IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 28 June 2023, accepted 10 August 2023, date of publication 18 August 2023, date of current version 24 August 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3306518

RESEARCH ARTICLE

Joint Beamforming, Power Control, and Interference Coordination: A Reinforcement Learning Approach Replacing Rewards With Examples

JENG-SHIN SHEU, CHENG-KUEI HUANG, AND CHUN-LUNG TSAI
Department of Computer Science and Information Engineering, National Yunlin University of Science and Technology, Yunlin 640002, Taiwan
Corresponding author: Jeng-Shin Sheu (jssheu@yuntech.edu.tw)

ABSTRACT In this paper, we consider the problem of multi-cell interference coordination by joint beamforming and power control. Recent efforts have explored the use of reinforcement learning (RL) methods to tackle this complex optimization problem. Typically, a decentralized multi-agent framework is adopted, wherein each base station operates as an independent RL agent. This distributed coordination has gained attention because designing a reward function that effectively captures the condition of the entire cellular network is challenging for single-agent RL models. However, the distributed approach introduces unique challenges, particularly the non-stationary of the multi-agent environment, as agents continually adapt their policies to interact with one another. The non-stationary environment necessitates information exchange among agents, as local observations of each agent are insufficient to fully capture the true state of the environment. Unfortunately, this information exchange incurs a significant overhead, thereby limiting data transmission capabilities. To address these challenges, we propose a novel single-agent RL approach that eliminates the need for information exchange and the conventional reward function. Instead, we leverage success examples to guide the learning process. Simulation results show that the proposed approach outperforms the existing multi-agent method and theoretical algorithm in terms of sum rates. Additionally, our approach ensures a uniform quality of service while maximizing the overall sum rate.

INDEX TERMS Beamforming, example-based control, interference coordination, power control, reinforcement leaning.

I. INTRODUCTION

In mobile communication systems, base stations (BSs) utilize the frequency reuse concept to share the available spectrum. That is, the co-channel cells are separated by a sufficient distance to mitigate inter-cell interference (ICI), thereby maintaining the communication quality. However, it results in a reduced system capacity [1]. In fact, because of the ever-growing demand for mobile traffic, the universal frequency reuse (UFR) has long been adopted to achieve aggressive spectrum reuse and simplify frequency planning. Under UFR, all BSs simultaneously share the entire available spectrum, leading to significant levels of ICI. The signal-to-noise-plus-interference ratio (SINR) plays a crucial role in determining the spectral efficiency of a link. Unfortunately, the adoption of UFR leads to substantial ICI, particularly impacting users at the cell borders and degrading their signal qualities.

To ensure a uniform achievable rate for cellular networks, network multiple-input multiple-output (MIMO) transmission has been employed to improve the signal quality for cell-edge users [2], [3]. In Long-Term Evolution-Advanced (LTE-A), this technique is known as coordinated multipoint (CoMP) [4]. However, implementing network MIMO

The associate editor coordinating the review of this manuscript and approving it for publication was Tariq Masood.

entails high processing and implementation complexity [5]. Inter-cell interference coordination (ICIC) has emerged as a more practical approach for mitigating ICI [6], [7]. ICIC involves imposing certain constraints on the radio resource management to enhance channel conditions for severely interfered user equipments (UEs), thereby achieving high spectral efficiency. Two commonly adopted ICIC strategies are soft frequency reuse (SFR) and fractional frequency reuse (FFR) [8], [9], [10]. Specifically, SFR has been widely utilized in LTE-A to minimize ICI at cell borders and improve overall system performance.

Another promising approach for coordinating interference among co-channel cells is the joint optimization of transmit beamforming and power control [11], [12]. In practical cellular setups, BSs are equipped with multiple antennas, while UEs typically have single antennas. When multiple antennas are deployed at the transmitter, the transmit beam pattern can be adjusted to minimize the interference experienced by the receivers in co-channel cells. However, in scenarios where antenna arrays are employed at the receivers, each receive beamforming operation is performed independently, thereby limiting its ability to mitigate interference for all co-channel receivers simultaneously. Therefore, there is a distinction between transmit and receive beamforming in the cellular network, as the former can be jointly designed, while the latter cannot. It is important to note that when the number of antennas is greater than or equal to the number of co-channel receivers, transmit beamforming enables the strategic placement of nulls in the directions of each co-channel receiver [13], [14]. In addition to beamforming, coordinating the transmit power among BSs is also crucial. This introduces the challenge of jointly optimizing transmit beamforming and power control across co-channel cells.

In cellular commutation systems, the system capacity refers to the achievable sum rate of all UEs within the system. However, maximizing the system capacity poses a challenge as it involves a non-convex objective function, making it an NP-hard problem. To tackle this issue, several suboptimal methods have been proposed to obtain feasible solutions, which can be categorized into centralized and distributed approaches.

In a centralized framework, all BSs share their channel state information (CSI) with a central controller, responsible for calculating feasible solutions for each BS. Fractional programming (FP) [15] reformulates the original non-convex problem as a sequence of convex problems, enabling the use of an iterative optimization algorithm operating in a centralized way. On the other hand, the distributed framework allows individual BSs to compute their own transmit beamforming and power based on local observations in a distributed manner. In [16], the authors propose a distributed algorithm for maximizing the system sum rate by reformulating it as an iterative minimization of weighted mean-square error. This reformulation establishes an equivalence relation, allowing each BS to optimize its transmission strategy in a distributed way while considering the impact on the overall sum rate.

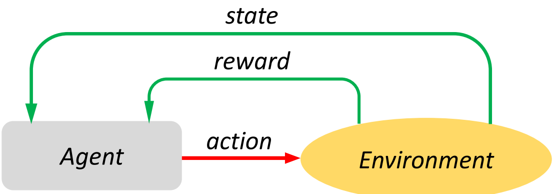

FIGURE 1. The agent-environment interface of an RL system.

The coordination between concurrent co-channel BSs presents a complex *decision-making* problem. In addition to the aforementioned theoretical approaches, reinforcement learning (RL) frameworks can be utilized to learn decision policies for sequential decision problems. RL is a machine learning approach that rewards desired behaviors and penalizes undesired ones. An RL model comprises two main entities: an agent and an environment, as illustrated in Fig. 1. The agent learns to interact with the environment by taking a sequence of actions. Throughout this interaction, the agent observes both the states and rewards provided by the environment.

Recently, an RL-based approach has been proposed for distributed downlink-beamforming coordination in cellular networks [17]. In this approach, each BS in the network functions as an RL agent, operating within a shared environment. However, RL techniques were initially developed for single-agent settings and stationary environments. Therefore, in the multi-agent setting, a major challenge arises due to the non-stationarity of the shared environment [18], [19], [20]. The nonstationary nature of the environment arises from the fact that all agents learn concurrently and independently, and the actions taken by any one agent impact the objectives of the other agents. Consequently, in a multi-agent setting, in order for agents to make informed decisions, they need to exchange specific information among themselves to understand the behavior of other concurrent agents.

In addition to the extensive information exchange required among agents, designing a suitable reward function poses a significant challenge in multi-agent learning [21], [22]. Manually engineered reward functions are critical for the success of RL models. Arguably, the need for a well-designed reward function imposes significant constraints on RL. To this, some previous works in [23] and [24] have explored alternative methods, such as imitation learning, to avoid the explicit design of reward functions. In imitation learning, RL agents learn from a set of expert demonstrations without relying on predefined rewards. This approach is inspired by the concept of generative adversarial networks [25]. By performing distribution matching, imitation learning minimizes the divergence between the target distribution and the distribution resulting from the agent's interaction with the environment.

While imitation learning methods greedily imitate demonstrated actions, it is important to note that aggregation errors can lead to deviations from the demonstrated states. Addressing this challenge, a novel learning method called *C-learning* was recently proposed [26]. C-learning reformulates classic goal-conditioned RL by training a classifier to predict and control the future state of the environment. Building upon the C-learning approach, an example-based control method [27] enables agents to not only reach specific *goals* but also solve tasks. In this problem setting, a collection of *success examples* representing success states is used to teach the agent to recognize "what the world would look like if the task were solved."

To address the challenges of information exchange and reward function design, we propose a single-agent RL-based method for joint transmit beamforming and power coordination. Our approach eliminates the need for information exchange and avoids the manual design of a complex reward function by leveraging example-based control through binary classification. We use success examples to guide the RL model towards optimal achievable sum rates. Our method offers several advantages compared to existing approaches. Firstly, it overcomes the nonstationary nature of environments resulting from changing decision-making policies. Secondly, it alleviates the burden of designing intricate reward functions, which are often challenging and resource-intensive to obtain. Thirdly, our method eliminates the need for information exchange among BSs, reducing radio resource overhead and improving data transmission and system capacity. Additionally, our approach ensures a uniform achievable rate for co-channel UEs regardless of their locations, while maximizing the overall system sum rate. Finally, by substituting examples for rewards, our RL model can tackle more general tasks, expanding its learning capabilities beyond specific goals. Traditional RL models, in contrast, focus solely on reaching desired states associated with specific goals. Simulation results demonstrate that our method outperforms both multi-agent and theoretical FP methods in terms of system sum rates.

The remaining sections of this paper are organized as follows. Section II presents the system and channel models. In Section III, we provide a comprehensive review of the multi-agent RL approach. Section IV introduces the problem formulation and presents the derivation of the example-based RL method. The proposed example-controlled single-agent RL model for joint transmit beamforming and power control coordination is presented in Section V. Simulation results are provided in Section VI. Finally, Section VII concludes the paper.

In this paper, the following notation is used. The superscripts T and $\dagger$ represent the matrix transpose and conjugate transpose, respectively. The expectation, floor and modulo operations are denoted by $\mathbb{E}(\cdot)$, $\lfloor\cdot\rfloor$ and $\mathrm{mod}(\cdot)$, respectively. The function $\mathcal{CN}(m, v)$ denotes the complex normal distribution with mean m and variance v. $\mathcal{U}(a, b)$ represents the uniform distribution over the range $[a, b]$. The symbol $\sim$ denotes "distributed as", and $\jmath \equiv \sqrt{-1}$ is the imaginary root. $\mathbb{C}^{a\times b}$ represents spaces in $a \times b$ matrices with complex entries. The notation $\|\mathbf{w}\|_2$ denotes the Euclidean norm of vector $\mathbf{w}$.

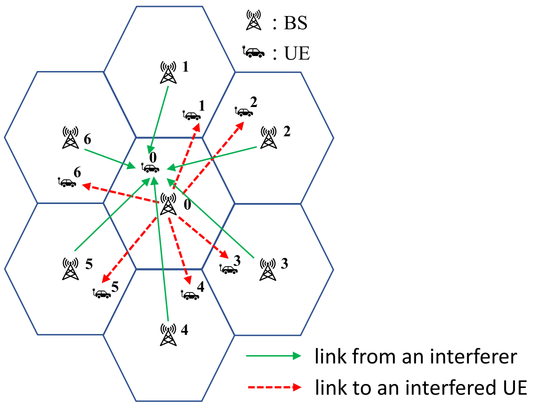

FIGURE 2. Multi-cell multi-user MISO wireless system.

II. SYSTEM AND CHANNEL MODELS

Consider a downlink cellular network consisting of B cells, where each BS is equipped with a uniform linear antenna array of N elements. Within each cell, single-antenna UEs are served simultaneously using orthogonal frequency bands to avoid interference within the cell. The commonly employed technique for achieving this is orthogonal frequency-division multiple access (OFDMA) [28]. OFDMA is a multicarrier modulation scheme that divides the input data stream into multiple sub-streams, which are transmitted in parallel over different orthogonal subchannels. The number of sub-streams is determined to ensure that the bandwidth of each subchannel is smaller than the coherence bandwidth of the channel, thus promoting relatively flat fading characteristics across the orthogonal subchannels.

Due to the absence of interference within the cell, we employ a single-user detection mechanism [29], [30]. Consequently, each orthogonal frequency band in the cellular network accommodates B co-channel UEs. For the sake of simplicity, let us assume that the bth UE establishes a *direct link* with the bth BS (i.e., its home BS). As a result, the downlink scenario can be modeled as a multi-input single-output (MISO) system, as shown in Fig. 2. For each direct link, there are two distinct kinds of neighboring entities: the *interferers* and the *interfered neighbors*. In Fig. 2, for direct link between BS 0 and UE 0, the set of UEs {UE1, UE2, ..., UE6} represents the neighbors *interfered* by BS 0, whereas the set of BSs {BS1, BS2, ..., BS6} corresponds to the *interferers* to UE 0. The received signal at UE b at time t can be expressed as follows:

$$y_b(t) = \mathbf{w}_b(t)\,\mathbf{h}^{\dagger}_{b,b}(t)\,x_b(t) + \sum\nolimits_{j\neq b} \mathbf{w}_j(t)\,\mathbf{h}^{\dagger}_{j,b}(t)\,x_j(t) + z_b(t), \tag{1}$$

where $\mathbf{w}_b(t) \in \mathbb{C}^{N\times 1}$ is the downlink beamforming vector at BS b, $\mathbf{h}_{j,b}(t) \in \mathbb{C}^{N\times 1}$ represents the downlink channel vector between BS j and UE b, $x_b(t) \in \mathbb{C}$ is the data transmitted from BS b and $z_b(t) \in \mathbb{C}$ denotes the additive white Gaussian noise (AWGN) with variance σ^2. The beamforming vector $\mathbf{w}_b(t)$ is a unit vector. Let $P_j(t) \equiv \mathbb{E}\left[\left|x_j(t)\right|^2\right]$ denote the transmit power from BS j at time slot t. In (1), the first product term represents the contribution from the direct link, while the second term, which is a summation of product terms, accounts for the total interference caused by the interferers $\{\text{BS}\, j, \forall j \neq b\}$.

Now let's define the channel model between any pair of BS j and UE k. Following a similar approach to [17], we assume a block fading channel model. In this model, the large-scale fading remains unchanged within a block of consecutive time slots, while the small-scale fading varies for each individual time slot. Let $\beta_{j,k}$ represent the large-scale fading coefficient and L be the number of multipaths. Within a block fading at time slot t, the small-scale fading coefficient and the angle of departure (AOD) for path l are denoted by $\alpha_{j,k}^{(l)}(t)$ and $\phi_{j,k}^{(l)}(t)$, respectively. Additionally, the fading coefficient $\alpha_{j,k}^{(l)}(t)$ follows a complex Gaussian distribution $\mathcal{CN}\left(0, 1/L\right)$ and the AOD $\phi_{j,k}^{(l)}(t)$ is with uniform distribution $\mathcal{U}\left(\theta_{j,k} - \Delta/2, \theta_{j,k} + \Delta/2\right)$, where Δ is the angular spread and $\theta_{j,k}$ is the nominal AOD. The Gauss-Markov (GM) model [31], a widely used model for characterizing the fading processes, is employed in this paper. Specifically, we utilize the first-order GM process to describe the small-scale fading as follows:

$$\alpha_{j,k}^{(l)}(t+1) = \rho \alpha_{j,k}^{(l)}(t) + \sqrt{1-\rho^2} u_{j,k}(t), \tag{2}$$

where $u_{j,k}(t) \sim \mathcal{CN}(0, 1)$ is the white Gaussian driving noise. The parameter $\rho \in [0, 1]$ represents the correlation coefficient, where a small value indicates fast fading, while a large value indicates slow fading. The array steering vector $\mathbf{a}(\phi) \in \mathbb{C}^{N \times 1}$ in the direction of ϕ is given by

$$\mathbf{a}(\phi) = \left[1, e^{\jmath 2\pi (d/\lambda)\cos\phi}, \ldots, e^{\jmath 2\pi (d/\lambda)(N-1)cos\phi}\right]^T \in \mathbb{C}^{N\times 1}, \tag{3}$$

where λ and d denote the wavelength and antenna spacing, respectively. We assume that the antenna spacing is half of the wavelength, i.e., $d = \lambda/2$.

Therefore, the downlink channel vector between BS j and UE k at time slot t is expressed as follows:

$$\begin{aligned}\mathbf{h}_{j,k}(t) \\ = \sqrt{\beta}_{j,k} \sum\nolimits_{l=1}^{L} \alpha_{j,k}^{(l)}(t)\, \mathbf{a}_l\left(\phi_{j,k}^{(l)}(t)\right), \forall j, k \in \{1, \ldots, B\}\end{aligned} \tag{4}$$

According to (1), the SINR observed by UE b at time slot t is given by

$$\Gamma_b(\mathbf{W}(t), \mathbf{P}(t)) = \frac{P_b(t)\left|\mathbf{h}_{b,b}^{\dagger}(t)\,\mathbf{w}_b(t)\right|^2}{+\sigma^2}, \tag{5}$$

where we define $\mathbf{W}(t) \equiv [\mathbf{w}_1(t), \mathbf{w}_2(t), \ldots, \mathbf{w}_B(t)]$ and $\mathbf{P}(t) \equiv [P_1(t), P_2(t), \ldots, P_B(t)]$. The corresponding achievable instantaneous rate is given by

$$R_b(\mathbf{W}(t), \mathbf{P}(t)) = \log_2(1 + \Gamma_b(\mathbf{W}(t), \mathbf{P}(t))). \tag{6}$$

In conventional radio cellular systems, the ICI term in (5) is often considered as an uncontrollable background noise, similar to the thermal noise. However, it is crucial to acknowledge the presence of a race condition among co-channel cells, where the BS serving a specific UE becomes an interferer to the co-channel UEs in neighboring cells. In this paper, we introduce a novel interference coordination method based on RL to effectively manage and control the ICI experienced within each cell.

III. REVIEW AND LIMITATIONS OF MULTI-AGENT RL APPROACHES

In this section, we present a comprehensive review of the current state-of-the-art multi-agent RL approaches used for distributed multi-cell interference coordination [16], [17], which serve as performance benchmarks for our proposed method. In the multi-agent framework, each BS in a cellular network acts as an RL agent operating in a shared environment. However, this multi-agent setup introduces nonstationarity to the environment, as each agent continually adjusts its policy to adapt to the actions of other agents. As a result, convergence of models in such nonstationary environments is slow, and the solutions generated tend to be suboptimal. To address this issue, constant information exchange between agents is essential. Additionally, the choice of reward function greatly impacts the performance of the system [21], [22]. While sophisticated reward functions can enhance performance, they necessitate substantial information exchange overhead.

In the following subsections, we provide a detailed examination of information exchange and reward function design techniques utilized in distributed multi-agent RL approaches. Subsequently, we discuss the limitations associated with multi-agent RL approaches.

A. INFORMATION EXCHANGE

In a multi-agent environment, relying solely on the local information observed by an individual agent is insufficient to capture the complete state of the environment. Consequently, each agent must engage in information exchange with other concurrent agents [16], [17] to obtain a holistic understanding of the environment's global state. However, this necessitates a significant allocation of radio resources, introducing a substantial overhead. Moreover, the aggregation of global information collected from all BSs results in an exceedingly *high-dimensional* state space, which hampers the efficiency of the learning process. In contrast, single-agent RL approaches circumvent the need for such additional radio resource overhead and entail much smaller state lengths, enabling more efficient learning.

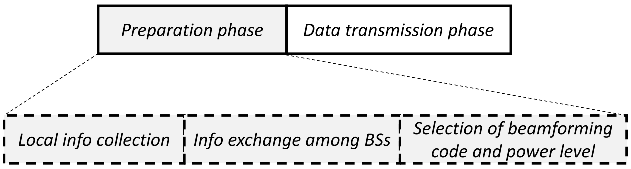

FIGURE 3. Detail of preparation phase.

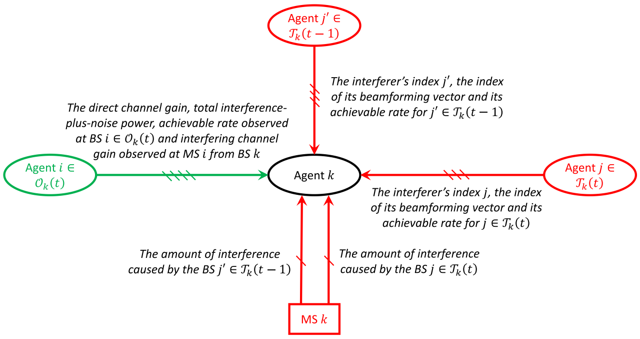
Agent $j' \in \mathcal{T}_k(t-1)$
The interferer's index j', the index of its beamforming vector and its achievable rate for $j' \in \mathcal{T}_k(t-1)$
The direct channel gain, total interference-plus-noise power, achievable rate observed at BS $i \in \mathcal{O}_k(t)$ and interfering channel gain observed at MS i from BS k
Agent $i \in \mathcal{O}_k(t)$
Agent k
Agent $j \in \mathcal{T}_k(t)$
The interferer's index j, the index of its beamforming vector and its achievable rate for $j \in \mathcal{T}_k(t)$
The amount of interference caused by the BS $j' \in \mathcal{T}_k(t-1)$
The amount of interference caused by the BS $j \in \mathcal{T}_k(t)$
MS k

FIGURE 4. Information exchange in the second sub-phase of preparation phase for multi-agent RL setup in [17].

In the framework design [17], each time slot is divided into two phases: the preparation phase and the data transmission phase, as depicted in Fig. 3. The preparation phase comprises three sub-phases, where the first two sub-phases are dedicated to gathering information for the transmit beamforming codes and power levels in the subsequent third sub-phase. The subsequent third phase involves the downlink data transmission. Similar to [17] and [22], the environment state observed by each agent encompasses information from three key aspects: the agent's local side, the surrounding interferers, and the interfered neighbors. Specifically, during the first sub-phase of the preparation phase, the agent collects seven *local* information elements from its own perspective. The first three elements correspond to the indices of the transmit power, beamforming vector, and achievable rate from the previous time slot. The remaining four elements are obtained through feedback from the UE being served, including direct downlink channel gains and interference-plus-noise powers obtained from the last two measurements.

Before delving into the details of the information exchange among all BSs during the second sub-phase of the preparation phase, let us revisit the concept of the two neighborhood sets associated with the direct link of each BS, as illustrated in Fig. 2. These sets consist of the interferers (i.e., the interfering BSs) and the interfered neighbors (i.e., the interfered UEs). Specifically, for the kth direct link between BS k and UE k, the interferer set, denoted as $\mathcal{T}_k(t)$, contains the indices of the BSs that cause significant interference to UE k at time slot t. Conversely, the interfered set, denoted as $\mathcal{O}_k(t)$, comprises the indices of UEs that experience interference from BS k at time slot t. Taking BS 0 as an example, as illustrated in Fig. 2, the interferer set $\mathcal{T}_0(t) = \{1, 2, \ldots, 6\}$ represents the BSs that impose substantial interference on the direct link between BS 0 and UE 0. Furthermore, BS 0 itself causes significant interference to UEs in the interfered set $\mathcal{O}_0(t) = \{1, 2, .., 6\}$.

After agent k takes an action at time slot t, it acquires information about the changes in the surrounding environment and the impact it has on the interfered neighbors by exchanging information with the interferers in $\mathcal{T}_k(t)$ and the interfered neighbors in $\mathcal{O}_k(t)$, respectively. According to [17], the information exchange process is summarized in Fig. 4. Through this procedure, at each time slot t, agent k gathers $4|\mathcal{T}_k(t)| + 4|\mathcal{T}_k(t-1)|$ pieces of information from the interferers in the current and previous time slots. Additionally, $4|\mathcal{O}_k(t)|$ pieces of information are collected from the interfered neighbors in $O_k(t)$. Thus, in the second sub-phase of the preparation phase, each agent k needs to gather a total of $4|\mathcal{T}_k(t)|+4|\mathcal{T}_k(t-1)|+4|\mathcal{O}_k(t)|$ pieces of information at each time slot t.

B. REWARD FUNCTION DESIGN

Due to the distributed nature of multi-agent RL, each agent autonomously optimizes its achievable rate by selecting actions it considers best. However, this decentralized framework can lead to agents causing significant interference to one another, thereby diminishing the overall system sum rate. This interference is a consequence of the intricate interactions among simultaneous agents, resulting in nonstationary environments. Nonstationary environments pose challenges in designing the reward function, as they require complex computations and substantial radio resource overhead.

In multi-agent environments, the design of an appropriate RL reward function plays a crucial role in optimizing the overall system performance [21], [22]. In [17], a reward function is formulated to maximize the achievable rate as defined in (6). The reward function computed by agent b at time slot t is given by the following expression:

$$f_b(t) = R_b(\mathbf{W}(t), \mathbf{P}(t)) - \Psi_b(\mathbf{W}(t), \mathbf{P}(t)), \tag{7}$$

where $R_b(\mathbf{W}(t), \mathbf{P}(t))$ is the instantaneous rate achieved by BS b in (6) and the function $\Psi_b(\mathbf{W}(t), \mathbf{P}(t))$ serves as a *penalty* imposed on the agent b. This penalty represents the overall *reduction* in achievable rate caused by the interference from BS b to the interfered BSs in $\mathcal{O}_b(t+1)$. It is defined by

$$\begin{aligned}
&\Psi_b(\mathbf{W}(t), \mathbf{P}(t)) \\
&= \sum\nolimits_{j \in \mathcal{O}_b(t+1)} \Big\{ \log_2 \Big(1 + \frac{P_j(t) \left|\mathbf{h}_{j,j}^{\dagger}(t)\, \mathbf{w}_j(t)\right|^2}{\sum_{i \neq b,j} P_i(t) \left|\mathbf{h}_{i,j}^{\dagger}(t)\, \mathbf{w}_i(t)\right|^2 + \sigma^2}\Big) \\
&\quad - R_j(\mathbf{W}(t), \mathbf{P}(t)) \Big\}, \ \forall b
\end{aligned} \tag{8}$$

In (8), the interference caused by BS b is subtracted from the SINR of each interfered BS j in $\mathcal{O}_b(t+1)$. Therefore, the achievable rate of BS j is calculated without considering the interference from BS b. This indicates that a higher penalty Ψ_b in (8) indicates that the action taken by agent b introduces significant interference to the system, thereby leading to a reduced reward in (7).

C. LIMITATIONS OF MULTI-AGENT RL APPROACHES

In a multi-agent RL environment, the observed environment state by each agent consists a tuple of $7+4\left|\mathcal{I}_k\left(t\right)\right|+4\left|\mathcal{I}_k\left(t-1\right)\right|+4\left|\mathcal{O}_k\left(t\right)\right|$ environment features. Among these, 7 dimensions are derived from the local side, while the remaining dimensions arise from the information exchange among agents. Consequently, the state space becomes *highly dimensional*, posing challenges in practical implementation. For instance, in [17], even when considering only the first-tier of co-channel cells (i.e., a set size of 6 for both interferers and interfered neighbors), the state dimension escalates to 79. High-dimensional RL state spaces present two challenges. Firstly, they necessitate a substantial allocation of radio resources, which reduces the available resources for actual data transmission. Thus, this resource allocation overhead can impact the overall system performance.

Secondly, high-dimensional state spaces make it more challenging to learn optimal solutions. As the state space grows larger, the learning process becomes more complex and computationally intensive, making it harder to converge to an optimal solution. Thus, addressing the challenges posed by high-dimensional state spaces is crucial in developing practical and efficient interference coordination methods.

IV. PROBLEM FORMULATION AND DERIVATION OF EXAMPLE-BASED RL VIA BINARY CLASSIFICATION

The multi-agent distributed framework encounters three challenges: the design of reward functions, the adaptation to nonstationary environments, and the significant requirement of information exchange overhead. They have led to limited exploration of single-agent RL models, primarily due to the difficulty of formulating an appropriate reward function to maximize the sum rate of cellular networks.

Motivated by these challenges, we propose a novel single-agent RL approach that eliminates the need for a reward function, effectively addressing these difficulties. In this section, we provide the problem formulation and derive an example-based RL approach that utilizes binary classification to solve the problem. By leveraging success examples and avoiding the reward function design and information exchange, our method offers a practical and efficient solution for interference coordination in cellular networks.

A. PROBLEM FORMULATION

We aim to jointly optimize the transmit beamforming vectors in $\mathbf{W}\left(t\right)$ and power control in $\mathbf{P}\left(t\right)$ to maximize the following achievable sum rate:

$$\begin{aligned} R\left(\mathbf{W}\left(t\right),\mathbf{P}\left(t\right)\right) &\equiv \sum\nolimits_{b=1}^{B} R_b\left(\mathbf{W}\left(t\right),\mathbf{P}\left(t\right)\right) \\ &= \sum\nolimits_{b=1}^{B} \log_2\left(1+\Gamma_b\left(\mathbf{W}\left(t\right),\mathbf{P}\left(t\right)\right)\right) \end{aligned} \tag{9}$$

More generally, our goal is to maximize the achievable sum rate objective function in (9) with respect to $\mathbf{W}\left(t\right)$ and $\mathbf{P}\left(t\right)$, subject to specific constraints:

$$\max_{\mathbf{W}(t)\in\mathcal{W}\mathbf{P}(t)\in\mathcal{P}} \sum\nolimits_{b=1}^{B} \log_2\left(1+\Gamma_b\left(\mathbf{W}\left(t\right),\mathbf{P}\left(t\right)\right)\right) \tag{10a}$$

$$\text{subject to } \Lambda_b\left(\mathbf{W}\left(t\right),\mathbf{P}\left(t\right)\right) \gtreqless \Lambda^*, \forall b \tag{10b}$$

In (10a), $\mathcal{W}$ refers to the beamforming codebook, and $\mathcal{P}$ represents the set of transmit power levels. In (10b), the set of functions $\left\{\Lambda_b\left(\mathbf{W}\left(t\right),\mathbf{P}\left(t\right)\right),\forall b\right\}$ is utilized to impose specific constraints on the problem and Λ^* represents a pre-defined positive threshold. The symbol $\gtreqless$ represents either the greater than or equal to $(\geq)$ or less than or equal to $(\leq)$ operators, depending on the specific type of constraints. For instance, with rate constraints, the constraint function $\Lambda_b\left(\mathbf{W}\left(t\right),\mathbf{P}\left(t\right)\right)$ corresponds to the rate of BS b (i.e., $R_b\left(\mathbf{W}\left(t\right),\mathbf{P}\left(t\right)\right)$ in (6)), and Λ^* denotes a predetermined minimum requirement for the achievable rate. Then the constraints in (10b) ensure that the rate of each UE in the system will be not below the minimum rate requirement Λ^*. In this case, the constraints in (10b) are equivalently expressed as the set of inequalities: $R_b\left(\mathbf{W}\left(t\right),\mathbf{P}\left(t\right)\right) \geq \Lambda^*, \forall b$.

While existing multi-agent RL approaches [17], [22] focus on maximizing the system sum rate as stated in (10a), they cannot guarantee the fulfillment of the minimum rate requirement for each BS. In contrast, our proposed approach provides a straightforward integration of the constraints specified in (10b) into the maximization of the achievable sum rate in (10a). This integration is achieved through the use of success examples, which guide the RL model towards optimal solutions that satisfy the specified constraints.

B. EXAMPLE-BASED RL WITH BINARY CLASSIFICATION FOR THE OPTIMIZATION PROBLEM IN (10)

Now, we proceed to derive the example-based RL without a reward function, which allows us to train the agent to effectively tackle the task outlined in (10). This approach utilizes success examples to maximize the sum rate in (10a) while adhering to the constraints specified in (10b). Unlike conventional goal-conditioned RL methods, our RL agent is trained in a versatile manner to handle various tasks, rather than solely aiming to achieve a specific goal. This diversity in training enables the agent to acquire policies that can successfully address tasks in novel environments, even those that present previously unseen success examples [23], [24], and [27].

Our learning method is primarily inspired by the C-learning algorithm, which pioneered the solution to classic goal-conditioned RL by predicting and controlling future states of the environment [26]. Based on the experiences gathered from a particular policy, the C-learning algorithm trains the agent to predict the futures states associated with a different policy. By acquiring the capability to predict future states, we can influence subsequent states through policies that lead to the desired future states. This entails the estimation of the probability density function of future states, which is a challenging task. Rather than directly training the estimator, the C-learning algorithm employs contrastive learning to train a binary classifier that can discriminate between "future states" and "random states". Subsequently, the learned classifier is employed to derive the future state density function using Bayes' rule. In this paper, we leverage the same concept to train a binary classifier that predicts whether a given task is successfully solved at a specific time step.

To apply RL to address the optimization problem in (10), it is essential to establish a well-defined Markov process with the dynamics $p(s_{t+1}\,|\,s_t,a_t)$ and an initial state distribution $p_1(s_1)$, where s_t and a_t represent the state and action at time step t, respectively. We introduce a binary random variable $e_t\in\{0,1\}$ to indicate whether the *task* is solved at time step t. Thus, the function $p(e_t=1\,|\,s_t)$ represents the probability of solving the task given state s_t. With a policy $\pi_\phi(a_t\,|\,s_t)$ parameterized by ϕ, the function $p^\pi(e_t=1\,|\,s_t,a_t)$ represents the likelihood of the task being solved at the state s_t under the policy $\pi_\phi(a_t\,|\,s_t)$. The γ-discounted probability of solving the task at a *future*step $t^+\in\{t,t+1,\ldots\}$ can be *recursively* expressed as follows, relating the current and next time steps:

$$\begin{aligned}p^\pi(e_{t^+}=1\,|\,s_t,a_t) &= (1-\gamma)\,p^\pi(e_t=1\,|\,s_t,a_t)\\&\quad+\gamma\,\mathbb{E}_{p(s_{t+1}\,|\,s_t,a_t),\pi_\phi(a_{t+1}\,|\,s_{t+1})}\left[p^\pi(e_{t+}=1\,|\,s_{t+1},a_{t+1})\right].\end{aligned}\tag{11}$$

Based on the recursive identity in (11), the example-based control problem aims to find the policy $\pi_\phi(a_t\,|\,s_t)$ that maximizes the following likelihood:

$$\begin{aligned}&\arg\max_\pi p^\pi(e_{t+}=1)\\&=\arg\max_\pi \mathbb{E}_{p_1(s_1)},\left[p^\pi(e_{t+}=1\mid s_1,a_1)\right],\end{aligned}\tag{12}$$

where the future step t^+ takes values from $t^+\in\{1,2,\ldots\}$ as it starts from the initial state s_1. The objective $p^\pi(e_{t+}=1)$ in (12) serves as an equivalent reward function to be maximized in traditional RL. However, existing RL algorithms cannot be directly applied to the example-based control problem in (12) due to the unknown likelihood $p^\pi(e_t=1\,|\,s_t,a_t)$. Nevertheless, in this task-solving problem, the distribution of success states, $p(s_t\,|\,e_t=1)$, can be learned from experiences acquired through interactions with the environment.

Gathering experiences from environment can be prohibitively expensive in many applications. As a result, an *off-policy* version is often proposed, allowing the agent to learn a policy using experiences from a replay buffer $\mathcal{D}$ collected from other policies. This approach facilitates learning from two distinct datasets. The first dataset, the replay buffer $\mathcal{D}$, contains a sequence of off-policy transitions $\{(s_t,a_t,s_{t+1})\sim p(s_t,a_t,s_{t+1})\}$, offering valuable information about the *environment dynamics*. The second dataset consists of *success examples* in $\mathcal{S}^*=\{s^*\sim p(s_t\,|\,e_t=1)\}$, which serve to illustrate the desired the task that the agent aims to accomplish. Additionally, it is necessary to determine the frequency with which each state s_t is visited in order to properly define the example-based control problem. It has been noted in [27] that example-based control methods are robust to the choice of the state distribution being visited. Therefore, in our approach, we conduct example-based control using the "uniform distribution of success examples" denoted by $p_U(s_t\,|\,e_{t+}=1)$.

Now, let's outline the approach for predicting future success states. Similar to C-learning [26], we train a binary classifier $C^\pi_\theta(s_t,a_t)$, where θ represents the classifier's parameters. This classifier is specifically designed to indirectly estimate the probability distribution mentioned in (11). Its primary task is to discriminate between *positive* "success examples" and *negative* "unlabeled random transitions". To train the classifier, we need to sample success examples and unlabeled random transitions. Success examples in positive set are sampled from the conditional distribution $p^\pi(s_t,a_t\,|\,e_{t+}=1)$, while unlabeled random transitions in negative set are sampled from the marginal distribution $p(s_t,a_t)$.

It is worth noting that the sampled positive set represents an incomplete set of success examples, as some examples in the unlabeled set may also be positive. This raises the question of how to effectively train a classical classifier using such an atypical training set. Fortunately, the authors in [32] have demonstrated that, even with positive and unlabeled samples, a binary classifier can be trained to predict probabilities that differ only by a constant factor from the probabilities produced by a model trained on a typical training set consisting of completely labeled positive and negative samples. Based on their findings, we assign the weighting coefficients $p(e_{t+}=1)$ and 1.0 to the sets of positive and unlabeled examples, respectively. Thus, the Bayes optimal solution for our binary classifier is given by

$$\begin{aligned}&C^\pi_\theta(s_t,a_t)\\&=\frac{p(e_{t+}=1)\times p^\pi(s_t,a_t\,|\,e_{t+}=1)}{p(e_{t+}=1)\times p^\pi(s_t,a_t\,|\,e_{t+}=1)+1.0\times p(s_t,a_t)}.\end{aligned}\tag{13}$$

By applying Bayes' formula, we can equivalently express the probability distribution $p^\pi(e_{t+}=1\,|\,s_t,a_t)$ in (11) as $p^\pi(s_t,a_t\,|\,e_{t+}=1)\,p(e_{t+}=1)\big/p(s_t,a_t)$. Remarkably, these three probability functions in this equivalence are coincidentally present in (13). Consequently, the probability of successfully solving the task at the future step t^+ in (11) can be derived from the probability predicted by the binary classifier in (13), as shown below:

$$p^\pi(e_{t+}=1\,|\,s_t,a_t)=\frac{C^\pi_\theta(s_t,a_t)}{1-C^\pi_\theta(s_t,a_t)}.\tag{14}$$

We proceed to train the binary classifier C^π_θ to maximize the following objective function:

$$\begin{aligned}\mathcal{L}^\pi(\theta)\equiv&\underbrace{p(e_{t+}=1)\times\mathbb{E}_{p^\pi}(s_t,a_t\mid e_{t+}=1)\left[\log C^\pi_\theta(s_t,a_t)\right]}_{\text{(a)}}\\&+1\times\mathbb{E}_{p(s_t,a_t)}\left[\log\left(1-C^\pi_\theta(s_t,a_t)\right)\right]\end{aligned}\tag{15}$$

where the success examples and random transitions are weighted with coefficients $p(e_{t+} = 1)$ and 1.0, respectively. In (15), the conditional distribution $p^{\pi}(s_t, a_t \mid e_{t+} = 1)$ and marginal distribution $p(s_t, a_t)$ are the distributions from which the success examples and random transitions are sampled, respectively. Specifically, the training of the classifier involves distinguishing between success examples and random transitions by utilizing the first and second expectations in (15), respectively.

The second expectation in (15) can be estimated by using Monte Carlo examples sampled from the replay buffer D, i.e., $(s_t, a_t) \sim p(s_t, a_t)$. However, due to the unavailability of the conditional distribution $p^{\pi}(s_t, a_t \mid e_{t+} = 1)$, we are unable to sample success examples to estimate the first expectation. Nevertheless, let's examine term (a) in (15), corresponding to the probability distribution $p(e_{t+} = 1)\, p^{\pi}(s_t, a_t \mid e_{t+} = 1)$, which can be factored into the product of $p^{\pi}(e_{t+} = 1 \mid s_t, a_t)$ and $p(s_t, a_t)$. Notably, $p^{\pi}(e_{t+} = 1 \mid s_t, a_t)$ is expressed by the *recursive identity* in (11) and can be estimated using the classifier's predictions in (14). As in [27], we substitute (14) and (11) into term (a) in (15) to reformulate the objective function as follows:

$$\begin{aligned}
&\mathcal{L}^{\pi}(\theta) \\
&\equiv (1-\gamma)\mathbb{E}_{p_U(s_t|e_{t+}=1),p(a_t|s_t)}\big[1 \times \underbrace{1 \times \log C_{\theta}^{\pi}(s_t, a_t)}_{\textbf{(a)}}\big] \\
&+\mathbb{E}_{p(s_t,a_t,s_{t+1})}[\underbrace{\gamma\omega \times \log C_{\theta}^{\pi}(s_t, a_t)}_{\textbf{(b)}}+\underbrace{\log\left(1 - C_{\theta}^{\pi}(s_t, a_t)\right)}_{\textbf{(c)}}]
\end{aligned} \tag{16}$$

where ω represents the expected prediction ratio of the binary classifier C_{θ}^{π}, as expressed in (14), at the next time step:

$$\omega \equiv \mathbb{E}_{p(a_{t+1} \mid s_{t+1})}\left[\frac{C_{\theta}^{\pi}(s_{t+1}, a_{t+1})}{1 - C_{\theta}^{\pi}(s_{t+1}, a_{t+1})}\right]. \tag{17}$$

In (16), the terms (a) and (b) correspond to term (a) in (15), which is associated with success examples. Furthermore, in (16), the binary classifier C_{θ}^{π} is trained to predict 1.0 and $\gamma\omega/(1+\gamma\omega)$ for the current success examples (term (a)) and the success examples the next time step (term (b)), respectively. For term (c), the classifier is trained to predict 0 for random transitions.

Now we utilize the classifier C_{θ}^{π} to assess actions generated by the policy π_{ϕ}. In this context, the binary classifier and RL policy act as the critic and actor networks [33], respectively. Based on the objective function in (16), the classifier C_{θ}^{π} is trained to minimize the following loss function of the critic network, consisting of two cross-entropy (CE) losses:

$$\begin{aligned}
&\min_{\theta}\{(1-\gamma) \\
&\times\mathbb{E}_{p_U(s_t|e_{t+}}=1\big), p(a_t \mid s_t)\left[CE\left(C_{\theta}^{\pi}(s_{t+1}, a_{t+1}); y = 1\right)\right] \\
&+(1+\gamma\omega)\ \times\mathbb{E}_{p(s_t,a_t,s_{t+1})} \\
&\times\left[CE\left(C_{\theta}^{\pi}(s_{t+1}, a_{t+1}); y{=}\gamma\omega/(1+\gamma\omega)\right)\right]\}\,,
\end{aligned} \tag{18}$$

where $CE(\cdot;\cdot)$ denotes the binary CE loss. The first CE loss in (18) corresponds to the success examples, which are uniformly sampled from the set of success examples, S^*. These success examples are assigned a positive label $y = 1$. On the other hand, the second CE loss is related to experiences sampled from the unlabeled replay buffer D. Note that while sampling from the unlabeled replay buffer, an experience can be either a positive success example or a negative random transition. According to [32], we assign the unlabeled samples the label $y = \gamma\omega/(1+\gamma\omega)$.

Given the classifier C_{θ}^{π}, the policy π_{ϕ} is updated to select actions that maximize the classifier's confidence in solving the task in the future:

$$\max_{\phi} \mathbb{E}_{\pi_{\phi}(a_t|s_t)}\left[C_{\theta}^{\pi}(s_t, a_t)\right] \tag{19}$$

The policy objective in (19) aligns with the objective used in an actor-critic iterative algorithm. In each iteration, we alternate between updating the critic and actor networks. At the end of each iteration, we store the transition in the replay buffer D. The actor and critic networks are iteratively updated until convergence is achieved or a predefined number of iterations is reached.

V. THE PROPOSED SIGNLE-AGENT RL-BASED APPROACH FOR JOINT TRANSMIT BEAMFORMING AND POWER CONTROL COORDINATION

In this section, we present our example-controlled RL approach for enhancing system sum rate in wireless communication systems. The section is divided into four subsections, each addressing a crucial aspect of our approach.

A. SYSTEM ARCHITECHTURE WITH EXAMPLE-CONTROLLED RL MODEL AND ACHIEVING CONSISTENT SIGNAL QUSLITY LEVEL

The system architecture, depicting the interaction between the agent and the environment, is illustrated in Fig. 5. The agent consists of two components: the critic and actor networks. The environment represents a cellular network comprising B BSs. In our proposed approach, the RL agent serves as the central controller, establishing connections with all the BSs through a backhaul network. Within each cell, multiple UEs operate simultaneously, with each UE being assigned a distinct orthogonal subchannel.

In each cell, the UEs are randomly distributed within the coverage area, and their perceived signal quality is largely influenced by the distance from the serving BS. To achieve a consistent signal quality level, network MIMO transmission techniques have been utilized [2], [3], [4]. However, these techniques often introduce significant processing overhead and implementation complexity [5]. Additionally, previous works such as [16] and [17] have explored distributed multi-agent RL approaches to maximize the achievable sum rate in (9).

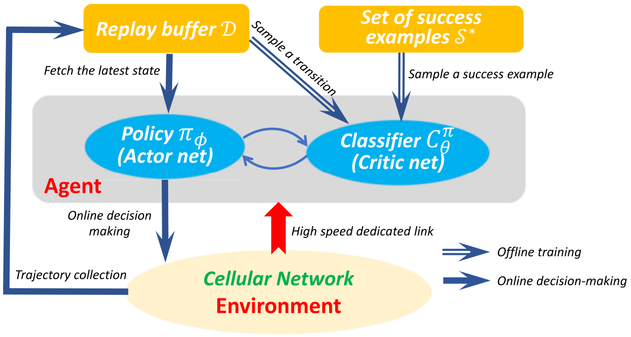
Replay buffer $\mathcal{D}$
Set of success examples $\mathcal{S}^*$
Sample a transition
Fetch the latest state
Sample a success example
Policy π_ϕ (Actor net)
Classifier C_θ^π (Critic net)
Agent
Online decision making
High speed dedicated link
Offline training
Online decision-making
Trajectory collection
Cellular Network Environment

FIGURE 5. The system architecture showcasing the agent-environment interaction.

Nevertheless, these approaches often prioritize the sum rate at expense of UEs located at the cell edges. This trade-off arises because maximizing the system sum rate in (9) relies on the water-filling principle [34], which allocates more power to UEs with better channel conditions and less power to UEs with poorer channel conditions. As a result, these approaches cannot ensure that each BS achieves a rate higher than a predetermined minimum threshold.

To enhance the signal quality for UEs with poor channel conditions, the implementation of rate constraints can be highly beneficial. Specifically, we propose using example control methods that indirectly impose constraints on the maximization of the objective function in (10a) using success examples, whose achievable rates are higher than a preset minimum rate. This involves defining the constraint function in (10b) based on the individual rates specified in (6). By doing so, it is ensured that the minimum individual rate among UEs remains above the predetermined threshold. This effectively improves the signal quality and performance for UEs with unfavorable channel conditions.

B. SORTED CHANNEL ASSIGNMENT FOR ENHANCED SYSTEM SUM RATE

To further enhance the overall system capacity, we also devise a simple yet effective method. Inspired by the concept in the condition number of channel matrix [34], which suggests that co-channel UEs with similar channel gains can potentially improve the system capacity, we propose a strategy to leverage this insight. Our approach involves the assignment of a group of co-channel UEs with comparable channel conditions to a specific frequency band. To achieve this, we utilize received signal strength (RSS) measurements as a basis for grouping UEs.

The UEs within each cell are sorted in descending order according to their RSS measurements, as illustrated in Fig. 6, where the same color or numbering indicates the utilization of the same frequency band. Assuming a consistent numbering of frequency bands across cells, each BS assigns the first frequency band to the UE closest to it, the second band to the next closest UE, and so on. We term this technique as *sorted channel assignment* (SCA). By utilizing this frequency planning technique, the co-channels UEs operating on the same band will have similar serving distances, in contrast to *random channel assignment* (RCA), where the channel assignment is unrelated to serving distance.

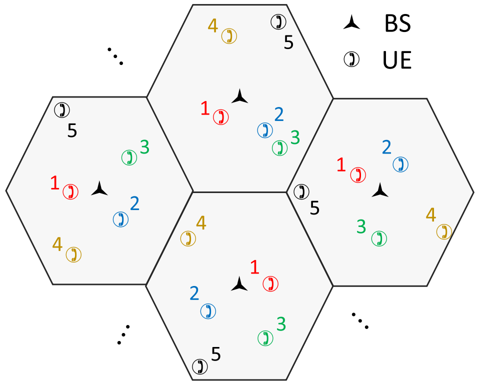

FIGURE 6. Sorting the UEs within each cell in descending order based on their RSS measurements.

SCA, through its effective grouping of co-channel UEs with similar large-scale channel conditions, contributes to reducing the condition number of the corresponding channel matrix for each group. This reduction is desirable since a condition number close to 1 indicates a well-conditioned channel matrix, resulting in increased capacity.

In summary, the combination of rate constraints in Subsection V-A and the implementation of SCA in Subsection V-B provides effective strategies for improving signal quality for UEs with poor channel conditions and further increasing the overall system capacity.

C. GNENRATION OF SUCCESS EXAMPLE SET

Recall that $\mathcal{W}$ and $\mathcal{P}$ represent the beamforming codebook and set of possible transmit power levels, respectively. Thus, the *action space* $\mathcal{A}$ is defined as $\mathcal{A} = \{(p, \mathbf{w}), p \in P, \mathbf{w} \in \mathcal{W}$. Each element in $\mathcal{A}$ consists of a pair of index values, one for the beamforming vector and the other for transmit power level. As there are B BSs in the cellular network, the action vector generated by our agent will be $2B$-dimensional. To meet a transmit power constraint $p_{\max}$, we evenly divide the power range $[0,p_{\max}]$ into $N_{\mathcal{P}}$ discrete levels. Accordingly, the set of possible transmit power levels is defined as $\mathcal{P} = \{0, \Delta_{\mathrm{p}}, 2\Delta_{\mathrm{p}}, \ldots, p_{\max}\}$, where $\Delta_{\mathrm{p}} \equiv p_{\max} \big/ (N_{\mathcal{P}} - 1)$. The codebook W comprises $N_{\mathcal{W}}$ beam codes, with each code specifying a distinct beam direction. We adopt the codebook design utilized [17], which allows for a higher number of codes than antenna elements, i.e., $N_{\mathcal{W}} \geq N$. The ith element of the kth code is given as follows:

$$\mathbf{w}_k[i] = \frac{1}{\sqrt{N}} \exp\left(\jmath \frac{2\pi}{S} \left\lfloor i \times \operatorname{mod}\left(k + \frac{N_{\mathcal{W}}}{2}, N_{\mathcal{W}}\right) \Big/ \frac{N_{\mathcal{W}}}{S} \right\rfloor \right), \tag{20}$$

where S represents the number of available phases for each antenna element. In this paper, we set the value of S to be 16.

In our proposed RL model, the agent acquires environment states from the connected BSs. The state is represented by a simple two-dimensional (2)-D) vector. The first entry corresponds to the *achievable system sum rate*, which is the optimization goal defined in (10a). The second entry relates to the constraints outlined in (10b). The specific definition of second state entry depends on the choice of the constraint function $\Lambda_b(\mathbf{W}(t), \mathbf{P}(t))$ in (10b). It can be specified in various ways, such as the *minimum* individual achievable rate, expressed as $\min_b\{R_b(\mathbf{W}(t), \mathbf{P}(t))\}$ or the *maximum* penalty, represented as $\max_b\{\Psi_b(\mathbf{W}(t), \mathbf{P}(t))\}$.

If we consider the minimum achievable rate constraint (i.e., $\Lambda_b(\mathbf{W}(t), \mathbf{P}(t)) \equiv R_b(\mathbf{W}(t), \mathbf{P}(t))$), the inequality constraints in (10b) ensure that the achievable rate of each BS *exceeds* the predetermined minimum threshold Λ^*, expressed as $\min_b\{R_b(\mathbf{W}(t), \mathbf{P}(t))\} \geq \Lambda^*$. Alternatively, if we adopt the penalty constraint (i.e., $\Lambda_b(\mathbf{W}(t), \mathbf{P}(t)) \equiv \Psi_b(\mathbf{W}(t), \mathbf{P}(t))$), the total achievable rate loss caused by any BS b is limited to be *less* than the minimum loss Λ^*, indicated by, $\max_b\{\Psi_b(\mathbf{W}(t), \mathbf{P}(t))\} \leq \Lambda^*$.

Our example-controlled RL model exhibits the ability to learn and solve a wide range of tasks defined by a collection of success examples in S^*. Each success example is characterized by a 2-D state that fulfills two conditions, one pertaining to the first state entry and the other concerning the second entry. As a result, the proposed single-agent RL model operates in a compact state space with a dimensionality of 2. In contrast, the distributed multi-agent RL method [17] has a much higher state dimensionality, reaching up to 79 even when considering a set size of 6 for both interferers and interfered neighbors.

The incorporation of success examples in our model facilitates the learning of more general notions of success. To achieve this, we train the binary classifier C_θ^π to effectively differentiate between success examples and random transitions, as explained in Section IV-B. The training process involves an atypical training dataset that comprises both the set of *success* examples S^* and the *unlabeled* examples from the replay buffer $\mathcal{D}$.

To generate the 2-D success examples in S^*, we consider two pairs of *positive* thresholds and margins, $\{\eta_1, \Delta_1\}$ and $\{\eta_2, \Delta_2\}$, to impose constraints on the first and second entries of success examples, respectively. The first entry corresponds to the system sum rate in (10a) and is randomly selected from the interval $[\eta_1, \eta_1 + \Delta_1]$. While higher sum rates are desirable, we need to set an upper bound $\eta_1 + \Delta_1$ on the system sum rates of success examples due to the limited capacity of cellular networks.

The second entry of a success example is associated with the constraints imposed on the objective function maximization in (10a). In this paper, we consider two constraint functions in (10b): the penalty Ψ_b in (8) and the individual achievable rate R_b in (6). By using the penalty, we can limit the *total loss of achievable rate* caused by any BS b. In this case, the second entries of success examples are uniformly sampled from the interval $[\eta_2 - \Delta_2, \eta_2]$, where $\eta_2 - \Delta_2$ is the lower bound of the total loss with $\eta_2 > \Delta_2 > 0$. Alternatively, to ensure a minimum *achievable rate* for each UE, we set the constraint function in (10b) to the UE's individual achievable rate, i.e., $\Lambda_b(\mathbf{W}(t), \mathbf{P}(t)) = R_b(\mathbf{W}(t), \mathbf{P}(t))$. In this case, the second entry of a success examples are randomly chosen from the interval $[\eta_2, \eta_2 + \Delta_2]$, where $\eta_2 + \Delta_2$ is the upper bound of the individual achievable rate.

Let's illustrate the generation of success examples. Assuming the first threshold/margin pair is $\{\eta_1 = 8, \Delta_1 = 4\}$, the first state entries are thus uniformly sampled from the intervals $[\eta_1, \eta_1 + \Delta_1] = [8, 12]$. This indicates that the system sum rate, which is to be maximized according to (10a), will be greater than 8 but upper bounded by 12. Assume threshold/margin pair $\{\eta_2 = 2, \Delta_2 = 1\}$ for the second state entry. If we employ the penalty constraint, the second state entry are thus randomly chosen from the interval $[\eta_2 - \Delta_2, \eta_2] = [1, 2]$. This implies that the total loss of achievable rate caused by any BS will be less than 2 but lower bounded by 1. Alternatively, employing the rate constraint, the second state entries are randomly sampled from the interval $[\eta_2, \eta_2 + \Delta_2] = [2, 3]$. Thus, the achievable rate for each UE will be greater than 2 but upper bounded by 3.

D. ALGORITHM OF OUR EXAMPLE-CONTROLLED RL APPROACH

The system architecture of our example-controlled RL approach is presented in Fig. 5, and the corresponding algorithm is summarized in Table 1. In Table 1, variables with superscripts (p) and (n) are associated with success examples and random transitions, respectively. The algorithm operates in an iterative fashion, alternating between improving the classifier C_θ^π (critic network with parameter set θ) and the policy π_ϕ (actor network with parameter set ϕ).

During each training iteration, the algorithm follows Step 1 to Step 5 for the offline training of the binary classifier C_θ^π, and Step 6 to Step 9 for the online update of the policy π_ϕ. In Step 1, we sample a state $s_t^{(\mathrm{p})}$ from the success example set S^* and generate an action $a_t^{(\mathrm{p})}$ conditioned on the state $s_t^{(\mathrm{p})}$ according to the policy π_ϕ. Similarly, in Step 2, a random transition $(s_t^{(\mathrm{n})}, a_t^{(\mathrm{n})}, s_{t+1}^{(\mathrm{n})})$ is sampled from the replay buffer $\mathcal{D}$, and an action $a_{t+1}^{(\mathrm{n})}$ conditioned on the state $s_{t+1}^{(\mathrm{n})}$ is generated using the policy π_ϕ. Step 3 involves calculating the probability of successfully solving the task at the next time step as defined by (14). Next, in Steps 4 and 5, we update the classifier C_θ^π based on the gradient of the loss function $\mathcal{L}(\theta)$ with respect to the parameter set θ. The output of classifier C_θ^π is the probability that the input is a success example. As specified in (18), the loss function $L(\theta)$ in Step 4 can be minimized if the classifier predicts a probability of 1 for success examples ($C_\theta^\pi\left(s_t^{(\mathrm{p})}, a_t^{(\mathrm{p})}\right) = 1$) and a probability of $\gamma\omega/(1 + \gamma\omega)$ at the *next time step* for random transitions ($C_\theta^\pi\left(s_t^{(\mathrm{n})}, a_t^{(\mathrm{n})}\right) = \gamma\omega/(1 + \gamma\omega)$). This training

TABLE 1. Algorithm of our example-controlled RL approach.

```
  Initialization: Both classifier $C_\theta^\pi$ and policy $\pi_\phi$
  Input: Replay buffer $\mathcal{D}$, success examples $\mathcal{S}^*$
  Output: $\pi_\phi$
  do
1     Sample a success example and an action:
        $\{s_t^{(\mathrm{p})} \sim \mathcal{S}^*,\ a_t^{(\mathrm{p})} \sim \pi_\phi(a|s_t^{(\mathrm{p})})\}$
2     Sample a random transition and an action:
        $\{(s_t^{(\mathrm{n})}, a_t^{(\mathrm{n})}, s_{t+1}^{(\mathrm{n})}) \sim \mathcal{D},\ a_{t+1}^{(\mathrm{n})} \sim \pi_\phi(a|s_{t+1}^{(\mathrm{n})})\}$
3     Calculate the classifier's prediction ratio at the
      next time step:
        $\omega \leftarrow C_\theta^\pi(s_{t+1}^{(\mathrm{n})}, a_{t+1}^{(\mathrm{n})}) / (1 - C_\theta^\pi(s_{t+1}^{(\mathrm{n})}, a_{t+1}^{(\mathrm{n})}))$
4     Get the loss function:
        $\mathcal{L}(\theta) \leftarrow (1-\gamma) CE\left(C_\theta^\pi(s_t^{(\mathrm{p})}, a_t^{(\mathrm{p})}); y = 1\right) +$
            $(1+\gamma\omega) CE\left(C_\theta^\pi(s_t^{(\mathrm{n})}, a_t^{(\mathrm{n})}); y = \frac{\gamma\omega}{(1+\gamma\omega)}\right)$
5     Update the classifier $C_\theta^\pi$: $\theta \leftarrow \theta - \eta \nabla_\theta \mathcal{L}(\theta)$
6     Fetch the latest state from $\mathcal{D} \rightarrow s_t$
7     Sample an action $a_t \sim \pi_\phi(a|s_t)$
8     Update the policy $\pi_\phi$:
        $\phi \leftarrow \phi + \eta \nabla_\phi \mathbb{E}_{\pi_\phi}[C_\theta^\pi(s_t, a_t)]$
9     Collect a new transition of experience:
        $\mathcal{D} \leftarrow \mathcal{D} \cup \{a_t, s_{t+1}\}$
  while not converged
```

TABLE 2. Thresholds and margins for generation of success examples.

	First state entry (System sum rate)	Second state entry (Constraints)
Rate constraint	$\eta_1 = 7.9, \Delta_1 = 6.9$	$\eta_2 = 4.9, \Delta_2 = 6.9$
Penalty constraint	$\eta_1 = 7.9, \Delta_1 = 6.9$	$\eta_2 = 5.9, \Delta_2 = 5.9$

approach, where the classifier is trained to predict the success probability for the next time step, resembles the bootstrapping in temporal difference learning [35].

Next, we proceed with the *online* update of the policy π_ϕ in Steps 6 to 9. Initially, in Step 6, we retrieve the most recent state s_t from the replay buffer $\mathcal{D}$. Conditioned on s_t, the policy (agent) generates an action a_t, which is then applied to the environment in Step 7. Following the application of the action, we obtain the latest state-action pair (s_t, a_t). In Step 8, we perform the gradient-ascent update on the policy π_ϕ, aiming to *maximize* the classifier's prediction $C_\theta^\pi(s_t, a_t)$ as defined in (19).

During the online learning process, after the agent applies the action a_t to the environment, the environment state transitions from the current state s_t to the subsequent state s_{t+1}. Hence, at the end of each iteration, we gather a new transition (s_t, a_t, s_{t+1}) and add it to the replay buffer $\mathcal{D}$ in Step 9. This enables the accumulation of additional training data for subsequent iterations. Through this iterative procedure, both the classifier and policy networks are updated until convergence is achieved or a predetermined number of training iterations is reached.

VI. SIMULATION RESULTS

In this section, we evaluate the performance based on the *average achievable rate*. The average achievable rate is the average rate per UE, which is calculated by dividing the system sum rate in (9) by the number of co-channel UEs. To establish benchmarks, we compare our results with two existing methods: the FP algorithm [15] and a multi-agent RL scheme [17]. The FP algorithm is a theoretical method capable of generating beam patterns for *arbitrary* AODs with *continuous* transmit power levels. Due to the infinite size of its action space, the FP algorithm is only applicable in theory and not feasible in practical implementations. On the other hand, the multi-agent RL method proposed in [17] is more practical but requires a significant amount of radio resource overhead for information exchange among agents, as discussed in Section III-A.

For the evaluation, we consider a two-tier homogeneous cellular network of 19 hexagonal cells. The cell radius is set to 200 meters and each BS is equipped with three antennas. To ensure accurate simulations, UEs located within 10 meters of the serving BS are excluded from our simulations. We assume that UEs are uniformly distributed within each cell. The channel parameters are set according to the benchmark method in [17]. The maximum transmit power constraint of the BSs is set to $p_{\max} = 38$ dBm. The large-scale fading coefficient $\beta_{j,k}$ between BS j and UE k incorporates both path loss and log-normal shadowing. The path loss is modeled as $120.9 + 37.6 \times \log_{10} d_{j,k}$ dB, where $d_{j,k}$ denotes the distance between BS j and UE k. The standard deviation of log-normal shadowing is 8 dB. We set the number of multipath L to 4 and the angular spread Δ to 3 degrees. The noise variance σ^2 is -114 dBm and the correlation coefficient ρ for small-scale fading is 0.64.

Unless explicitly stated otherwise, we use a codebook size of $N_{\mathcal{P}} = 5$ and four available power levels ($N_{\mathcal{W}} = 4$). The value of the second state entry depends on the selection of constraint functions in (10b). We consider two distinct constraints: the minimum achievable rate $\min_b\{R_b(\mathbf{W}(t), \mathbf{P}(t))\}$ (*rate constraint*) and the maximum penalty $\max_b\{\Psi_b(\mathbf{W}(t), \mathbf{P}(t))\}$ (*penalty constraint*). To generate success examples, we employ the thresholds and margins specified in Table 2. Unless explicitly indicated, we utilize the RCA as described in Section V. Also, in each of the following non-bar figures, the value for each time slot is calculated as the average of the preceding 500 time slots. The choice of a window size of 500 time slots for the moving average serves to smooth out short-term fluctuations and provide a clearer trend in the data. Decreasing the window size could lead to jaggedness and misrepresent trends. The 500 slots strike a balance for accurate representation.

In Fig. 7, we present a comparison of average achievable rates among different approaches, including the proposed method and two benchmarks: multi-agent RL and FP algorithm. The reward function of the multi-agent RL method is defined by (7), which incorporates penalties in (8). To ensure a fair comparison, we adopt the penalty function as the constraint function in (10b) for our proposed method. The results show that the proposed method significantly outperforms the multi-agent RL method significantly,

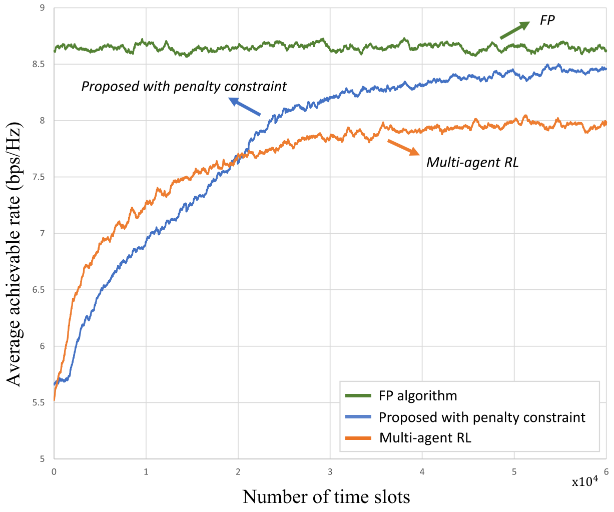

FIGURE 7. Average achievable rate comparisons of different approaches. ($N_{\mathcal{P}} = 5$ and $N_{\mathcal{W}} = 4$).

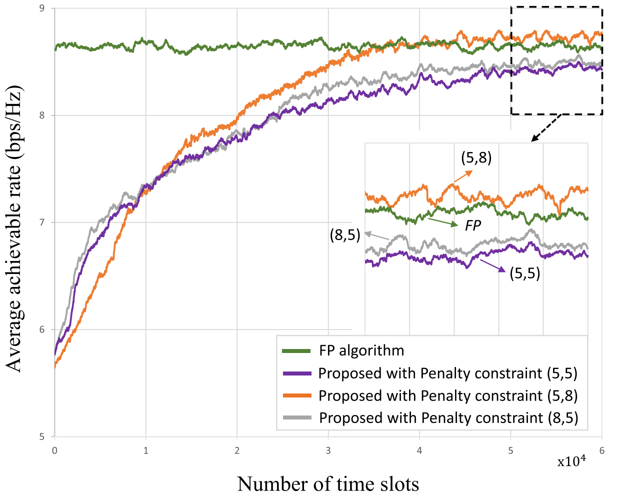

FIGURE 8. Average achievable rate comparison for different combinations of ($N_{\mathcal{P}}$,$N_{\mathcal{W}}$).

achieving approximately 98.18% of the average achievable rate of the theoretical FP algorithm. Several reasons contribute to the superior performance of the proposed method over multi-agent RL learning. Firstly, our proposed model eliminates the challenges associated with nonstationary environments, which often result in slow convergence and suboptimal solutions. Thus, the proposed method is able to converge to superior solutions. Secondly, we utilize success examples to guide the learning process, thereby avoiding the complexities associated with designing reward functions, that is particularly challenging in nonstationary environments. Consequently, the proposed method does not suffer from performance degradation caused by poorly designed reward functions. On the other hand, multi-agent models are inherently constrained by the nonstationary environment and the need to design suitable reward functions. Additionally, addressing nonstationary environments requires a substantial amount of information exchange overhead, which the proposed method effectively avoids.

Figure 8 illustrates a performance comparison for different combinations of $N_{\mathcal{P}}$ and $N_{\mathcal{W}}$. Specifically, among the possible combinations for an action space size of $A = 40$, we consider $(N_{\mathcal{P}}, N_{\mathcal{W}}) = (5, 8)$ and $(N_{\mathcal{P}}, N_{\mathcal{W}}) = (8, 5)$. By comparing these combinations to the baseline case of $(N_{\mathcal{P}}, N_{\mathcal{W}}) = (5, 5)$, we find that increasing the codebook size $(5, 8)$ is more effective in enhancing the system capacity compared to increasing the power level $(8, 5)$. This can be attributed to the fact that increasing the codebook size results in antenna beam patterns with narrower beamwidth, effectively reducing co-channel interference. Therefore, increasing the codebook size offers greater benefits to system capacity when the action space size $|A|$ is limited. To recap, in Fig. 7, the proposed method with $(N_{\mathcal{P}}, N_{\mathcal{W}}) = (5, 4)$ achieves 98.18% of the average achievable rate of the theoretical FP algorithm.

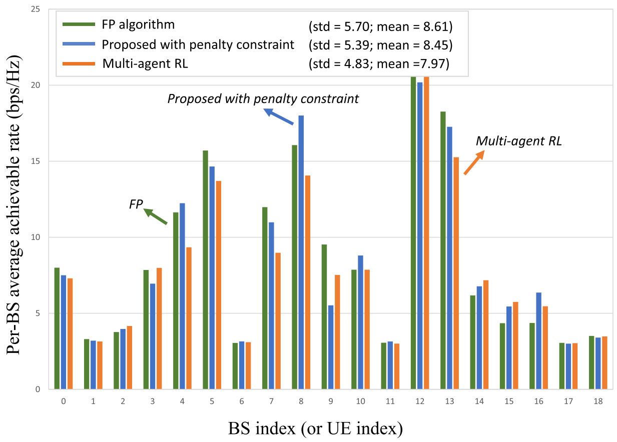

FIGURE 9. Per-BS achievable rate distribution across the 19 BSs corresponding to Fig. 7.

Interestingly, the results in Fig. 8 reveal that the proposed method with $(N_{\mathcal{P}}, N_{\mathcal{W}}) = (5, 8)$ can even outperform the suboptimal FP algorithm with an *infinite* action space size $(N_{\mathcal{P}} \rightarrow \infty, N_{\mathcal{W}} \rightarrow \infty)$. This observation highlights the suitability of the example-based RL approach utilizing binary classification for addressing the optimization problem outlined in (10).

To provide a deeper understanding of the system performance, Fig. 9 illustrates the distribution of achievable rates among these 19 BSs. It is evident that the individual rates of the BSs exhibit significant variations, with highest rate being 7.23 times greater than the lowest rate. Specifically, the standard deviation of the rates exceeds half of the average rate (std > mean/2), as shown in Fig. 9. This wide variation in individual rates arises due to the primary focus of the penalty constraint on maximizing the overall system sum rate, without considering the individual rates of the UEs.

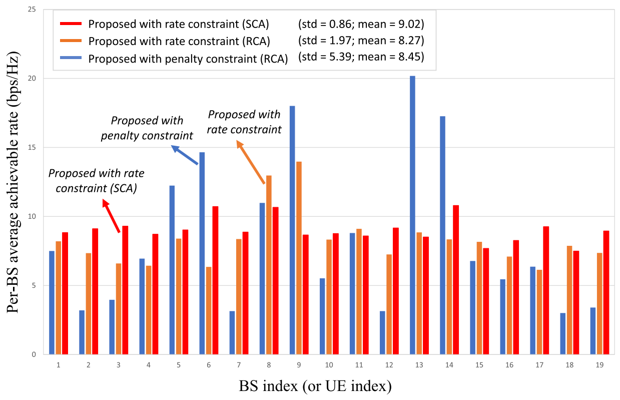

FIGURE 10. Per-BS achievable rate distribution across 19 BSs: comparison of different constraints and channel assignments.

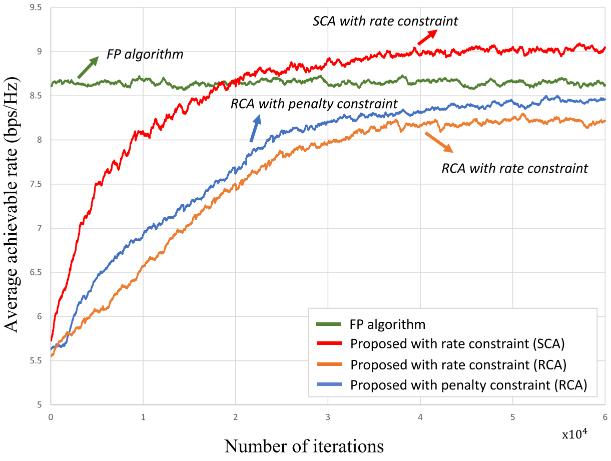

FIGURE 11. Average achievable rate corresponding to Fig. 10.

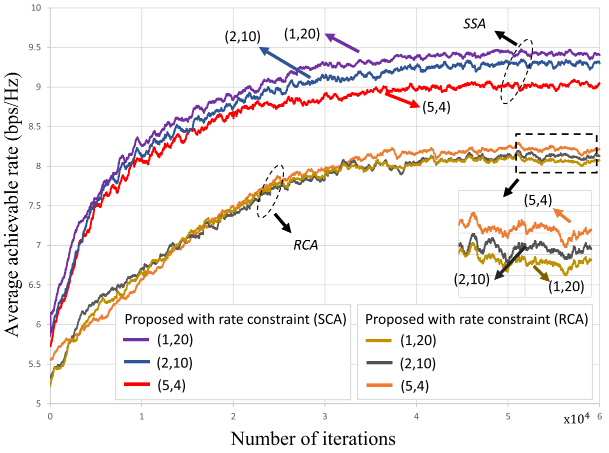

FIGURE 12. Combination selection of $N_{\mathcal{P}}$ and $N_{\mathcal{W}}$ for SCA and RCA under the constraint $|\mathcal{A}| = N_{\mathcal{P}} N_{\mathcal{W}} = 20$.

Consequently, more power is allocated to UEs with favorable channel conditions, while less power is allocated to UEs with poorer channel conditions [34].

As mentioned in Subsections V-A and V-B, employing rate constraints, in conjunction with SCA, provides an effective strategy for improving the signal quality of UEs with poor channel conditions and increasing the overall system capacity. In Fig. 10, we present a performance comparison of the proposed approach considering different constrains and channel assignments. The benchmark scenario utilizes the proposed method with penalty constraints and RCA, resulting in an achievable rate distribution with a standard deviation of 5.39 and a mean of 8.45. This indicates that the achievable rate distribution is significantly spread out relative to the mean. It should be noted that the penalty constraints, employed in the multi-agent model [17], not only result in an uneven rate distribution as shown in Fig. 9, but also require considerable radio resource overhead. To address these issues, we replace the penalty constraint with the rate constraint. Figure 10 demonstrates a substantial decrease in the standard deviation from 5.39 to 1.97 compared to using the penalty constraint, while the mean rate experiences a slight decrease from 8.45 to 8.27. In addition, by applying the SCA, we can further reduce the standard deviation and increase the mean rate. Figure 10 shows that replacing RCA with SCA in the proposed method with the rate constraint leads to a dramatic reduction in the standard deviation by half, from 1.97 to 0.86, while increasing the mean rate from 8.27 to 9.02. This improvement can be attributed to the utilization of SCA, which facilitates similar channel gains among co-channel UEs, thereby enhancing the overall system capacity.

Figure 11 illustrates the results of the average achievable rate corresponding to Fig. 10. When employing the RCA, it can be observed that using the rate constraint results in slightly lower average achievable rate compared to using the penalty constraint. This observation is consistent with the fact that the penalty constraint provides more comprehensive information about the environment compared to the rate constraint. However, the rate constraint offers the advantage of providing a more uniform rate distribution, as depicted in Fig. 10. This is because the rate constraint does not require information exchange between the BSs, unlike the penalty constraint, while still achieving a more balanced data rate among the UEs. Additionally, Fig. 11 demonstrates that the application of SCA to the proposed method with rate constraint even significantly outperforms the theoretical FP algorithm. As mentioned early, this improvement can be attributed to the ability of SCA to form co-channel UEs with relatively similar channel gains.

In order to reduce co-channel interference, it is indeed beneficial to have a finer granularity of elements in both the beamforming codebook $\mathcal{W}$ and the transmit power level set $\mathcal{P}$. However, this finer granularity leads to a large action space $\mathcal{A}$, which can make learning optimal RL policies more challenging. Thus, it is important to find a balance and reduce the size of the action space whenever possible. This trade-off can be explored by selecting appropriate values for $N_{\mathcal{P}}$ (the number of available power levels) and $N_{\mathcal{W}}$ (the codebook size).

Figure 12 investigates the selection of a better combination of $(N_{\mathcal{P}}, N_{\mathcal{W}})$ under the constraint $|\mathcal{A}| = N_{\mathcal{P}}N_{\mathcal{W}} = 20$. By varying the values of $N_{\mathcal{P}}$ and $N_{\mathcal{W}}$, the impact on system performance can be examined to find the optimal combination.

From the results in Fig. 12, it is observed that for the SCA, increasing the codebook size $N_{\mathcal{W}}$ improves the average system sum rate compared to increasing the number of power levels. This improvement arises because SCA assigns frequency bands to co-channel UEs in a manner that ensures the serving distances of these co-channel UEs from their respective BSs becomes more similar. As a result, it is recommended to increase the codebook size to narrow the beamwidth, effectively reducing co-channel interference. On the other hand, with RCA, there is a significant variation in the distances between different co-channel UEs and their respective serving BSs. Consequently, under a restricted action space size, RCA faces a trade-off between maximizing the sum rate through adaptive power loading and minimizing co-channel interference through beam pattern selection. In Fig. 12, the trade-off in choosing $N_{\mathcal{P}}$ and $N_{\mathcal{W}}$ for RCA is depicted, where the combination of $(N_{\mathcal{P}}, N_{\mathcal{W}}) = (5, 4)$ yields higher sum rates compared to other combinations. Overall, the choice of $(N_{\mathcal{P}}, N_{\mathcal{W}})$ depends on the channel assignment method employed and the specific trade-offs between power loading and beam pattern selection to effectively mitigate co-channel interference and maximize the system sum rate.

VII. CONCLUSION

Recently, there has been growing interest in the use of distributed multi-agent RL models to tackle downlink multi-cell interference coordination. This approach has gained attention due to the inherent challenges in designing reward functions for centralized single-agent RL models that can capture the condition of the entire cellular network can be challenging. However, one of the primary hurdles encountered in multi-agent distributed learning is the instability arising from nonstationary environments.

To overcome this challenge, we proposed a novel single-agent RL model that leverages success examples to optimize transmit beamforming and power control simultaneously, without relying on explicit reward functions. Notably, our approach eliminates the need for information exchange among BSs, setting it apart from multi-agent RL models. Simulation results showed that our model outperforms both the theoretical FP algorithm and the multi-agent model. Furthermore, our proposed model not only maximizes the sum rate, but also ensures a more uniform signal quality level among the co-channel UEs. This is of significant importance as it enhances the overall user experience and mitigates the issue of uneven service quality among UEs that share the same frequency resources.

REFERENCES

[1] T. S. Rappaport, *Wireless Communications: Principles and Practice*, Upper Saddle River, NJ, USA: Prentice-Hall, 1996, pp. 25–63.

[2] J.-S. Sheu and K.-M. Huang, "Performance comparison for single-user and multi-user network MIMO cellular systems with power management," *Appl. Sci.*, vol. 11, no. 21, p. 10298, Nov. 2021, doi: 10.3390/APP112110298.

[3] J.-S. Sheu, S.-H. Lyu, and C.-Y. Huang, "On antenna orientation for inter-cell interference coordination in cellular network MIMO systems," *J. Commun. Netw.*, vol. 18, no. 4, pp. 639–648, Aug. 2016, doi: 10.1109/JCN.2016.000087.

[4] M. S. Ali, E. Hossain, and D. I. Kim, "LTE/LTE—A random access for massive machine-type communications in smart cities," *IEEE Commun. Mag.*, vol. 55, no. 1, pp. 76–83, Jan. 2017.

[5] C. Kosta, B. Hunt, A. U. Quddus, and R. Tafazolli, "On interference avoidance through inter-cell interference coordination (ICIC) based on OFDMA mobile systems," *IEEE Commun. Surveys Tuts.*, vol. 15, no. 3, pp. 973–995, 3rd Quart., 2013.

[6] B. Soret, A. D. Domenico, S. Bazzi, N. H. Mahmood, and K. I. Pedersen, "Interference coordination for 5G new radio," *IEEE Wireless Commun.*, vol. 25, no. 3, pp. 131–137, Jun. 2018.

[7] A. S. Hamza, S. S. Khalifa, H. S. Hamza, and K. A. Elsayed, "A survey on inter-cell interference coordination techniques in OFDMA-based cellular networks," *IEEE Commun. Surveys Tuts.*, vol. 15, no. 4, pp. 1642–1670, 4th Quart., 2013.

[8] S. Kumar, S. Kalyani, and K. Giridhar, "Impact of sub-band correlation on SFR and comparison of FFR and SFR," *IEEE Trans. Wireless Commun.*, vol. 15, no. 8, pp. 5156–5166, Aug. 2016.

[9] S.-E. Elayoubi, O. Ben Haddada, and B. Fourestie, "Performance evaluation of frequency planning schemes in OFDMA-based networks," *IEEE Trans. Wireless Commun.*, vol. 7, no. 5, pp. 1623–1633, May 2008.

[10] J. Ghosh and D. N. K. Jayakody, "An analytical view of ASE for multicell OFDMA networks based on frequency-reuse scheme," *IEEE Syst. J.*, vol. 14, no. 1, pp. 645–648, Mar. 2020.

[11] F. Rashid-Farrokhi, K. J. R. Liu, and L. Tassiulas, "Transmit beamforming and power control for cellular wireless systems," *IEEE J. Sel. Areas Commun.*, vol. 16, no. 8, pp. 1437–1450, Oct. 1998.

[12] F. Wang, X. Wang, and Y. Zhu, "Transmit beamforming for multiuser downlink with per-antenna power constraints," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Sydney, NSW, Australia, Jun. 2014, pp. 4692–4697.

[13] D. Gerlach and A. Paulraj, "Adaptive transmitting antenna arrays with feedback," *IEEE Signal Process. Lett.*, vol. 1, no. 10, pp. 150–152, Oct. 1994.

[14] D. Gerlach and A. Paulraj, "Spectral reuse using transmit antenna array and feedback," in *Proc. Int. Conf. Acoustic, Speech Signal Process.*, Adelaide, SA, Australia, Apr. 1994, pp. 97–100.

[15] I. M. Stancu-Minasian, *Fractional Programming: Theory, Methods and Applications*. Norwell, MA, USA: Kluwer, 1992.

[16] Q. Shi, M. Razaviyayn, Z.-Q. Luo, and C. He, "An iteratively weighted MMSE approach to distributed sum-utility maximization for a MIMO interfering broadcast channel," *IEEE Trans. Signal Process.*, vol. 59, no. 9, pp. 4331–4340, Sep. 2011.

[17] J. Ge, Y.-C. Liang, J. Joung, and S. Sun, "Deep reinforcement learning for distributed dynamic MISO downlink-beamforming coordination," *IEEE Trans. Commun.*, vol. 68, no. 10, pp. 6070–6085, Oct. 2020.

[18] P. Hernandez-Leal, M. Kaisers, T. Baarslag, and E. M. de Cote, "A survey of learning in multiagent environments: Dealing with non-stationarity," 2017, *arXiv:1707.09183*.

[19] G. Papoudakis, F. Christianos, A. Rahman, and S. V. Albrecht, "Dealing with non-stationarity in multi-agent deep reinforcement learning," 2019, *arXiv:1906.04737*.

[20] A. Marinescu, I. Dusparic, and S. Clarke, "Prediction-based multi-agent reinforcement learning in inherently non-stationary environments," *ACM Trans. Auto. Adapt. Syst.*, vol. 12, no. 2, pp. 1–23, May 2017.

[21] Q. Zhang, Y.-C. Liang, and H. V. Poor, "Intelligent user association for symbiotic radio networks using deep reinforcement learning," *IEEE Trans. Wireless Commun.*, vol. 19, no. 7, pp. 4535–4548, Jul. 2020.

[22] 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.

[23] I. Kostrikov, O. Nachum, and J. Tompson, "Imitation learning via off-policy distribution matching," 2019, *arXiv:1912.05032*.

[24] S. Reddy, A. D. Dragan, and S. Levine, "SQIL: Imitation learning via reinforcement learning with sparse rewards," 2020, *arXiv:1905.11108*.

[25] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.

[26] B. Eysenbach, R. Salakhutdinov, and S. Levine, "C-learning: Learning to achieve goals via recursive classification," 2020, *arXiv:2011.08909*.

[27] B. Eysenbach, S. Levine, and R. Salakhutdinov, "Replacing rewards with examples: Example-based policy search via recursive classification," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, Dec. 2021, pp. 11541–11552. [Online]. Available: https://proceedings.neurips.cc/paper/2021/hash/5ffaa9f5182c2a36843f438bb1fdbdea-Abstract.html

[28] R. Van Nee and R. Prasad, *OFDM for Wireless Multimedia Communications*. Norwood, MA, USA: Artech House, 2000.

[29] X. Shang, B. Chen, and H. V. Poor, "Multiuser MISO interference channels with single-user detection: Optimality of beamforming and the achievable rate region," *IEEE Trans. Inf. Theory*, vol. 57, no. 7, pp. 4255–4273, Jul. 2011.

[30] H. V. Poor and S. Verdu, "Single-user detectors for multiuser channels," *IEEE Trans. Commun.*, vol. COM-36, no. 1, pp. 50–60, Jan. 1988.

[31] C. C. Tan and N. C. Beaulieu, "First-order Markov modeling for the Rayleigh fading channel," in *Proc. IEEE GLOBECOM*, 1998, pp. 3669–3674.

[32] C. Elkan and K. Noto, "Learning classifiers from only positive and unlabeled data," in *Proc. 14th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2008, pp. 213–220.

[33] V. R. Konda and J. N. Tsitsiklis, "Actor-critic algorithms," in *Advances in Neural Information Processing Systems*, vol. 12, S. A. Solla, T. K. Leen, and K.-R. Müller, Eds. Cambridge, MA, USA: MIT Press, 2000, pp. 1008–1014.

[34] T. David and V. Pramod, *Fundamentals of Wireless Communication*. Cambridge, U.K.: Cambridge Univ. Press, 2005.

[35] R. S. Sutton, "Learning to predict by the methods of temporal differences," *Mach. Learn.*, vol. 3, no. 1, pp. 9–44, Aug. 1988.

JENG-SHIN SHEU received the Ph.D. degree in electrical engineering from National Chung Cheng University, Taiwan, in 2002. From 2002 to 2006, he was a Postdoctoral Researcher with National Chiao Tung University, Taiwan. He is currently an Associate Professor with the Department of Computer Science and Information Engineering, National Yunlin University of Science and Technology. His research interests include cellular mobile systems and audio and speech processing. He was a recipient of the 2019 Premium Awards for Best Paper in *IET Optoelectronics*.

CHENG-KUEI HUANG received the bachelor's degree from the Department of Computer Science and Information Engineering, Lunghwa University of Science and Technology, in 2017, and the master's degree from the National Yunlin University of Science and Technology, Taiwan, in 2022. His research interests include mobile communication theory and natural language processing.

CHUN-LUNG TSAI received the bachelor's degree from the Department of Information and Communication Engineering, Chaoyang University of Science and Technology, Taiwan, in 2019, and the master's degree in computer science and information engineering from the National Yunlin University of Science and Technology, in 2023. His research interests include wireless communication systems and reinforcement learning.