the cell with different user numbers on our server. In the cells with different number of users, the time consumption of our proposed algorithm is roughly the same. The reason is

TABLE II
PERFORMANCE COMPARISON

User number	SE performance		Running time	
	DRL	Greedy	DRL	Greedy
20	3.79(bps/Hz)	4.06(bps/Hz)	10.77s	105.19s
30	3.62(bps/Hz)	3.67(bps/Hz)	10.14s	271.96s

that the maximum number of streams in different user cells is the same: $S_{\max} = \min\{M_t, L\} = 16$. As for the greedy method, the running time increases with the user numbers in the cell. The greedy algorithm needs to continuously add new users until the system is saturated. Once a new user is added, it needs to recalculate the precoding matrix and SE. If in a certain TTI, all users are scheduled to the largest SE, then the greedy algorithm will be repeated $\frac{L^2+L}{2}$ times, while for DRL, we can get a suboptimal scheduling sequence through one forward operation of our policy network.

V. CONCLUSION

Focusing on both the capacity optimization and the reduction of scheduling complexity for single-cell downlink scheduling issue in massive MIMO systems, we proposed a RL-based Actor-Critic framework to provide the optimal scheduled user combination based on the time-varying channels. Pointer network is investigated as the policy network in our proposed Actor-Critic framework, which transfers the complicated selection issue among user combinations to a user sequential selection issue based on conditional probability. Compared with the most DL or RL-based scheduling methods, with the increase of the number of tranceiver antennas in massive MIMO system, our proposal can effectively simplify the network complexity and solve the convergence issues. Simulation results show that the performance of the proposed RL-based model is very close to that of the greedy method for the test data set. Moreover, our proposal is robust and effective with the increase of the number of antennas and users. In 5G communication system, the number of transmitting antennas and users will be much larger than that of the 4G wireless communication network. Therefore, the traditional greedy algorithm in 5G network requires a long decision-making time, which is difficult to meet the needs of 5G communication network. Our proposed algorithm runs an order of magnitude faster than the greedy algorithm. Multi-agent based RL networks will be studied for the performance optimization of multi-cell massive MIMO system in the future.

REFERENCES

[1] C.-X. Wang, F. Haider, X. Gao, X.-H. You, Y. Yang, D. Yuan, H. M. Aggoune, H. Haas, S. Fletcher, and E. Hepsaydir, "Cellular architecture and key technologies for 5g wireless communication networks," *IEEE communications magazine*, vol. 52, no. 2, pp. 122–130, 2014.

[2] G. Femenias and F. Riera-Palou, "Scheduling and resource allocation in downlink multiuser mimo-ofdma systems," *IEEE Transactions on Communications*, vol. 64, no. 5, pp. 2019–2034, 2016.

[3] J. Nam, J.-Y. Ahn, A. Adhikary, and G. Caire, "Joint spatial division and multiplexing: Realizing massive mimo gains with limited channel state information," in *2012 46th annual conference on information sciences and systems (CISS)*. IEEE, 2012, pp. 1–6.

[4] J. Shi, W. Wang, X. Yi, J. Wang, X. Gao, Q. Liu, and G. Y. Li, "Learning to compute ergodic rate for multi-cell scheduling in massive mimo," *IEEE Transactions on Wireless Communications*, 2020.

[5] Y. Yang, Y. Li, K. Li, S. Zhao, R. Chen, J. Wang, and S. Ci, "Decco: Deep-learning enabled coverage and capacity optimization for massive mimo systems," *IEEE Access*, vol. 6, pp. 23 361–23 371, 2018.

[6] G. Bu and J. Jiang, "Reinforcement learning-based user scheduling and resource allocation for massive mu-mimo system," in *2019 IEEE/CIC International Conference on Communications in China (ICCC)*. IEEE, 2019, pp. 641–646.

[7] N. Zhao, Y.-C. Liang, D. Niyato, Y. Pei, M. Wu, and Y. Jiang, "Deep reinforcement learning for user association and resource allocation in heterogeneous cellular networks," *IEEE Transactions on Wireless Communications*, vol. 18, no. 11, pp. 5141–5152, 2019.

[8] Y. S. Nasir and D. Guo, "Deep reinforcement learning for distributed dynamic power allocation in wireless networks," *arXiv preprint arXiv:1808.00490*, 2018.

[9] C. Huang, R. Mo, and C. Yuen, "Reconfigurable intelligent surface assisted multiuser miso systems exploiting deep reinforcement learning," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 8, pp. 1839–1850, 2020.

[10] V. R. Konda and J. N. Tsitsiklis, "Actor-critic algorithms," in *Advances in neural information processing systems*, 2000, pp. 1008–1014.

[11] O. Vinyals, M. Fortunato, and N. Jaitly, "Pointer networks," in *Advances in neural information processing systems*, 2015, pp. 2692–2700.

[12] I. Bello, H. Pham, Q. V. Le, M. Norouzi, and S. Bengio, "Neural combinatorial optimization with reinforcement learning," *arXiv preprint arXiv:1611.09940*, 2016.

[13] M. Nazari, A. Oroojlooy, L. V. Snyder, and M. Takáč, "Reinforcement learning for solving the vehicle routing problem," *arXiv preprint arXiv:1802.04240*, 2018.

[14] W. Kool, H. Van Hoof, and M. Welling, "Attention, learn to solve routing problems!" *arXiv preprint arXiv:1803.08475*, 2018.

[15] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *arXiv preprint arXiv:1706.03762*, 2017.

[16] N. Mazyavkina, S. Sviridov, S. Ivanov, and E. Burnaev, "Reinforcement learning for combinatorial optimization: A survey," *arXiv preprint arXiv:2003.03600*, 2020.

[17] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[18] R. J. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," *Machine learning*, vol. 8, no. 3-4, pp. 229–256, 1992.

[19] Y. Wenxin, L. Kai, Z. Mingtuo, L. Jian, and Y. Yang, "Research on adaptive neural network for 3d channel amplitude prediction based on ray-tracing data," *Journal of University of Chinese Academy of Sciences*, accepted, 2020. (in Chinese).