An Interference-Oriented 5G Radio Resource Allocation Framework for Ultradense Networks

Tao Peng, *Member, IEEE*, Yichen Guo, Yachen Wang, Gonglong Chen, Feng Yang, and Wei Chen

***Abstract*—To cope with the explosive growth in demands of wireless network, ultradense network (UDN) technology is widely adopted, which could increase the capacity of wireless network, but also bring severe intercell interference (ICI). However, existing solutions cannot work well in such complex scenarios, due to the limits of their mechanisms. To solve the problem, in this article, an interference-oriented radio resource allocation framework is proposed with multiple usages, including supplying precise, stable, and timely performance feedbacks, near perfect offline training, and high compatibility. As the use of the framework is derived from precise interference identification, a practical regression-based interference modeling algorithm is proposed to support the framework. With in-depth analysis of the mechanism of interference, the proposed algorithm could efficiently and accurately model interference between users using only data collected from operating wireless networks. Compared with the baseline algorithm, the proposed algorithm could reach the same accuracy with training time of two orders of magnitude shorter. To further show the advantages of the framework, a high-performance double-deep-*Q*-network-based resource allocation algorithm is also proposed. By integrating into the proposed framework, the proposed algorithm could coordinate ICI better, with 40% to 101% higher energy efficiency compared with baseline algorithms.**

***Index Terms*—Deep reinforcement learning (DRL), interference identification, nonlinear regression, resource allocation, ultradense network (UDN).**

I. Introduction

RECENT years have witnessed an era of prosperity in telecommunications industry, driven by evolution of the mobile communication technology and thriving demands of emerging mobile applications. To cope with the exponential growth of demands, ultradense network (UDN) incorporates network densification into the fifth generation (5G) mobile communication systems, which could better suit the Internet of Things (IoT) scenarios and increase the capacity of wireless networks. However, it also exposes the networks to severe intercell interference (ICI), which requires scheduling frameworks with complete interference identification and avoidance capability to further develop the UDN technology.

Existing interference mitigation methods, such as ICI coordination (ICIC) and enhanced ICIC (eICIC) cannot effectively solve the problem. First, their dependence on signaling exchange among neighboring cells causes unacceptable expenses on interference coordination. Second, their mechanisms, such as fractional frequency reuse (FFR) or soft frequency reuse (SFR) will result in limited frequency band access. Besides, current single-cell scheduling algorithms cannot effectively apply to multiple cells. In conclusion, existing single-cell scheduling algorithms with interference mitigation methods are no longer suitable for ever-increasing deployment density of wireless networks.

Beyond the existing framework, numerous works have tried to effectively solve the multicell subchannel allocation problem. Within the scope of optimization, the problem is solved with either convex-optimization-based, or heuristic approaches. The convex-optimization-based approaches [1]–[4] could get global optima of the convex problem it solves. However, the subchannel allocation problem is a nonconvex mixed-integer programming problem. To apply these approaches, primal problems must be converted into solvable problems. Kuang *et al.* [1] and Yu and Yin [3] relaxed some variables in the primal problem, while [2] converts the primal problem into a known solvable problem. However, optima of the converted problem are usually not those of the primal one. Moreover, solving the converted problem is still computationally intensive. For UDN, these approaches are not feasible.

While heuristic approaches [5]–[12] are less expensive in computational cost. Despite the mild loss of optimality, they could still perform well. The subchannel allocation problem could be transformed into a combinational optimization problem, which makes the Hungarian method [13] suitable, as in [6] and [9]. While [5], [7], [11], [12] leverage graph theory to solve the problem. And, [8] uses the local search method to reduce complexity. In addition, game theory is feasible, such as auction method [10] and coalition formation game [12].

Artificial-intelligence (AI)-based approaches are also heuristic. Their ability of autonomous problem-solving suits UDN where manual manipulation of the problem becomes impossible. Two kinds of AI algorithms are commonly used: 1) deep learning (DL) and 2) reinforcement learning (RL) [14]. DL majorly operates in a supervised manner, which learns solutions by training with existing data sets. As, Sun *et al.* [15] used a supervised-DL-based algorithm to solve the power

Manuscript received 3 November 2021; revised 12 March 2022 and 28 April 2022; accepted 6 June 2022. Date of publication 16 June 2022; date of current version 7 November 2022. This work was supported in part by the Beijing Natural Science Foundation under Grant L192002, and in part by the China National Key Research and Development Program under Grant 2020YFB1808000. *(Corresponding author: Tao Peng.)*

The authors are with the Key Laboratory of Universal Wireless Communications, Ministry of Education, Beijing University of Posts and Telecommunications, Beijing 100876, China (e-mail: pengtao@bupt.edu.cn; guoyichen@bupt.edu.cn; 1475324023@qq.com; secpros@163.com; yangfengee04@gmail.com; chenweicmri@139.com).

Digital Object Identifier 10.1109/JIOT.2022.3183930

allocation problem. The result is promising, but generating data sets is burdensome. Thus, an unsupervised DL algorithm is utilized in [16]. It iteratively uses the feedback of the last output to train itself. In this way, it does not need pregenerated data set, but more time to train.

On the contrary, RL is purely unsupervised and autonomous, and could adjust its strategy when the environment changes, which makes RL algorithms feasible in decision making. One of the most widely used RL algorithms is deep RL (DRL), which combines DL and RL. Xu *et al.* [17] used centralized DRL controller to reduce power consumption. While in [18], the proposed DRL algorithm is trained centralizedly, and distributedly used by every base station. However, the trained algorithm does not target any specific base station, resulting in a loss of optimality. Lee *et al.* [19] used the Lagrange method to model the wireless network in a circumstance-independent way. Thus, the trained algorithm does not need to completely retrain to perform decently as the environment changes. Zhao *et al.* [20] and Gu *et al.* [21] used multiagent DRL to optimize the network. However, it may induce extra information exchange overhead, and convergence is not guaranteed.

Despite the performance, most algorithms assume full and perfect interference information, which cannot stand. To avoid this problem, in [22], measured signal-to-interference-plus-noise ratio (SINR) data are directly used, regardless of the imperfection of data caused by superpositions of multiple interference. While [23] takes service outage probability into consideration, but makes the resource allocation problem more complex.

There are also works that add extra components to resource allocation framework to extract information from the real network. With the help of machine learning (ML), [12] preprocesses environment observations with two deep neural networks (DNN). The performance is descent, but long-lasting training makes it impractical. Our previous works [11], [24] give better practice. By mining a huge amount of data collected from wireless networks using DNN, interference relationships could be obtained. However, they also take too much time in training. In summary, existing alleviating methods work, but not perfectly.

In this article, to cope with the crisis of interference information inaccessibility, an interference-oriented 5G radio resource allocation framework (IRAF) is proposed, with an interference modeling algorithm proposed to complement the ability of interference identification of the framework. Further, to better exploit the framework's merits of providing precise interference information, accurate performance feedbacks, and near-perfect offline training, a double deep Q-learning (DDQL)-based resource allocation algorithm is also proposed.

In this article, five algorithms are selected as baseline of the proposed resource allocation algorithm.

1) *"MGS" [25]:* The "MGS" algorithm is a user-centric conflict-graph-based resource allocation algorithm aimed at solving coalition formation game, which is adopted to formulate the resource allocation problem.
2) *"TIC" [26]:* The "TIC" algorithm solves coalition formation game that models the resource allocation problem distributedly but cooperatively.
3) *"PF" [27]:* The "PF" algorithm is the well-known proportional fairness algorithm.
4) *"CIAQ" [28]:* The "CIAQ" algorithm is a DL-based algorithm. The algorithm builds a DNN to reproduce the alternative direction method of multipliers (ADMM) iterative optimization procedure.
5) *"GLIS" [16]:* The "GLIS" algorithm is also based on ML, which uses discretized geographic location information of users and base stations to train a spatial convolution filter and generate the coarse interference status. And, a DNN allocates resources based on it.

And, the only algorithm [11] running in the analogous way as the proposed interference modeling algorithm is chosen as its baseline.

The main contributions of this article are summarized as follows.

1) *Interference-Oriented Versatile Resource Allocation Framework:* For the first time, an interference-oriented intelligent resource allocation framework with multiple usages is proposed. Using the ability to identify interference, the framework could supply precise interference information and accurate performance feedback to resource allocation algorithms to support their reliable operation and enable interference awareness as no existing works could do. In addition, a virtual environment almost identical to the actual network could be constructed of the interference information, which can support learning-based algorithms to train offline and be put online directly without extra online training as existing simulation-based offline trainers require.
2) *Precise and Fast Interference Modeling Algorithm:* To support the ability of interference identification of the proposed framework, for the first time, this article proposed an interference modeling algorithm with high practicality. Based on in-depth analysis of the mechanism of interference in wireless network, the proposed algorithm models the interference using a nonlinear function with each unknown parameter being the interference between an interested user and one of its interfering users. The parameter-solving process only requires data collected from the operating wireless network, and does not need extra hardware or radio resources. Besides, the algorithm is computationally friendly and of superior performance. Compared with [11], the proposed algorithm requires a shorter training time and less volume of data by two orders of magnitude to model the interference without loss of precision.
3) *Compact and Fast Intelligent Resource Allocation Algorithm:* In this article, a DDQL-based intelligent resource allocation algorithm is proposed to take full advantage of the framework. With precise interference information and stable performance feedback of the framework, the proposed algorithm largely reduces the

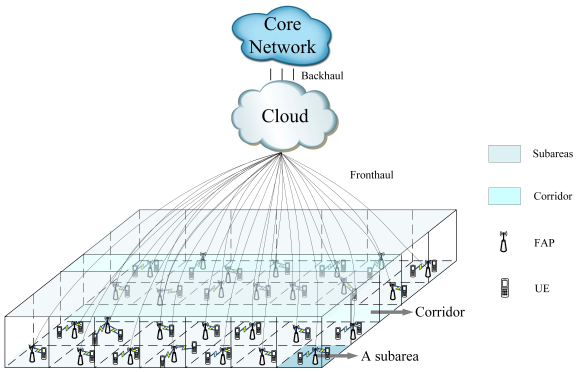

Fig. 1. Dual-strip UDN model.

scale of the neural network by increasing the granularity of the output from complete resource allocation scheme of a transmission time interval (TTI) to resource allocation decision of one femto access point (FAP), which had never been achieved before. Subsequently, the proposed algorithm could train and evaluate faster, and perform better, as verified by simulation results. Besides, as a learning-based algorithm, the proposed algorithm could also enjoy the near-perfect offline training provided by the framework, making it an excellent example to holistically show the merits of the framework.

This article is organized as follows. The brief introduction and the literature review are in Section I. The system model is described in Section II. Section III shows the structure of the resource allocation framework, and Section IV elaborates on the proposed interference modeling algorithm. The resource allocation algorithm is proposed in Section V. Finally, the simulation results and analyses are presented in Section VI, and in Section VII, a conclusion is drawn.

II. System Model

In the 5G era, to provide quality services in crowded places like stadiums, shopping malls and theaters, or in certain scenarios that deploy a massive number of IoT devices like smart factories, deploying FAPs densely is a simple, yet effective solution, but it also induces heavy ICI.

In this article, a dual-strip model is considered, as shown in Fig. 1. In the model, two strips of rooms are placed on each side of the corridor. Each strip has the same number of square rooms with randomly placed one FAP and N_p user equipment (UE) in each room. Thus, in the model, there are M cells and FAPs (In the remainder of this article, cell and FAP are used interchangeably), and $N = MN_p$ UEs (user and UE are also used interchangeably). The set of cells is denoted as $\mathbf{C} = \{C_1, C_2, \ldots, C_M\}$, and the UE set is denoted as $\mathbf{U} = \{U_1, U_2, \ldots, U_N\}$.

The system adopts cloud radio access network (C-RAN) architecture, where all the FAPs are connected to a centralized controller in the cloud [29]. This architecture has natural advantages in handling ICI coordination and multicell resource allocation. The system bandwidth is B, consists with K resource blocks (RBs), and each RB has a bandwidth of W.

The uplink transmission is considered in this article, which adopts a single-carrier frequency division multiple access (SC-FDMA) scheme. Therefore, users within the same cell cannot interfere with each other. However, it is not practical to allocate orthogonal resources to all users, due to insufficiency of radio resources [30]. Thus, resources must be reused, leading to severe ICI.

In this article, the FAPs could utilize all the RBs in radio resource set $\mathbf{R} = \{RB_1, RB_2, \ldots, RB_K\}$ to serve associated users. The uplink SINR of user U_i associated with FAP C_j at RB_k is

$$\mathrm{SINR}_i^{(k)} = \frac{P_i G_{i,j}^{(k)}}{\sum_{U_n \in \mathbf{U} \setminus \mathbf{U}_j} \alpha_{n,m_n}^{(k)} P_n G_{n,j}^{(k)} + \sigma^2} \tag{1}$$

where P_i is the transmission power of user U_i, $\mathbf{U}_j$ is the set of users in cell C_j, and $\alpha_{n,m_n}^{(k)}$ is the resource allocating status for user U_n in cell C_{m_n} at RB_k, with m_n being the subscript of the cell U_n associated with. And σ^2 is the variance of zero-mean addictive white Gaussian noise (AWGN), and $G_{i,j}^{(k)}$ is the channel gain between user U_i and FAP C_j at RB_k.

Apart from the uplink SINR given in (1) and Shannon's capacity formula, the transmission rate is also affected by the modulation and coding scheme (MCS). Thus, the upper bound of SINR, $\mathrm{SINR}_{\max}$, is applied and set to the SINR threshold of the highest channel quality indicator (CQI), which leads to the highest MCS. Additionally, in this article, all the RBs are considered homogeneous. Thus, the uplink transmission data rate of user U_i per RB is

$$R_i = W \log_2(1 + \min(\mathrm{SINR}_i, \mathrm{SINR}_{\max})). \tag{2}$$

III. Interference-Oriented 5G Radio Resource Allocation Framework

To meet the need of radio resource management approaches with strong capability of interference avoidance and coordination, in this section, an interference-oriented centralized 5G radio resource allocation framework is proposed, consisting of three major parts: 1) environment; 2) performance reasoner; and 3) resource allocator. The structure of the IRAF is shown in Fig. 2.

A. Environment

The environment part refers to actual wireless networks. In the IRAF, the environment takes the resource allocation schemes of the entire TTI. After the transmission is finished, it will feed measured SINRs of every allocated user at each RB back to the history data set, which will be organized for each user.

B. Performance Reasoner

The performance reasoner part is one of the major parts of the IRAF, bridging the actual wireless network and the resource allocator with its outstanding power of interference identification from the interference modeling component using data provided by the history data set.

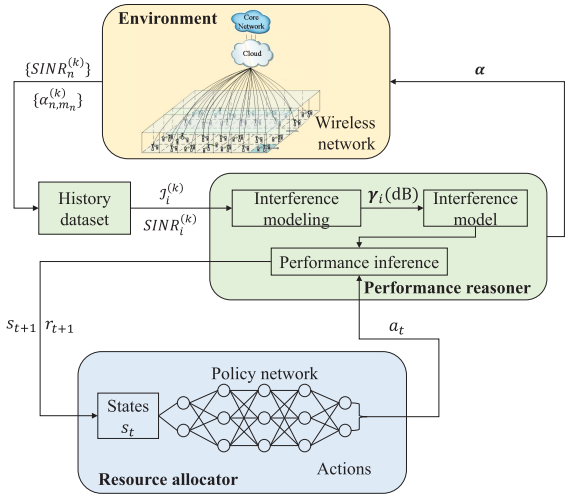

Fig. 2. Structure of the IRAF with the interference modeling algorithm proposed in Section IV and the resource allocation algorithm proposed in Section V integrated.

The crucial functions of the performance reasoner with their requirements for the interference modeling component are summarized as follows.

1) *Isolating Environment Fluctuations:* The fluctuation of the environment leads to high deviation in feedbacks, which affects the performance of the resource allocator. To isolate the fluctuation, the IRAF *requires* the interference modeling component to provide stable interference models, which leads to stable performance inference and better performance of the resource allocator.
2) *Providing Precise Performance Inference and Enabling Interference Awareness:* The IRAF *requires* the interference modeling component to provide interference models with high precision and speed to support the performance inference component to precisely infer performance. Thus, the accurate interference model could facilitate traditional interference coordination methods, such as ICIC or eICIC and resource allocation algorithms to better identify and coordinate users with severe interference.
3) *Acting as Offline Trainer:* Traditional offline training routines use a simulator to generate scenarios that are *similar* to the target wireless network to train the learning-based resource allocator offline. Thus, additional online training is needed to mitigate the performance loss caused by the difference between the simulated scenarios and the target network. However, in the IRAF, the resource allocator could be trained offline with a scenario almost *identical* to the actual wireless network built by the performance reasoner using precise interference models. In this way, offline training is sufficient for the resource allocator to operate normally, thus no online training is needed.

To meet the requirements, the algorithm proposed in Section IV is the ideal choice.

C. Resource Allocator

In the IRAF, the resource allocator interacts with the performance reasoner to obtain information and allocate resources. The results of allocation are buffered in the performance reasoner. Upon the completion of resource allocation of the current TTI, the performance reasoner submits the allocation results to the environment. There is no additional requirement for algorithms the allocator uses, showing the great compatibility of the IRAF. However, to better exploit the IRAF, a newly designed algorithm is preferred, as the one proposed in Section V.

IV. Nonlinear-Regression-Based Interference Modeling Algorithm

The IRAF proposed in Section III resolves the interference information availability crisis with the ability to precisely identify interference. However, to the best of our knowledge, there is not any interference modeling algorithm that could support practical usage of the IRAF. In this section, a low-cost high-precision regression-based interference modeling algorithm is proposed, which could perfectly enable the IRAF to function as expected.

A. Problem Formulation

In the uplink direction of wireless network, the signal power of interested user U_i received by its associated FAP C_{m_i} is

$$S_{i,m_i} = P_i G_{i,m_i} \tag{3}$$

where G_{j,m_i} is the channel gain between user U_j and FAP C_{m_i}.

And, the interference power of arbitrary interference user U_j received by FAP C_{m_i} is

$$I_{j,m_i} = P_j G_{j,m_i}. \tag{4}$$

Thus, the SINR of the interested user U_i could be expressed as

$$\begin{aligned} \text{SINR}_i &= \frac{P_i G_{i,m_i}}{\sum_{U_n \in \mathbf{U} \setminus \mathbf{U}_{m_i}} \alpha_{n,m_n} P_n G_{n,m_i} + \sigma^2} \\ &= \frac{S_{i,m_i}}{\sum_{U_n \in \mathbf{U} \setminus \mathbf{U}_{m_i}} \alpha_{n,m_n} I_{n,m_i} + \sigma^2} \end{aligned} \tag{5}$$

where α_{n,m_n} denotes the resource allocation status of user U_n and its associated FAP C_{m_n} for the same RB. By solving S_{i,m_i}, I_{j,m_i}, and σ^2, the SINR could be inferred for any given resource allocation status.

Taking the reciprocal of both sides of (5), the unknown parameters are in the numerator

$$\tilde{\gamma}_i = \frac{\sum_{U_n \in \mathbf{U} \setminus \mathbf{U}_{m_i}} \alpha_{n,m_n} I_{n,m_i} + \sigma^2}{S_{i,m_i}} \tag{6}$$

where $\tilde{\gamma}_i = \text{SINR}_i^{-1}$ is the reciprocal of the SINR of user U_i, named interference-plus-noise-to-signal ratio (INSR). Furthermore, by splitting the numerator, (6) becomes

$$\tilde{\gamma}_i = \sum_{U_n \in \mathbf{U} \setminus \mathbf{U}_{m_i}} \alpha_{n,m_n} \tilde{\gamma}_{i,n} + \tilde{\gamma}_{i,\sigma^2} \tag{7}$$

Entry #	TTI #	RB#	C_2			…	C_M			SINR (dB)
			$U_{\lvert\mathbb{U}_1\rvert+1}$	…	$U_{\lvert\mathbb{U}_1\rvert+\lvert\mathbb{U}_2\rvert}$	…	…	U_{N-1}	U_N	
1	1	RB_3	1	…	0	…	…	1	0	3.80497
2		RB_4	0	…	1	…	…	0	0	10.856
3	3	RB_{K-2}	0	…	0	…	…	0	1	19.0655
4		RB_{K-1}	0	…	0	…	…	0	0	73.1232
5		RB_K	1	…	0	…	…	1	0	4.65886
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
$\lvert\mathcal{D}\rvert$-3	t-2	RB_6	0	…	0	…	…	0	0	11.0561
$\lvert\mathcal{D}\rvert$-2		RB_7	0	…	0	…	…	0	1	23.2665
$\lvert\mathcal{D}\rvert$-1	t-1	RB_{12}	0	…	1	…	…	0	0	12.2965
$\lvert\mathcal{D}\rvert$	t	RB_1	0	…	0	…	…	0	0	74.5114

Fig. 3. Example of training data set for user U_1.

where $\tilde{\gamma}_{i,n} = I_{n,m_i}/S_{i,m_i}$ is the reciprocal of the signal-to-interference ratio (SIR), named interference-to-signal ratio (ISR), of the interested user U_i with only one active interference user, U_n. Similarly, $\tilde{\gamma}_{i,\sigma^2} = \sigma^2/S_{i,m_i}$ is the reciprocal of the signal-to-noise ratio (SNR) of the interested user U_i, named noise-to-signal ratio (NSR). With (7), by solving the ISRs and the NSR, the INSR (and thus, SINR) of the interested user could be easily obtained with given resource allocation status. Further, the ISRs and the NSR contain interference information of the interested user.

B. Data Organization

In the training process of the nonlinear-regression-based interference modeling algorithm (NLRA), a series of per-user data sets consisting of entries of interference status of RB occupied by the specific interested user and the SINR of the user at the corresponding RB measured by the associated FAP is required, as depicted in Fig. 3, which could be directly collected from operating wireless network.

Taking U_i as the interested user, the data set is denoted as $\boldsymbol{\mathcal{D}}_i$. And, the interference status of the kth entry $\boldsymbol{\mathcal{I}}_i^{(k)}$ in $\boldsymbol{\mathcal{D}}_i$ is the resource allocation status $\boldsymbol{\alpha}^{(k)}$ of all users but the ones in cell C_{m_i}. And the measured SINR, expressed in decibel, is denoted as $\text{SINR}_{i,\text{real}}^{(k)}$, corresponding to $\boldsymbol{\mathcal{I}}_i^{(k)}$.

C. Algorithm Elaboration

As shown in (7), it is similar to linear regression model with the interference status as explanatory variables and the ISRs and the NSR as unknown parameters. By adding the error term ϵ_{LR}, (7) becomes standard linear regression model

$$\tilde{\gamma}_i^{(k)} = \tilde{\gamma}_{i,\sigma^2} + \sum_{\alpha_{n,m_n} \in \boldsymbol{\mathcal{I}}_i^{(k)}} \alpha_{n,m_n} \tilde{\gamma}_{i,n} + \epsilon_{LR} \tag{8}$$

where $\tilde{\gamma}_i^{(k)} = 1/\text{SINR}_{i,\text{real}}^{(k)}$. With least-squares (LS) method, the unknown parameters could be solved analytically. But the solved parameters are the maximum-likelihood estimation of their exact values if and only if the error term ϵ_{LR} fits the Gaussian distribution [31]. However, multiplicative short-term fading cannot be captured by the additive error term [32], resulting in the non-Gaussian distribution for ϵ_{LR}. Besides, the range of the measured SINRs spans a few dozens of decibels, which corresponds to a range of nearly ten orders of magnitude of $\tilde{\gamma}_i^{(k)}$, troubling the LS solver. The same problem exists in $\tilde{\boldsymbol{\gamma}}_i$, where the gap between $\tilde{\gamma}_{i,\sigma^2}$ and $\tilde{\gamma}_{i,n}$ could also reach around ten orders of magnitude.

Thus, a nonlinear regression model is introduced to solve the problems above

$$\text{SINR}_{i,\text{real}}^{(k)} = -10\lg\left(10^{-\gamma_{i,\sigma^2}(\text{dB})/10} + \sum_{\alpha_{n,m_n} \in \boldsymbol{\mathcal{I}}_i^{(k)}} \alpha_{n,m_n} 10^{-\gamma_{i,n}(\text{dB})/10}\right) + \epsilon_{NLR} \tag{9}$$

with the objective function being

$$f_{NLR}\left(\boldsymbol{\mathcal{I}}_i^{(k)}; \boldsymbol{\gamma}_i(\text{dB})\right) = -10\lg\left(10^{-\gamma_{i,\sigma^2}(\text{dB})/10} + \sum_{\alpha_{n,m_n} \in \boldsymbol{\mathcal{I}}_i^{(k)}} \alpha_{n,m_n} 10^{-\gamma_{i,n}(\text{dB})/10}\right) \tag{10}$$

where $\boldsymbol{\gamma}_i(\text{dB}) = \{\gamma_{i,\sigma^2}(\text{dB})\} \cup \{\gamma_{i,n}(\text{dB}) | \alpha_{n,m_n} \in \boldsymbol{\mathcal{I}}_i^{(k)}\}$, $\gamma_{i,\sigma^2}(\text{dB}) = -10\lg\tilde{\gamma}_{i,\sigma^2}$, and $\gamma_{i,n}(\text{dB}) = -10\lg\tilde{\gamma}_{i,n}$ are the unknown parameters set, the SNR of user U_i, and the SIRs between U_i and interfering user U_n, all in decibel. Taking -10 times of logarithms of both sides of (8) makes the inferred SINR presented in decibel, which is in consistence with that in data set, and is more useful than its anti-logarithmic form. Furthermore, the multiplicative fading becomes additive, thus ϵ_{NLR} could capture them effectively. Besides, the range of the measured SINRs in decibel spans less than one hundred, which could be easily handled by the solver. And, that is also the reason all the unknown parameters are turned into decibel.

With effective utilization of data, the LS method requires fewer data to solve problems, while it is also exposed more to effect of abnormal data. In wireless networks, however, abnormal data are inevitable due to short-term fading. Thus, in the NLRA, the Huber loss (11) is introduced to mitigate the effect

$$\rho(x) = \begin{cases} x^2, & 0 \le |x| \le 1 \\ 2|x| - 1, & |x| > 1. \end{cases} \tag{11}$$

With the Huber loss, the solved parameters are

$$\boldsymbol{\gamma}_i^*(\text{dB}) = \arg\min_{\boldsymbol{\gamma}_i(\text{dB})} \sum_{k=1}^{|\boldsymbol{\mathcal{D}}_i|} C^2 \rho\left(\frac{\left\|\text{SINR}_{i,\text{real}}^{(k)} - f_{NLR}\left(\boldsymbol{\mathcal{I}}_i^{(k)}; \boldsymbol{\gamma}_i(\text{dB})\right)\right\|^2}{C^2}\right) \tag{12}$$

where $\boldsymbol{\gamma}_i^*(\text{dB}) = \{\gamma_{i,\sigma^2}^*(\text{dB})\} \cup \{\gamma_{i,n}^*(\text{dB}) | \alpha_{n,m_n} \in \boldsymbol{\mathcal{I}}_i^{(k)}\}$, and C controls the margin between normal and abnormal data.

It is worthwhile to mention that the solution of the NLRA is obtained by numerical approaches like trust-region-based methods [33], rather than a guaranteed analytical solution in

Algorithm 1 NLRA

Input: Training dataset $\boldsymbol{\mathcal{D}} = \{\boldsymbol{\mathcal{D}}_1, \ldots, \boldsymbol{\mathcal{D}}_N\}$
Output: Interference model $\boldsymbol{\gamma}^*(\text{dB}) = \{\boldsymbol{\gamma}_1^*(\text{dB}), \ldots, \boldsymbol{\gamma}_N^*(\text{dB})\}$
1: **for** each user $U_n \in \mathbf{U}$ **do**
2: Take dataset $\boldsymbol{\mathcal{D}}_n$ from $\boldsymbol{\mathcal{D}}$
3: Solve (12) with trust-region-based method[33] and obtain $\boldsymbol{\gamma}_n^*(\text{dB})$
4: Push $\boldsymbol{\gamma}_n^*(\text{dB})$ into $\boldsymbol{\gamma}^*(\text{dB})$
5: **end for**
6: **return** $\boldsymbol{\gamma}^*(\text{dB})$

Algorithm 2 SINR Prediction Algorithm With Interference Model

Input: Resource allocation status $\boldsymbol{\alpha} = \{\alpha_{n,m_n} | n \in \{1, \ldots, N\}\}$, user with SINR to be predicted U_i, interference model $\boldsymbol{\gamma}^*(\text{dB})$
Output: Predicted SINR for U_i as $\text{SINR}_{i,\text{pred}}$
1: $\boldsymbol{\mathcal{I}}_i = \boldsymbol{\alpha}$
2: Remove entries of all users within the same cell as U_i in $\boldsymbol{\mathcal{I}}_i$
3: Take $\boldsymbol{\gamma}_i^*(\text{dB})$ from $\boldsymbol{\gamma}^*(\text{dB})$
4: Solve $\text{SINR}_{i,\text{pred}}$ by taking $\boldsymbol{\gamma}_i^*(\text{dB})$ and $\boldsymbol{\mathcal{I}}_i$ into (13)
5: **return** $\text{SINR}_{i,\text{pred}}$

linear regression problems, due to nonlinearity. The proposed NLRA is summarized in Algorithm 1.

Using the solved parameters, the SINR of arbitrary user U_i could be predicted with interference status $\boldsymbol{\mathcal{I}}_i$ derived from the resource allocation scheme

$$\text{SINR}_{i,\text{pred}} = -10\lg\left(10^{-\gamma^*_{i,\sigma^2}(\text{dB})/10} + \sum_{\alpha_{n,m_n}\in\boldsymbol{\mathcal{I}}_i} \alpha_{n,m_n} 10^{-\gamma^*_{i,n}(\text{dB})/10}\right). \quad (13)$$

The SINR prediction process described above is summarized in Algorithm 2.

V. DDQL-Based Resource Allocation Algorithm

As the interference information acquisition problem is alleviated by the IRAF, most centralized resource allocation algorithms could be used in the resource allocator. However, as they are not specially designed for the framework, not all the merits of the IRAF could be shown. In this section, a DDQL-based resource allocation algorithm (DRAA) aimed at solving the general sum-rate maximization (GSRMax) problem with high performance and low computational complexity is proposed to better utilize the IRAF by exquisitely designing the algorithm.

A. Problem Formulation

The GSRMax problem is formulated as follows:

$$\max_{\boldsymbol{\alpha}} \sum_{k=1}^{K}\sum_{n=1}^{N} \alpha_{n,m_n}^{(k)} w_n R_n^{(k)} \quad (14a)$$

subject to

$$\sum_{n'\in\mathbf{U}_m} \alpha_{n',m}^{(k)} \le 1 \quad \forall m \in \{1, 2, \ldots, M\} \quad (14b)$$

$$\sum_{n''\in\mathbf{U}\setminus\mathbf{U}_m} \alpha_{n'',m}^{(k)} = 0 \quad \forall m \in \{1, 2, \ldots, M\} \quad (14c)$$

$$\alpha_{n,m_n}^{(k)} \in \{0, 1\} \quad \forall n \in \{1, 2, \ldots, N\} \quad \forall k \in \{1, 2, \ldots, K\} \quad (14d)$$

$$\sum_{k=1}^{K} \alpha_{n,m_n}^{(k)} P_n \le P_{\max} \quad \forall n \in \{1, 2, \ldots, N\} \quad (14e)$$

where $\boldsymbol{\alpha} = \{\alpha_{n,m}^{(k)} \in \{0,1\} | m \in \{1, 2, \ldots, M\}, n \in \{1, 2, \ldots, N\}, k \in \{1, 2, \ldots, K\}\}$ is resource allocation indicator matrix, with the constraint (14c) to retain user association relationships. And, w_n is the weight of user U_n. Further, $\boldsymbol{w} = \{w_1, \ldots, w_n\}$ is denoted as the set of weights.

Due to orthogonality among every RB, resource allocation in each RB is independent. Thus, problem (14) becomes

$$\sum_{k=1}^{K} \max_{\boldsymbol{\alpha}^{(k)}} \sum_{n=1}^{N} \alpha_{n,m_n}^{(k)} w_n R_n^{(k)} \quad \text{subject to (14b), (14c), (14d), (14e)} \quad (15)$$

where $\boldsymbol{\alpha}^{(k)} = \{\alpha_{1,m_1}^{(k)}, \alpha_{2,m_2}^{(k)}, \ldots, \alpha_{n,m_n}^{(k)}, \ldots, \alpha_{N,m_N}^{(k)}\}$ is the resource allocation indicator vector showing the resource allocation status of RB_k.

In this article, for w_n, two specific cases are considered as follows.

1) $w_n = 1\ \forall n \in \{1, 2, \ldots, N\}$: All users share the same weight. Thus, the problem becomes the pure sum-rate maximization (SRMax) problem.
2) $w_n = \bar{R}_{n,T}^{-1}\ \forall n \in \{1, 2, \ldots, N\}$: The weight of user U_n at Tth TTI is inversely proportional to the user's long-term average transmission rate $\bar{R}_{n,T}$ (considering the uniformity and the deduction below, $w_{n,T}$ is written as w_n). Thus, the problem, named weighted SRMax (WSRMax), takes fairness into consideration. The long-term average transmission rate is

$$\bar{R}_{n,T} = \sum_{k=1}^{K} \bar{R}_{n,T}^{(k)} \quad (16)$$

where

$$\bar{R}_{n,T}^{(k)} = \lambda R_{n,T}^{(k)} + (1-\lambda)\bar{R}_{n,T-1}^{(k)} \quad (17)$$

is the time-weighted average rate of user U_n at RB_k at TTI T, and $\lambda \in (0, 1)$ is the weighting factor.

The problem (15) is an integer programming problem, which is NP-hard and highly nonconvex. In this article, by coordinating with the IRAF, RL is introduced to solve such a problem with low complexity and high optimality.

B. Introduction to Double Deep Q-Learning

RL algorithms could control the environment to optimally reach the target through exploring and exploiting [34].

The basic structure of the RL system, depicted in Fig. 4, consists of two entities (environment and agent). Three types of information (state s_t and action a_t at time step t, and reward

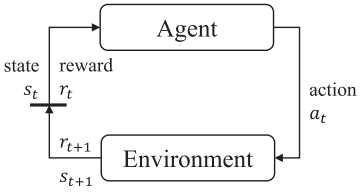

Fig. 4. Basic structure of RL.

r_{t+1} obtained at the next time step) are exchanged between the entities. An RL system is used to solve a problem that could be modeled as the Markov decision process (MDP) with a four-tuple $< \mathbf{S}, \mathbf{A}, \mathbf{P}, \mathbf{r} >$, where

1) $\mathbf{S}$ is the set of all states;
2) $\mathbf{A}$ is the set of all actions;
3) $\mathbf{P} = \{P_a(s, s')|a \in \mathbf{A}, s \in \mathbf{S}, s' \in \mathbf{S}\}$ is the set of transition probabilities, where $P_a\ (s, s'\ ) = \Pr\{s_{t+1} = s'\ |s_t = s,\ a_t = a\}$ is the probability of taking action a in state s at time t leading to the state transitioned to s' at time $t + 1$;
4) $\mathbf{r} = \{r_a(s, s')|a \in \mathbf{A}, s \in \mathbf{S}, s' \in \mathbf{S}\}$ is the set of immediate rewards received after state transitioned from s to s' due to action a.

The objective of RL algorithms is to determine the optimal policy π to solve the target problem. Q-learning is a value-based RL algorithm, which solves the optimal policy by maximizing the expected return at given state and action (i.e., the value of Q-function). For complex scenarios, fitting the Q-function with DNN is feasible

$$Q(s, a; \theta) \approx Q_\pi(s, a) = E_\pi[G_t|s_t = s, a_t = a] \tag{18}$$

where θ is the weights of DNN, known as policy network, and G_t is the return at time t. This algorithm is called deep Q-learning (DQL).

Considering training stability [35], a target network θ^- which is periodically synchronized with the policy network and an experience replay memory storing history experiences are added. Further, to avoid Q-value over-estimation in DQL, in this article, DDQL [36] is utilized. The Bellman equation of the DDQL is

$$Q_{\text{target}}\left(r, s'; \theta, \theta^-\right) = r + \gamma Q\left(s', \arg\max_{a'} Q\left(s', a'; \theta\right); \theta^-\right) \tag{19}$$

where $\gamma \in [0, 1]$ is the discount factor and the loss function is expressed as follows:

$$L\left(\mathbf{D}; \theta, \theta^-\right) = \sum_{(s,a,r,s') \in \mathbf{D}} \left(Q_{\text{target}}\left(r, s'; \theta, \theta^-\right) - Q(s, a; \theta)\right)^2 \tag{20}$$

where $\mathbf{D}$ is a batch of experiences, and each experience tuple (s, a, r, s') stores the current state s, the action taken a, the reward r, and the next state s'.

C. MDP Formulation

There are two fundamental requirements for the RL problem to be modeled as MDP. First, all states are required to have the Markov property

$$\Pr\{s_{t+1}|s_t\} = \Pr\{s_{t+1}|s_1, \ldots, s_t\}. \tag{21}$$

And second, results of all actions should be deterministic [34]. With the IRAF, the second requirement is met. However, the first requirement needs delicate design of the MDP.

1) SRMax Problem: To maximize the sum rate of the system in a TTI. In the DRAA, the decision-making process is serialized by making decision on one FAP at an RB. Thus, each time step is correlated to each other. To decouple those steps, the MDP is constructed as follows.

1) *States:* The states are constructed with two parts. The first part is the occupation status $\mathbf{O} = [O_1, \ldots, O_n, \ldots, O_N]$, where $O_n \in \{0, 1\}$ is the occupation indicator. If the current RB has been allocated to user U_n, $O_n = 1$; otherwise, $O_n = 0$. The second part is the decision status $\mathbf{F} = [F_1, \ldots, F_m, \ldots, F_M]$, where $F_m \in \{0, 1\}$ is the decision indicator. If an FAP C_m has made decision, $F_m = 0$; otherwise, $F_m = 1$. Decisions are linked to actions. If any action related to an FAP is taken, the action is the decision the FAP made. Thus, a state $s = [\mathbf{O}, \mathbf{F}] = [O_1, \ldots, O_N, F_1, \ldots, F_M]$.
2) *Actions:* There are two types of actions. The first one is FAP C_{m_i} allocates the current RB to user U_i, denoted as a^+_{i,m_i}. If a^+_{i,m_i} is taken, the state transitions as O_i becomes one and F_{m_i} becomes zero. The other one is that FAP C_m decides not to allocate the current RB to any associating user, denoted as a^0_m. If a^0_m is taken, the state transitions as F_m turns into zero. Thus, $\mathbf{A} = \{a^+_{1,1}, \ldots, a^+_{i,m_i}, \ldots, a^+_{N,M}, a^0_1, \ldots, a^0_m, \ldots, a^0_M\}$. The construction of the action set already makes sure that the constraint (14c) and (14d) are met. Note that both kinds of actions of FAP C_m are only available when $F_m = 1$. Thus, constraint (14b) is guaranteed to meet. With the definition of actions, s_{t+1} only relies on s_t, so the formulation of states has the Markov property.
3) *Reward:* The immediate reward of each action is proportional to the change in achievable sum rate at RB_k

$$r_t = \delta\left(R^{(k)}_{(t)} - R^{(k)}_{(t-1)}\right) \tag{22}$$

where $R^{(k)}_{(t)}$ is the achievable sum rate of the system at RB_k at time step t. And δ is the scaling factor, limiting rewards to the range acceptable to the DRAA.

The time-step formulation enables the DRAA to perceive interference, but wireless networks cannot provide performance feedback and interference information in time and with the desired degree of granularity. Thus, the DRAA is specifically designed for the IRAF, as only the IRAF could fulfill its requirements.

2) WSRMax Problem: For the WSRMax problem, as in (16) and (17), $\bar{R}^{-1}_{n,t}$, is updated every TTI. However, within a TTI, the weight remains unchanged. So, it is feasible to formulate the MDP of the WSRMax problem in a way similar to that of the SRMax problem.

1) *States:* The states also consist of two parts. The first part is the subchannel occupation status $\mathbf{O}$, which is the same as that in the SRMax problem. The second part

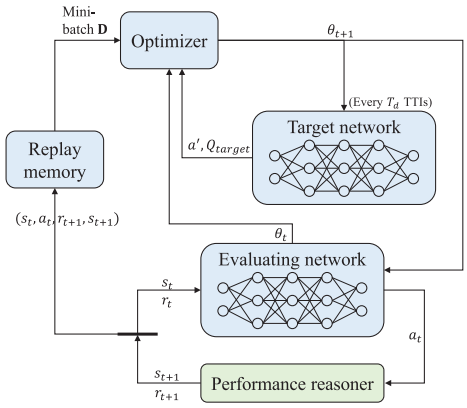

Fig. 5. Structure of the DRAA.

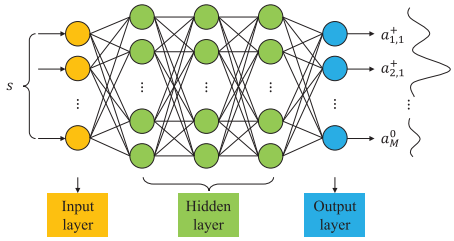

Fig. 6. Structure of deep Q-network.

is the weight vector $\mathbf{W}' = [w'_1, \ldots, w'_n, \ldots, w'_N]$, with decision status $\mathbf{F}$ embedded using $w'_n = w_n F_n$.

2) *Actions:* The actions are the same as those of the SRMax problem.

3) *Reward:* The immediate reward of each action is the change in achievable weighted sum rate at the current RB (supposing it is RB_i)

$$r_t = \sum_{j=1}^{N} \alpha^{(i)}_{m_j,j} \bar{R}^{-1}_{n,T} \left(R^{(i)}_{j,(t)} - R^{(i)}_{j,(t-1)} \right) \tag{23}$$

where $R^{(i)}_{j,(t)}$ is the achievable rate of user U_j at RB_i at time step t. And after the TTI is over, weights are updated with respect to every user.

D. DDQL-Based Resource Allocation Algorithm

The structure of the DRAA is shown in Fig. 5. The algorithm interacts with the performance reasoner to obtain interference and performance information, allocate radio resources, and improve its policy.

The policy and the target network have identical structure, which is a 5-layer fully connected neural network with one input layer, one output layer, and three hidden layers, as illustrated in Fig. 6. For the SRMax problem, the input layer has $(M + N)$ nodes, while for the WSRMax problem, it has $2M$ nodes. The layout of hidden layers is the same for both problems. All three hidden layers have $(M + N)$ nodes. And the output layer has $(M + N)$ nodes, which is the same as the number of actions. In both networks, each node uses rectified linear unit (ReLU) as activation function

$$\phi(x) = \max(0, x). \tag{24}$$

Algorithm 3 DRAA

Input: Weights for every user at current TTI $\boldsymbol{w}$
Output: Resource allocation scheme set for current TTI $\mathcal{S}$

for each $RB_k \in \mathbf{R}$ **do**
  Initialize $s_t = s_0$ and $\mathbf{A_v} = \mathbf{A}$
  for each $t \in \{0, 1, \ldots, M-1\}$ **do**
    Choose a_t with ε-greedy strategy (25)
    Taking a_t
    Observe the environment with s_{t+1} and r_{t+1}
    if $t == M - 1$ **then**
      $s_{t+1} =$ **NULL**
    end if
    Push $(s_t, a_t, s_{t+1}, r_{t+1})$ into the replay memory
    if $t \mod T_e == 0$ **then**
      Train the policy network with Algorithm 4
    end if
    Remove all the unavailable actions from $\mathbf{A_v}$
  end for
  Push $\mathbf{O}$ from s_M into $\mathcal{S}$ as S_k
end for
return $\mathcal{S} = \{S_1, \ldots, S_K\}$

The parameters of the policy network are randomly initialized, and the target network uses the same initial parameters as the policy network. The replay memory is initially empty.

The initial state s_0 for the SRMax problem is $\mathbf{0}$ for $\mathbf{O}$ part and $\mathbf{1}$ for $\mathbf{F}$ part, where $\mathbf{0}$ and $\mathbf{1}$ are vectors with all elements being zero and one, respectively. While for the WSRMax problem, in s_0, $\mathbf{O}$ is also $\mathbf{0}$, and $\mathbf{W}'$ is $\mathbf{1}$. Besides, the initial valid action set $\mathbf{A_v}$ is the same as $\mathbf{A}$, as all the actions can be chosen in the beginning.

The complete procedure of the algorithm is described as follows and summarized in Algorithm 3.

STEP 1 (Choosing Action): The policy network uses the current state s_t as input, and chooses an action a_t with ε-greedy strategy

$$a_t = \begin{cases} \text{random}(\mathbf{A_v}), & u \le \varepsilon \\ \arg\max_{a \in \mathbf{A_v}} Q(s_t, a; \theta), & \text{otherwise} \end{cases} \tag{25}$$

where random($\mathbf{A_v}$) is choosing a random element from set $\mathbf{A_v}$, and $u \sim U[0, 1]$, $\varepsilon \in (0, 1)$ is the probability of choosing a random action.

STEP 2 (Taking Action): The chosen action is taken and the state transitions into s_{t+1}, with reward r_{t+1} returned. And a_t, along with the rest of actions that could have been taken by the FAP and actions that allocate resources to users who have reached the maximum transmission power, is removed from the valid action set $\mathbf{A_v}$. Thus, the last constraint (14e) is satisfied.

STEP 3 (Updating Replay Memory): The experience tuple $(s_t, a_t, s_{t+1}, r_{t+1})$ is pushed into the replay memory. If the replay memory is full and a new tuple is coming, the oldest tuple will be replaced.

Algorithm 4 Training Algorithm for the DRAA

Input: Current TTI number T, replay memory $\mathbf{D}'$, policy network θ, target network θ^-, action randomization threshold ε

Output: Updated θ, θ^- and ε

if $\mathbf{D}'$ has enough tuples to form a mini batch **then**
  Take a random mini-batch $\mathbf{D}$ of experience tuples from $\mathbf{D}'$
else
  go to *extraTasks*
end if
for each (s, a, r, s') in $\mathbf{D}$ **do**
  if $s' == \mathbf{NULL}$ **then**
    $Q_{target} = r$
  else
    $Q_{target} = r + \gamma Q(s', \arg\max_{a'} Q(s', a'; \theta); \theta^-)$
  end if
end for
Calculate loss L with (20)
Perform a gradient descent step on L with respect to θ with gradient clipping (26)
extraTasks:
Update ε with (27)
if $T \mod T_d == 0$ **then**
  $\theta^- = \theta$
end if
return θ^-, θ and ε

STEP 4 (Updating Policy): Every T_e steps, run Algorithm 4 to update the policy network and other parameters.

To reduce computational resource consumption and enhance stability, the training algorithm does not execute at every time step, but at every T_e time steps. Additionally, the target network updating interval T_d is far longer than T_e, and gradient clipping is also utilized

$$g_{\text{clipped}} = \max(\min(g, 1), -1) \tag{26}$$

where g is the original gradient.

The ε-greedy strategy in Algorithm 3 is the key of exploring and exploiting. The longer the algorithm runs, the more information it gets, thus exploiting known information becomes more preferred than exploring the environment for better performance, leading to the descent of ε. In this article, ε linearly descends from $\varepsilon_{\max}$ to $\varepsilon_{\min}$ during training with

$$\varepsilon = \max\left(\varepsilon_{\max} - \frac{T}{T_t}(\varepsilon_{\max} - \varepsilon_{\min}), \varepsilon_{\min}\right) \tag{27}$$

where $\varepsilon_{\max}$ and $\varepsilon_{\min}$ are the maximum and minimum value of ε, T_t is the descending duration, and T is the current TTI number.

The training algorithm trains the policy network and is also in charge of synchronization from the policy network to the target network. The training algorithm is summarized in Algorithm 4.

TABLE I
SIMULATION PARAMETERS

Parameter	Value
Side length of cells	10 m
Width of the corridor	20 m
Number of room strips	4
Number of rooms per strip	4
Number of FAPs per cell	1
Number of users per cell N_p	3
Minimum distance between FAPs	8 m
System bandwidth B	5 MHz
Number of RBs K	25
RB bandwidth W	180 kHz
Semi-persistent criterion P_0	-67 dB
Max transmission power of UE $P_{\max}$	23 dBm
Path loss compensation factor α	0.7
Path loss model	$(38.46 + 20 \lg d)$ dB
White noise power density	-174 dBm/Hz
Upper bound of the SINR $\text{SINR}_{\max}$	34.6 dB
Upper bound of ε, $\varepsilon_{\max}$	1.00
Lower bound of ε, $\varepsilon_{\min}$	0.00
Network synchronization interval T_d	20 TTI
Policy network training interval T_e	3 time-steps
Training duration T_t	5000 TTI

VI. Simulation Results and Analysis

In this section, the performance of the IRAF, the NLRA, and the DRAA is evaluated through simulations. Besides, the computational complexity of the NLRA and the DRAA is also discussed.

A. Simulation Configurations

A one-floor dual-strip UDN is deployed, with a 20-m wide corridor and two four-room strips laying on both sides of the corridor. Each room is a square with sides of length 10 m, and it has randomly placed $N_p = 3$ users and an FAP. Thus, there are $M = 16$ cells and FAPs and $N = 48$ users in total. Users are subscribed to the FAP in the same room, and the minimum distance between FAPs is 8 m. Each FAP has full access to the 5-MHz spectrum consisting of 25 RBs.

Considering uplink transmission power control (TPC), in this article, the transmitting power of user U_i is

$$P_i(\text{dB}) = P_0 + \alpha \text{PL}_{i,m_i}(\text{dB}) \tag{28}$$

where $P_0 = -67$ dB is semi-persistent factor, $\alpha = 0.7$ is path loss compensation factor, and PL_{i,m_i} is the path loss between U_i and C_{m_i} in decibel. Besides, each user can and can only connect to one cell. For arbitrary user U_i, at any RB, the signal is transmitted using the same power P_i, and the total maximum transmission power $P_{\max} = 23$ dBm(200 mW) at all RBs is the same for all UEs. The path loss between UE U_i and FAP C_j is

$$\text{PL}_{i,j}(\text{dB}) = 38.46 + 20 \lg d_{i,j} \tag{29}$$

with $d_{i,j}$ being the distance between UE U_i and FAP C_j. And the upper bound of the SINR, $\text{SINR}_{\max}$, is 34.6 dB. The comprehensive simulation parameters are shown in Table I.

Besides, in this article, the data sets used in evaluating NLRA are collected from a system-level dynamic simulation platform, 5G-air-simulator [37]. And, one of the layouts used in the simulation is illustrated in Fig. 7.

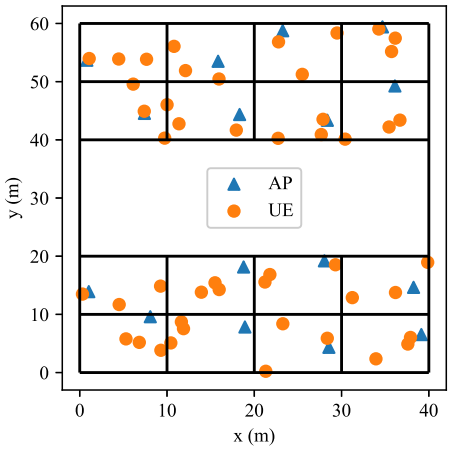

Fig. 7. One of the layouts used in simulation.

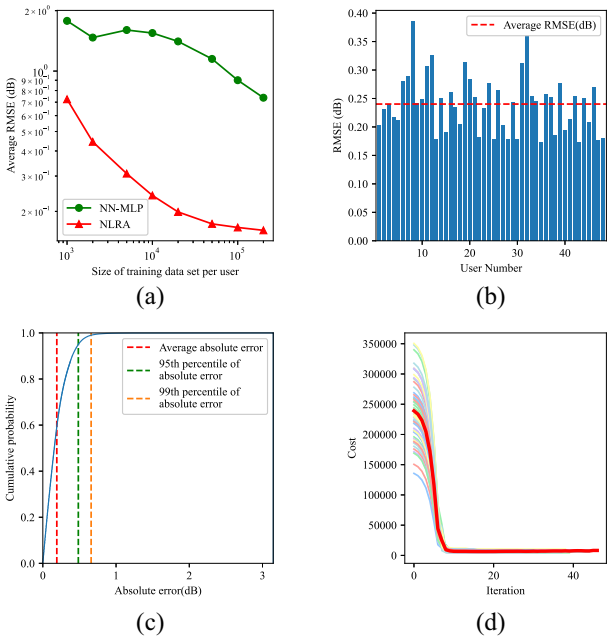

Fig. 8. Predictive and convergent performance of the NLRA. (a) Average predictive RMSE of the SINR against various training data set sizes. (b) Predictive RMSE of the NLRA for each user. (c) CDF of absolute predictive deviation of the NLRA. (d) Convergent curve of the NLRA. The per-user training data set size used in (b)–(d) is 10 000.

B. Performance of the NLRA

In this section, the performance of the NLRA proposed in Section IV is evaluated. The only baseline algorithm available is the NN-MLP [11].

The performance of both the NLRA and the NN-MLP is shown in Fig. 8. The deviation in Fig. 8(a) and (b) is presented in root-mean-square error (RMSE)

$$\mathrm{RMSE}_m = \sqrt{\frac{\sum_{i=1}^{N_m}\left(\mathrm{SINR}_{m,(i)}^{\mathrm{ideal}} - \mathrm{SINR}_{m,(i)}^{\mathrm{pred}}\right)^2}{N_m}} \tag{30}$$

where $\mathrm{SINR}_{m,(i)}^{\mathrm{ideal}}$ and $\mathrm{SINR}_{m,(i)}^{\mathrm{pred}}$ are the ideal (i.e., fast fading is excluded) and the predicted SINR, expressed in decibel, of entry i in testing data set for user U_m, and N_m is the size of testing data set for user U_m. In this article, $N_m = 20\,000\ \forall m$. In Fig. 8(a), the predictive performance of the SINR for both algorithms against various data set sizes are shown. As the data set size grows, the performance of both algorithms becomes better. However, the performance of the NLRA is far better than that of the NN-MLP. To reach the same level of mean RMSE (around 0.7 dB), the NLRA only needs 1000 entries per user, while the NN-MLP requires around 200 000. And when the NLRA uses 200 000 entries of data per user to train, the mean RMSE could reach 0.161 dB, which is 4.6 times less than that of the NN-MLP.

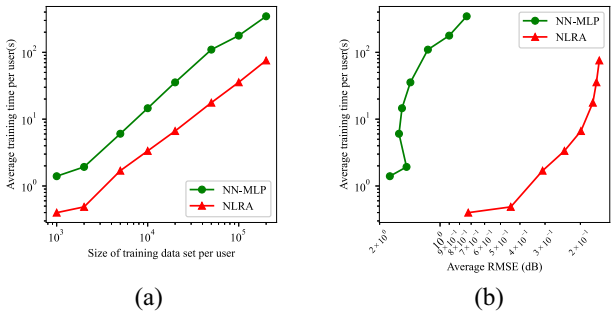

Fig. 9. Training time consumption of the NLRA. (a) Training time consumption against various training data set sizes. (b) Predictive RMSE against average training time.

Fig. 8(b) shows the per-user SINR prediction RMSE at data set size of 10 000. Among all users, the largest RMSE is less than 0.4 dB, and only 12.5% of users (six users) suffer from RMSE over 0.3 dB, showing high consistency of performance.

The mean absolute error (MAE) is defined as the average of all absolute values of the difference between ideal and predicted SINRs, both in decibel

$$\mathrm{MAE}_m = \frac{\sum_{i=1}^{N_m}\left|\mathrm{SINR}_{m,(i)}^{\mathrm{ideal}} - \mathrm{SINR}_{m,(i)}^{\mathrm{pred}}\right|}{N_m}. \tag{31}$$

Fig. 8(c) illustrates the cumulative distribution function (CDF) of all users' absolute predictive deviance for all testing entries (960 000 in total) at the training data set size of 10 000. The average MAE is 0.191 dB, which is greater than about 60% of entries. The 95th and the 99th percentiles of absolute predictive deviance are 0.485 and 0.66 dB, respectively, showing extremely high predictive precision.

Further, the NLRA converges quickly. As depicted in Fig. 8(d), the cost reduces rapidly within the first few iterations, and all users could converge within 50 iterations. Note that, in the figure, the red bold line stands for the average cost among users, while the other lines show the convergent curve of each user. Faster convergence leads to less time consumption.

As illustrated in Fig. 9(a), at the same training data set size, the training time consumption of the NLRA is shorter than that of the NN-MLP by one order of magnitude. Combining the results of time consumption and predictive precision, the NLRA consumes two orders of magnitude shorter training time to reach the same predictive performance as that of the NN-MLP, as shown in Fig. 9(b).

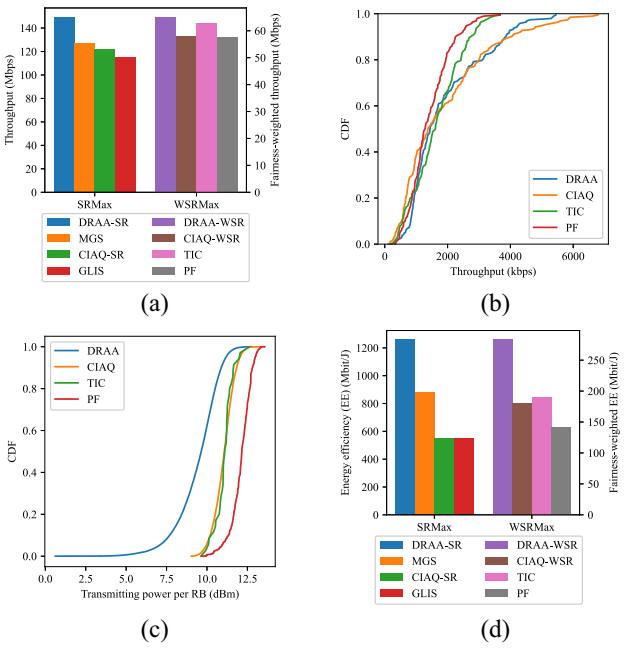

Fig. 10. Performance of algorithms in Scenario 1. (a) (Weighted) throughput in solving the SRMax and the WSRMax problem. (b) CDF of per user throughput in solving the WSRMax problem. (c) CDF of per RB aggregated transmission power in solving the WSRMax problem. (d) (Weighted) EE in solving the SRMax and the WSRMax problem.

C. Performance of the DRAA

In this section, the performance of the DRAA proposed in Section V is evaluated, and is compared with five baseline algorithms (i.e., the MGS, the TIC, the PF, the CIAQ, and the GLIS) introduced in Section I. Among the baseline algorithms, the MGS and the GLIS are specifically targeted at the SRMax problem, while the TIC and the PF are focused on the WSRMax problem. And, the CIAQ could be applied to both.

In this article, two different scenarios are considered to evaluate the performance of these algorithms:

Scenario 1: Algorithms could obtain any information they need directly, perfectly, and instantaneously from wireless network.

Scenario 2: Algorithms run in the IRAF, with full interference information and other inaccessible data obtained from the IRAF.

Only scenario 1 is discussed in this section to show the theoretical performance of all the algorithms. While performance in scenario 2 is evaluated in the next section to show the effectiveness of the IRAF.

Fig. 10(a) shows the theoretical performance of all the algorithms. In solving the SRMax problem, the DRAA achieves 14.86%, 19.93%, and 27.31% higher throughput than the MGS, the CIAQ, and the GLIS, respectively.

As for the WSRMax problem, the DRAA gains 12.33%, 3.77%, and 12.90% in fairness-weighted throughput, compared with the CIAQ, the TIC, and the PF.

Apart from the excellent overall performance, in solving the WSRMax problem, the average throughput of users with less than the 20th percentile of per-user throughput is greatly improved. As shown in Fig. 10(b), the average throughput is 737.07 Kb/s for the DRAA, which is 55.48%, 35.15%, and 16.71% higher than the CIAQ, the TIC, and the PF.

Fig. 10(c) depicts the CDF of transmission power at each RB in solving the WSRMax problem. With appropriate interference coordination of the DRAA, transmission power of the system is drastically reduced. Thus, the energy efficiency (EE) of the system is greatly enhanced, as illustrated in Fig. 10(d). In solving the SRMax problem, using the DRAA, the EE of the system increases by more than 40% compared to the rest algorithms, while an over 49% gain in fairness-weighted EE is obtained when solving the WSRMax problem with the DRAA.

D. Effectiveness of the IRAF

Using the IRAF, some algorithms (the MGS and the TIC, for instance) with impractical need to iteratively obtain performance information to allocate resources could operate normally. While for learning-based algorithms (the CIAQ for instance), the IRAF could become a good trainer. As the DRAA is specifically designed for the IRAF, it could benefit from both.

In Fig. 11(a), the darker parts on the top of the bars indicate increments in performance in scenario 2 over scenario 1, and the lighter parts mean otherwise. In solving the SRMax problem, the gap between the two scenarios is at most 2.51%. While in solving the WSRMax problem, the maximum gap becomes slightly wider, as the problem becomes more complex. For the TIC, the gap is 5.81%, but for the rest of algorithms, there is almost no difference ($\leq 0.5\%$). Fig. 11(b) and (c) also bear a clear resemblance to their counterparts in scenario 1, which further stresses the effectiveness of the IRAF.

The difference in EE is depicted in Fig. 11(d), and the darker and lighter parts have the same meaning as that in Fig. 11(a). Though some of the results show greater differences, which is expected as transmission power is also considered in EE, the overall difference remains low. Only the TIC reports a gap of over 10% between scenarios. These results show the great similarity between the environment built by the IRAF and the actual wireless network.

Further, the convergent curve of the DRAA is shown in Fig. 11(e). The red bold line shows the average normalized performance of the DRAA among different layouts, while the rest of the lines are the convergent curve in each layout. As shown in Fig. 11(e), the DRAA could converge within 5000 iterations, and the results of the last 2000 iterations are generated in online usage, which shows that the offline trained algorithm could be put online seamlessly with the help of the IRAF.

E. Complexity Analysis

1) Complexity of Algorithm 1: In this article, the NLRA uses trust-region reflective (TRF) algorithm [33] to solve. For each user, at each iteration, the TRF algorithm needs to evaluate the Jacobian and Hessian matrix of the data set w.r.t. $(N - N_p + 1)$ unknown parameters, which has the complexity of $O((N - N_p + 1)^2) \sim O(N^2)$. Thus, denoting the average

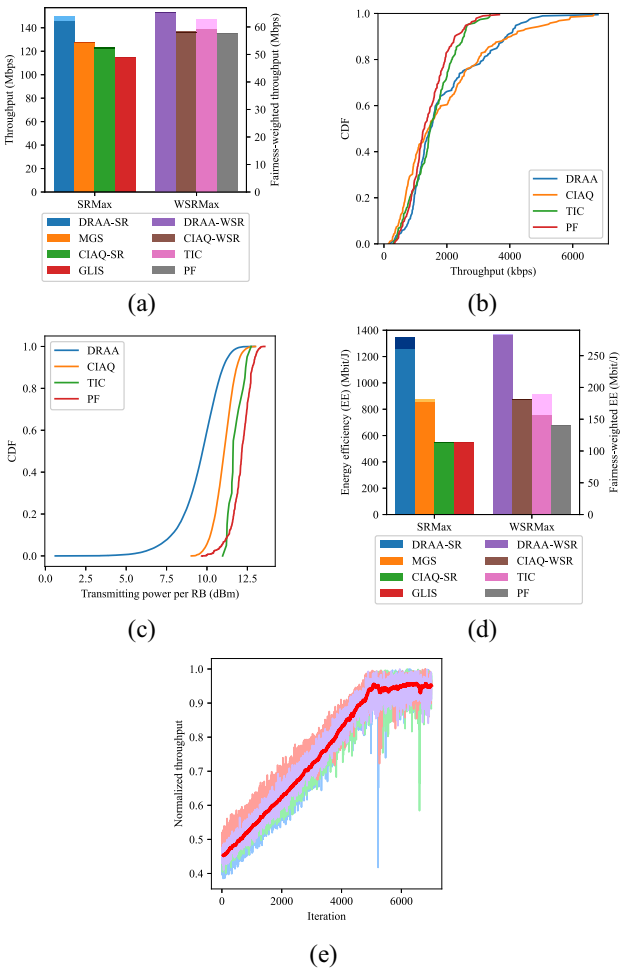

Fig. 11. Performance of algorithms in scenario 2. (a) (Weighted) throughput in solving the SRMax and the WSRMax problem. (b) CDF of per user throughput in solving the WSRMax problem. (c) CDF of per RB aggregated transmission power in solving the WSRMax problem. (d) (Weighted) EE in solving the SRMax and the WSRMax problem. (e) Convergent curve of the DRAA.

iteration number until the algorithm converged for all users as E, and the data set size for each user is $|\boldsymbol{\mathcal{D}}|$, the complexity of the algorithm is $O(|\boldsymbol{\mathcal{D}}|EN^3)$.

2) Complexity of Algorithm 2: Only a single run of (13) is needed to get the prediction result. Thus, the complexity is $O(N)$.

3) Complexity of Algorithm 3: The most computationally expensive part of the algorithm is evaluating the 5-layer neural network with the number of nodes of each layer proportional to N. Thus, the computational complexity is $O(N^2)$.

4) Complexity of Algorithm 4: What is computationally intensive in Algorithm 4 is target acquiring and weight updating. As two network evaluations are required to generate a target for a tuple in $\boldsymbol{D}$, the computational complexity of target acquiring is $O(|\boldsymbol{D}|N^2)$. Furthermore, as backpropagation process in weight updating has the same computational complexity as the feedforward evaluation of the network, the computational complexity of weight updating is $O(N^2)$. Thus, the overall computational complexity of Algorithm 4 is $O(|\boldsymbol{D}|N^2)$.

VII. Conclusion

In this article, a novel IRAF is proposed, with an interference modeling algorithm (NLRA) as a source of the IRAF's interference identification ability, and a resource allocation algorithm (DRAA) to fully leverage the IRAF and show its effectiveness proposed. Using the ability of the performance reasoner module to accurately identify interference, the IRAF could provide resource allocators and interference coordination approaches with interference information unavailable in regular wireless networks. Further, the IRAF could isolate network fluctuations, and train resource allocator offline with the capability of seamless online usage, which are also based on the ability to accurately and stably identify interference. To support the ability of the IRAF, the NLRA is proposed, which could precisely model interference between users using only seconds of historical data collected from operating wireless networks and taking less than one second to train. The unprecedentedly low data requirement and fast training speed make the NLRA the first algorithm that meets the requirements of the IRAF. To show the merits of the IRAF thoroughly, the DRAA, with high performance and fast training, is proposed. With precise and stable interference and performance information provided by the IRAF, the DRAA could effectively coordinate interference, and thus could reach higher throughput and improve EE by more than 40%. Besides, as a learning-based algorithm, the DRAA could use the IRAF to enable near-perfect offline training. Furthermore, in comparison, the merit of high compatibility of the IRAF is shown. Compared with the performance of algorithms in the ideal network, almost no performance loss is observed when the algorithms operate in the IRAF, which also better demonstrates the versatility of the IRAF.

References

[1] Q. Kuang, W. Utschick, and A. Dotzler, "Optimal joint user association and multi-pattern resource allocation in heterogeneous networks," *IEEE Trans. Signal Process.*, vol. 64, no. 13, pp. 3388–3401, Jul. 2016.

[2] S. Y. Kim, J. A. Kwon, and J. W. Lee, "Sum-rate maximization for multicell OFDMA systems," *IEEE Trans. Veh. Technol.*, vol. 64, no. 9, pp. 4158–4169, Sep. 2015.

[3] J. Yu and C. Yin, "Block-level resource allocation with limited feedback in multicell cellular networks," *J. Commun. Netw.*, vol. 18, no. 3, pp. 420–428, 2016.

[4] S. V. Hanly, L. L. Andrew, and T. Thanabalasingham, "Dynamic allocation of subcarriers and transmit powers in an OFDMA cellular network," *IEEE Trans. Inf. Theory*, vol. 55, no. 12, pp. 5445–5462, Dec. 2009.

[5] E. Pateromichelakis, M. Shariat, A. U. Quddus, and R. Tafazolli, "Graph-based Multicell scheduling in OFDMA-based small cell networks," *IEEE Access*, vol. 2, pp. 897–908, 2014.

[6] N. Forouzan and S. A. Ghorashi, "Inter-cell interference coordination in downlink orthogonal frequency division multiple access systems using Hungarian method," *IET Commun.*, vol. 7, no. 1, pp. 23–31, 2013.

[7] Y. Yu, E. Dutkiewicz, X. Huang, and M. Mueck, "Downlink resource allocation for next generation wireless networks with inter-cell interference," *IEEE Trans. Wireless Commun.*, vol. 12, no. 4, pp. 1783–1793, Apr. 2013.

[8] G. Li and H. Liu, "Downlink radio resource allocation for multi-cell OFDMA system," *IEEE Trans. Wireless Commun.*, vol. 5, no. 12, pp. 3451–3459, Dec. 2006.

[9] F. Wang, W. Chen, H. Tang, and Q. Wu, "Joint optimization of user association, subchannel allocation, and power allocation in multi-cell multi-association OFDMA heterogeneous networks," *IEEE Trans. Commun.*, vol. 65, no. 6, pp. 2672–2684, Jul. 2017.

[10] K. Yang, N. Prasad, and X. Wang, "An auction approach to resource allocation in uplink OFDMA systems," *IEEE Trans. Signal Process.*, vol. 57, no. 11, pp. 4482–4496, Nov. 2009.
[11] J. Cao *et al.*, "Resource allocation for ultradense networks with machine-learning-based interference graph construction," *IEEE Internet Things J.*, vol. 7, no. 3, pp. 2137–2151, Mar. 2020.
[12] G. Cao, Z. Lu, X. Wen, T. Lei, and Z. Hu, "AIF: An artificial intelligence framework for smart wireless network management," *IEEE Commun. Lett.*, vol. 22, no. 2, pp. 400–403, Feb. 2018.
[13] H. W. Kuhn, "The Hungarian method for the assignment problem," *Naval Res. Logist. Quart.*, vol. 2, nos. 1–2, pp. 83–97, Mar. 1955.
[14] Q. Mao, F. Hu, and Q. Hao, "Deep learning for intelligent wireless networks: A comprehensive survey," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 4, pp. 2595–2621, 4th Quart., 2018.
[15] H. Sun, X. Chen, Q. Shi, M. Hong, X. Fu, and N. D. Sidiropoulos, "Learning to optimize: Training deep neural networks for wireless resource management," in *Proc. IEEE Workshop Signal Process. Adv. Wireless Commun.*, Jul. 2017, pp. 1–6.
[16] W. Cui, K. Shen, and W. Yu, "Spatial deep learning for wireless scheduling," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 6, pp. 1248–1261, Jun. 2019.
[17] Z. Xu, Y. Wang, J. Tang, J. Wang, and M. C. Gursoy, "A deep reinforcement learning based framework for power-efficient resource allocation in cloud RANs," in *Proc. IEEE Int. Conf. Commun.*, Jul. 2017, pp. 1–6.
[18] Y. S. Nasir and D. Guo, "Multi-agent deep reinforcement learning for dynamic power allocation in wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 10, pp. 2239–2250, Oct. 2019.
[19] H. S. Lee, J. Y. Kim, and J. W. Lee, "Resource allocation in wireless networks with deep reinforcement learning: A circumstance-independent approach," *IEEE Syst. J.*, vol. 14, no. 2, pp. 2589–2592, Jun. 2020.
[20] 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 Trans. Wireless Commun.*, vol. 18, no. 11, pp. 5141–5152, Nov. 2019.
[21] B. Gu, X. Zhang, Z. Lin, and M. Alazab, "Deep multi-agent reinforcement learning-based resource allocation for Internet of Controllable Things," *IEEE Internet Things J.*, vol. 8, no. 5, pp. 3066–3074, Mar. 2021.
[22] C. Zhao, X. Xu, Z. Gao, and L. Huang, "A coloring-based cluster resource allocation for ultra dense network," in *Proc. IEEE Int. Conf. Signal Process. Commun. Comput.*, Aug. 2016, pp. 1–5.
[23] Y. Xu, G. Li, J. Tang, and L. Luan, "Robust resource allocation for uplink sum rate maximization in multi-cell heterogeneous networks," in *Proc. IEEE Int. Conf. Commun. Workshop*, 2018, pp. 1–6.
[24] T. Peng *et al.*, "A data-driven and load-aware interference management approach for ultra-dense networks," *IEEE Access*, vol. 7, pp. 129514–129528, 2019.
[25] J. Cao, T. Peng, Z. Qi, R. Duan, Y. Yuan, and W. Wang, "Interference management in ultradense networks: A user-centric coalition formation game approach," *IEEE Trans. Veh. Technol.*, vol. 67, no. 6, pp. 5188–5202, Jun. 2018.
[26] M. Ahmed, M. Peng, M. Abana, S. Yan, and C. Wang, "Interference coordination in heterogeneous small-cell networks: A coalition formation game approach," *IEEE Syst. J.*, vol. 12, no. 1, pp. 604–615, Mar. 2018.
[27] R. Kwan, C. Leung, and J. Zhang, "Proportional fair multiuser scheduling in LTE," *IEEE Signal Process. Lett.*, vol. 16, no. 6, pp. 461–464, Jun. 2009.
[28] X. Liao, J. Shi, Z. Li, L. Zhang, and B. Xia, "A model-driven deep reinforcement learning heuristic algorithm for resource allocation in ultra-dense cellular networks," *IEEE Trans. Veh. Technol.*, vol. 69, no. 1, pp. 983–997, Jun. 2020.
[29] A. Abdelnasser and E. Hossain, "Resource allocation for an OFDMA cloud-RAN of small cells underlaying a Macrocell," *IEEE Trans. Mobile Comput.*, vol. 15, no. 11, pp. 2837–2850, Nov. 2016.
[30] B. Ma, M. H. Cheung, V. W. S. Wong, and J. Huang, "Hybrid overlay/underlay cognitive femtocell networks: A game theoretic approach," *IEEE Trans. Wireless Commun.*, vol. 14, no. 6, pp. 3259–3270, Jun. 2015.
[31] M. G. Kendall and A. Stuart, *The Advanced Theory of Statistics: Inference and Relationship*, vol. 2, 4th ed. London, U.K.: Griffin, 1979.
[32] Y. Guo, C. Hu, T. Peng, H. Wang, and X. Guo, "Regression-based uplink interference identification and SINR prediction for 5G ultra-dense network," in *Proc. IEEE Int. Conf. Commun.*, 2020, pp. 1–6.
[33] M. A. Branch, T. F. Coleman, and Y. Li, "A subspace, interior, and conjugate gradient method for large-scale bound-constrained minimization problems," *SIAM J. Sci. Comput.*, vol. 21, no. 1, pp. 1–23, Jan. 1999.
[34] R. S. Sutton and A. G. Barto, *Reinforcement Learning, Second Edition: An Introduction*, 2nd ed. Cambridge, MA, USA: MIT Press, 2018.
[35] V. Mnih *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.
[36] H. van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double Q-learning," in *Proc. 30th AAAI Conf. Artif. Intell.*, Sep. 2015, pp. 2094–2100.
[37] S. Martiradonna, A. Grassi, G. Piro, and G. Boggia, "5G-air-simulator: An open-source tool modeling the 5G air interface," *Comput. Netw.*, vol. 173, May 2020, Art. no. 107151.

Tao Peng (Member, IEEE) received the bachelor's, master's, and Ph.D. degrees from Beijing University of Posts and Telecommunications (BUPT), Beijing, China, in 1999, 2002, and 2010, respectively.

He is currently an Associate Professor with BUPT. He was the Chair of the Device-to-Device Technical Discussion Group, IMT-A/IMT-2020 Propulsion Group, CCSA. He has authored more than 120 academic papers with 30 SCI indexed, and an inventor of more than 50 international and domestic patents. His current research interests include B5G/6G communication and network, cognitive radio, satellite network, and UAV communication and application.

Yichen Guo received the B.S. degree in communication engineering from Beijing University of Posts and Telecommunications (BUPT), Beijing, China, in 2018, where he is currently pursuing the Ph.D. degree with the Key Laboratory of Universal Wireless Communications (Ministry of Education).

His current research interests include interference and radio resource management in ultra-dense industrial networks.

Yachen Wang received the M.S. degree in computer science and technology from the University of York, York, U.K., in 2003.

His current research interests include cloud networks, wireless computing, and edge computing.

Gonglong Chen received the Ph.D. degree in computer science and technology from Zhejiang University, Hangzhou, China, in 2020.

His current research interests include cloud networks, mobile computing, and cellular network.

Feng Yang received the Ph.D. degree in electrical engineering from Tsinghua University, Beijing, China, in 2009.

His current research interests include cloud networks, edge computing, and wireless computing.

Wei Chen received the Ph.D. degree in computer science and technology from Beijing University of Posts and Telecommunications, Beijing, China, in 2010.

His current research interests include cloud networks, mobile computing, and wireless computing.