Offline Reinforcement Learning for Wireless Network Optimization with Mixture Datasets

Kun Yang*, Cong Shen*, Jing Yang†, Shu-ping Yeh‡, Jerry Sydir‡
* Department of Electrical and Computer Engineering, University of Virginia, USA
† Department of Electrical Engineering, The Pennsylvania State University, USA
‡ Intel Corporation, USA

***Abstract*—The recent development of reinforcement learning (RL) has boosted the adoption of online RL for wireless radio resource management (RRM). However, online RL algorithms require direct interactions with the environment, which may be undesirable given the potential performance loss due to the unavoidable exploration in RL. In this work, we first investigate the use of *offline* RL algorithms in solving the RRM problem. We evaluate several state-of-the-art offline RL algorithms, including behavior constrained Q-learning (BCQ), conservative Q-learning (CQL), and implicit Q-learning (IQL), for a specific RRM problem that aims at maximizing a linear combination of sum and 5-percentile rates via user scheduling. We observe that the performance of offline RL for the RRM problem depends critically on the behavior policy used for data collection, and further propose a novel offline RL solution that leverages heterogeneous datasets collected by different behavior policies. We show that with a proper mixture of the datasets, offline RL can produce a near-optimal RL policy even when all involved behavior policies are highly suboptimal.**

***Index Terms*—Radio Resource Management, Offline Reinforcement Learning, Deep Reinforcement Learning.**

I. Introduction

There is a growing interest in applying reinforcement learning (RL) to solving radio resource management (RRM) problems in wireless networks. Several unique properties in wireless RRM are the driving force behind this new trend. First, many of the RRM operations are sequential in nature, where a resource allocation decision is made, the network performance is observed, and then fed back to the decision maker to update the policy. Second, real-world wireless network optimization problems are often too complex to be modeled as simple optimization problems, which calls for *model-free* solutions that can be adaptive to the unknown deployment. Third, there are well-established control and feedback mechanisms in modern wireless networks, making it easy to observe system states and collecting performance indicators.

These features have sparked significant efforts in developing RL solutions for wireless RRM. An overview of related works is given in Section II. Majority, if not all, of the existing works utilize *online RL*, where the RL policy gradually improves by *interacting with the environment* with no data prior to deployment. The exploration of the originally unknown environment, especially during the early stages where information about the environment is scarce and RL exploration is almost random, is an indispensable component for online RL, but is also one of the major obstacles that prevent state-of-the-art RL algorithms from being deployed in real-world wireless networks. The lack of performance guarantee during RL exploration means that the network users may have to temporarily suffer from poor Quality of Service (QoS) so that the learning agent can gather information about the deployment for a potentially better RL policy. This tradeoff, however, is undesirable for the wireless network operator compared with model-based or rule-based solutions, which may not achieve as good a performance as online RL after it converges, but does not suffer from potentially significant initial performance degradation.

In this paper, we advocate to adopt *offline reinforcement learning* [1] for wireless network optimization. Offline RL aims at training RL agents using accessible datasets collected *a priori* and thus completely gets around online interactions. This paradigm is particularly suitable for wireless RRM, because in practice wireless operators already have deployed some policy that controls resource allocation, and there are mature mechanisms to collect the operational data.

We study the feasibility and performance of offline RL for wireless RRM by evaluating state-of-the-art offline RL algorithms, including behavior constrained Q-learning (BCQ) [2], conservative Q-learning (CQL) [3], and implicit Q-learning (IQL) [4], for a wireless user scheduling problem that aims at maximizing a linear combination of sum and 5-percentile rates. The potential of offline RL is demonstrated via extensive system simulations, and we observe that the performance of offline RL for the user scheduling problem depends critically on the behavior policy used for data collection – dataset collected from a bad behavior policy does not lead to a good RL policy. Towards solving this problem, we propose a novel offline RL solution that leverages heterogeneous datasets collected by different behavior policies. Somewhat surprisingly, mixing datasets collected by different behavior policies allows offline RL to produce a near-optimal policy, even when all involved behavior policies are highly suboptimal. We further discuss the potential reasons for mixture dataset to benefit offline RL, and present possible future research directions.

The rest of the paper is organized as follows. Related works are surveyed in Section II. The wireless network model and figure of merit are presented in Section III. The Markov

The work of KY and CS is partially support by the US National Science Foundation under awards CNS-2002902, ECCS-2029978, and SII-2132700. The work of JY is supported in part by US NSF under awards CNS-2003131, ECCS-2030026, and ECCS-2143559. This is the camera-ready version of the paper accepted at Asilomar 2023

Decision Process formulation of the user scheduling problem and the online RL solution are discussed in Section IV. Section V presents the basic framework of offline RL for wireless user scheduling, and reports the initial experiment results. The new solution of offline RL with mixture datasets is presented in Section VI, together with the experimental results. Finally, Section VII concludes the paper.

II. Related Works

Online RL for RRM. Existing literature on wireless Radio Resource Management (RRM) primarily utilizes online RL methods. For instance, deep Q-networks have been applied to power allocation problems in a centralized setting [5], [6], and single-agent deep RL has been used for joint power and channel allocation [7]. A comprehensive comparison of various online RL algorithms in wireless network optimization is detailed in [8]. Additionally, multi-agent reinforcement learning (MARL) has been implemented for power allocation [9] and broader resource management and interference mitigation tasks [10].

Offline RL. Unlike the online RL algorithms, offline RL focuses on learning RL policies exclusively from offline datasets, and has attracted significant interest in RL research [1]. Because offline RL cannot update policy by interacting with the environment, most methods choose to be conservative to mitigate potential distributional shift. Among the algorithms, batch-constrained Q-learning (BCQ) [2], conservative Q-learning (CQL) [3] and implicit Q-learning (IQL) [4] are the most state-of-the-art model-free deep offline RL algorithms. We will adopt these algorithms in our paper. Theoretical understanding towards optimal offline RL is also an active research direction, where data coverage [11], [12] and critical states [13] have been investigated.

III. System Model

In this section, we present the wireless environment and then discuss the figure of merit for the RRM problem.

A. Wireless Environment

We study a wireless network with N access points (APs) and M user equipments (UEs). Time is slotted and each episode consistes of T time slots. APs are randomly placed and stationary throughout the Radio Resource Management (RRM) process, ensuring diverse scenario coverage. UEs are also randomly positioned at the start of each episode within a defined area, adhering to minimum distance constraints from APs and other UEs. The primary focus is on an indoor environment where UEs move at a slow pace, with a maximum speed of 1 m/s, randomly changing positions in each time slot. A mirror-back mechanism is employed for UEs reaching coverage boundaries or violating distance constraints.

UEs are associated with one of the APs at the beginning of each episode, and we assure that every UE would be associated with one and only one AP. At each time slot t, the channel between AP i and UE j, denoted as $h_{i,j}(t)$, follows the standard 3GPP indoor path-loss model [14], log-normal shadowing and short-term standard frequency-flat Rayleigh fading.

The task of our studied RRM problem is *user scheduling*, i.e., to determine which BS to serve which UE (or to turn off without serving any UE) for each time slot t. In reality, user association happens at a much slower time scale than user scheduling. Thus, we first perform user association at the beginning of each episode, and keep this association unchanged throughout the current episode. User scheduling then happens on a per-time-slot basis.

Under this setting, we model the instantaneous data rate for user j using Shannon capacity $C_j(t) = \log_2(1 + \mathsf{SINR}_j(t))$, where $\mathsf{SINR}_j(t)$ denotes the signal to interference plus noise ratio (SINR) of user j at time slot t. We further define the *average user throughput* for user j as $\bar{C}_j = \frac{1}{T}\sum_{t=1}^{T} C_j(t)$.

User association rule. At the beginning of each episode, a user pool $\mathbb{P}_i$ is created for each AP i based on the maximum reference signal received power (RSRP) of each user. More specifically, user j will be added to the user pool of AP i if $i = \arg\max_n \mathsf{RSRP}_{n,j}, \forall n \in \{1, \cdots, N\}$. An AP is allowed to only observe and measure users in its own user pool, and scheduling decisions are limited to these users.

B. Figure of Merit

If the figure of merit for wireless RRM is to maximize the system-level averaged data rate (across all users), then the solution boils down to always selecting the "best" UEs (in terms of the SINR) at each time step. This can be formulated as an optimization problem for each time slot, and there are extensive works studying different variants of this problem. However, almost all practical wireless networks must consider *fairness* across all UEs when solving the RRM problem. From the data rate perspective, the overall system figure of merit must consider both the *sum* and *tail* behaviors. This is often captured by the **sum rate** $C_{\text{sum}} = \sum_{j=1}^{M} \bar{C}_j$, and the **5-percentile rate** $C_{5\%}$, which is the solution of the following problem:

$$\begin{aligned} &\text{maximize} && C \\ &\text{subject to} && P[\bar{C}_j > C] \geq 0.95, \forall j \in \{1, \cdots, M\}. \end{aligned}$$

The main figure of merit of this work is a linear combination of the sum rate and the 5-percentile rate, parameterized by (μ, η) as

$$R_{\text{score}} = \mu C_{\text{sum}} + \eta C_{5\%}. \quad (1)$$

We note that this weighted sum allows us to adjust the balance between sum and tail rates, by varying the parameters μ and η. However, directly maximizing R_{score} is a non-trivial task that faces several challenges. Both the sum and 5-percentile rates are *long-term* performance measures that depend on the history of actions in an episode. The time-dependency of actions implies that we cannot take the optimization-per-slot approach to find a (near-)optimal solution. Additionally, the 5-percentile rate itself is a complicated measure that does not have a close-form expression, and the dynamic nature of the system (channel randomness, user movement, etc) further adds to the difficulty of optimizing R_{score}.

Another related metric is *proportional fairness (PF)*. We define the PF ratio for user j as, $\mathsf{PF}_j(t) = w_j(t)C_j(t)$ where

$$w_j(t) = 1/\tilde{C}_j(t), \quad \tilde{C}_j(t) = \alpha C_j(t) + (1-\alpha)\tilde{C}_j(t-1)$$

and $\tilde{C}_j(0) = C_j(0)$. The PF ratio does not directly translate to the sum and tail rates, but will be used in the policies to limit the action space in each time slot. We note that other metrics such as delay [15], [16] can also be incorporated in the formulation.

IV. RL FORMULATION

In this section, we show how to solve the wireless RRM problem using RL. This is accomplished by first formulating the original system as a Markov Decision Process (MDP), and then discussing how to train a centralized *online* RL to control all the APs in the environment.

A. MDP Formulation

An episodic MDP is described by a tuple $M = (S, A, r, \gamma, P, T)$, where S and A stand for the state and action spaces respectively, r is the reward function mapping a state-action pair to a reward signal that reflects our design objective, P is the transition kernel advancing the current state-action pair to the next state in a random fashion, and T is the length of the episode. We define the key components of the episodic MDP for the wireless scheduling problem as follows.

Observation. For each AP i, we apply a top-k selection of the UEs in its user pool to collect observations. The criterion of top-k UEs is by sorting all UEs in the pool based on the PF-ratio $w_{i,k}(t)$ and only keeping the largest k UEs. We note that this is a common technique in the existing literature to deal with large amount of UEs [9], [10]. Then, with the top-k UEs, the AP measures the current SINR for each UE, and the local observation at AP i is defined as $o_i(t) = (\mathsf{SINR}_{i,1}(t), w_{i,1}(t), \cdots, \mathsf{SINR}_{i,k}(t), w_{i,k}(t))$. Finally, with all local observations, the learning agent creates the global observation by stacking the local ones as $O(t) = (o_1(t), \cdots, o_N(t))$.

Action. For each AP, the possible actions are to either select one from its top-k users to serve, or to turn itself off and serve no UE. The action space for each AP thus has size $k+1$, and the global action space is of size $(k+1)^N$.

Reward. The objective R_{score} defined in Eqn. (1) represents the final performance and cannot be directly decomposed into reward signals for each step. We thus adopt an existing design from [10] that has been shown to achieve a balanced tradeoff between sum and tail rates $r(t) = \sum_{j=1}^{M}(w_j(t))^\lambda C_j(t)$. By tuning the parameter λ, we can achieve the desired tradeoff between sum rate and 5% rate as detailed in [10].

B. Online RL

We present an online reinforcement learning (RL) baseline for the Radio Resource Management (RRM) problem using the Soft Actor-Critic (SAC) algorithm [17], known for its stability and efficient policy optimization. SAC, an off-policy algorithm, stands out for its faster convergence and employs policy iteration steps in the actor-critic structure to balance policy update control and exploration.

Our system-level simulator, aligned with the wireless environment specified in Section III-A, facilitates the online RL training. In this phase, we interact with a randomly generated environment over 5000 episodes, each containing 200 time steps, across a maximum of 350 training epochs. Policy evaluations are conducted in 10 different environments at the end of each epoch, with results averaged over 20 distinct training setups.

Other than the online RL policy trained via SAC, we also evaluate several rule-based baseline methods as follows. We note that these baseline methods will be used as behavior policies in the subsequent offline RL study.

- **Random.** At each time step, each AP randomly chooses one of its top-k users in the user pool to serve.
- **Greedy.** AP always chooses one of its top-k users with the largest SINR to serve.
- **Time division multiplexing (TDM).** All top-k UEs are served in a round-robin fashion. In each time slot, only the scheduled UE and its serving AP are active.
- **ITLinQ. [18]** It is a generalized independent set-based scheduling algorithm where we select UEs with large PF ratios while creating small interference.

In our experiment, we set the number of APs to 4 and the number of UEs ranging from 10 to 24. The results in Fig. 1 indicate that the performance of online reinforcement learning significantly surpasses that of all rule-based baselines with sufficient training. However, it is noteworthy that with limited training, the performance of online RL can be worse than rule-based baselines. This highlights a critical trade-off in online learning: the necessity of exploration in reinforcement learning can initially hinder performance when training is inadequate. Offline RL, on the other hand, works completely on the offline dataset and does not suffer from these explorations.

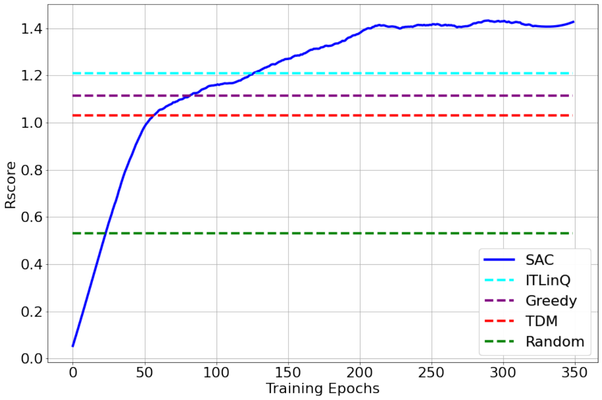

Fig. 1. Training the online SAC agent. Results are validated over 10 new environments and averaged across 10 independent runs.

V. OFFLINE RL

To address the limitations of online RL, we resort to *offline* RL for RRM. We first describe how to collect the offline dataset, and then evaluate the state-of-the-art offline RL algorithms for the same user scheduling problem as described before.

A. Dataset

Offline RL allows the system to enjoy the advantages of RL without direct interactions with the environment. This is made possible by using offline datasets. The most common approach to have such datasets is through collecting operational data associated with the existing policies. For example, for the wireless RRM problem, wireless operators often have existing solutions that have been deployed in the target environment. We can rely on the data collected by these existing solutions, which are called the *behavior policies (BPs)*, to train an offline RL policy.

In the user scheduling problem, we have deployed four rule-based policies described in Section IV-B as BPs. In addition, we also include two other BPs that are based on online RL. These two policies differ in how well they are trained – one is early stopped at epoch 125 while the other is stopped at epoch 350. Their performances can be identified from the yellow dash line and black dash line in Fig. 3.

With datasets collected from these BPs, we summarize the offline RL experiment procedure as follows.

1) Choose a BP π_β from all available BPs.
2) Run π_β on the environment to collect a dataset D_{π_β}.
3) Train policy π_θ using an offline RL algorithm (see next subsection) on the dataset D_{π_β}.

We remark that the dataset collected in Step 2) may have poor quality because the corresponding BP may not achieve good performance. For example, as we see from Fig. 3, the four rule-based policies all have significant performance gaps compared with the well-trained online RL. We are interested in evaluating whether a "good" RL policy can be trained from datasets that may come from "bad" BPs.

B. Offline RL for User Scheduling

In this study, we focus on the application of three leading model-free offline reinforcement learning algorithms in addressing the user scheduling problem in wireless communications: Behavior Constrained Q-learning (BCQ) [2], Conservative Q-learning (CQL) [3], and Implicit Q-learning (IQL) [4]. Given the limited space of this paper, an exhaustive exploration of these algorithms is beyond our current purview. We encourage readers interested in a more detailed analysis to refer to the respective original papers for in-depth insights. However, we would like to point out that all three methods use principle of *conservative exploration* [19] to address the distributional shift issue but only differs on the methodologies instead of neural network architectures.

High-quality dataset. Utilizing the best online RL policy as BP, we gathered one million trajectories for our offline dataset. In offline RL, each epoch consists of 10,000 mini-batch updates with 64 data samples per batch. The results in Fig. 2, presented alongside the original online RL performance for comparison, show that offline RL algorithms converge faster than online RL without a need for exploration.

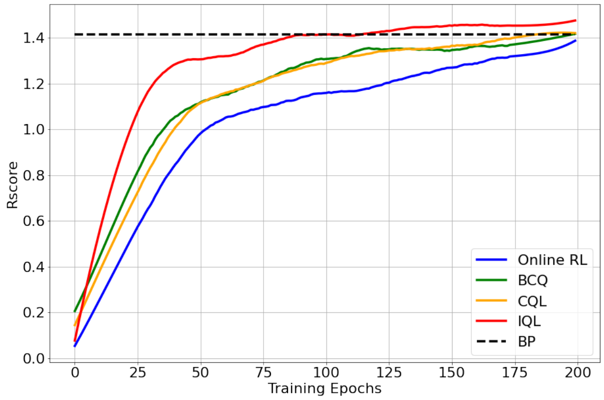

Fig. 2. Comparing different SOTA offline RL algorithms with an expert dataset (generated from the best online RL policy). Policies are evaluated using the same 10 validation environments as the online RL experiments.

Low-quality dataset. When investigating the effectiveness of offline RL with lower quality datasets, we focused only on IQL as it is the best-performing algorithm in our previous tests. The results in Fig. 3 suggest that the quality of the original BP limits the performance of the offline RL policy – the performance gain of IQL over the BP is very modest for all policies. This finding has motivated us to propose a method of combining several low-quality datasets to enhance offline RL performance, detailed in Section VI.

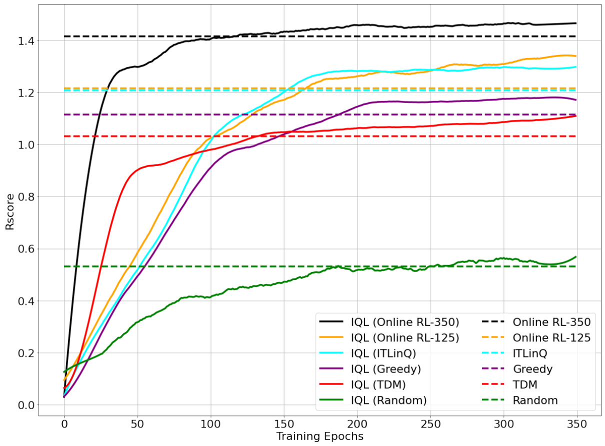

Fig. 3. Testing offline RL with datasets generated from different behavior policies. Validation is on the same 10 validation environments.

VI. Offline RL with Mixture Datasets

A. Mixture Datasets from Multiple BPs

The previous results demonstrate the potential of offline RL for the user scheduling problem in wireless RRM, but also suggest that the gain can be limited by the adopted behavior policy used to collect the dataset. The key open problem is the following.

Can we train a high-performance offline RL policy using datasets from low-performance behavior policies?

We answer this question positively by developing a novel offline RL solution. Our key new idea is that although the dataset generated by a *single* low-performance BP may not contain enough information to learn a near-optimal RL policy, the cumulative dataset from multiple low-performance BPs may have sufficient diversity to cover the (near-)optimal state-action pairs, although each BP may cover only a portion of them.

We evaluate this idea using the same experimental setting as in Section V, and report the results in Fig. 4. We consider two different combinations of BPs, both with $L = 4$:

1) **Mixed-RL:** "bad" online RL (trained with 125 epochs), Greedy, TDM, and Random.
2) **Mixed-ITLinQ:** ITLinQ, Greedy, TDM, and Random.

To have a consistent comparison, we adopt the same data allocation[1] for both cases: online RL or ITLinQ (50%), Greedy (20%), TDM (20%), and Random (10%). The results are presented in Fig. 4. We can see that although the data samples are coming from low-performance BPs, leveraging the mixture dataset leads to significant improvement for offline RL. In fact, Mixed-RL can almost converge to the optimal online RL performance as shown in Fig. 4, and Mixed-ITLinQ is only slightly worse. Both outperform all of the individual BPs by noticeable margins. These results demonstrate that even with several low-performance BPs, we can still leverage the offline datasets to achieve near-optimal RL performance.

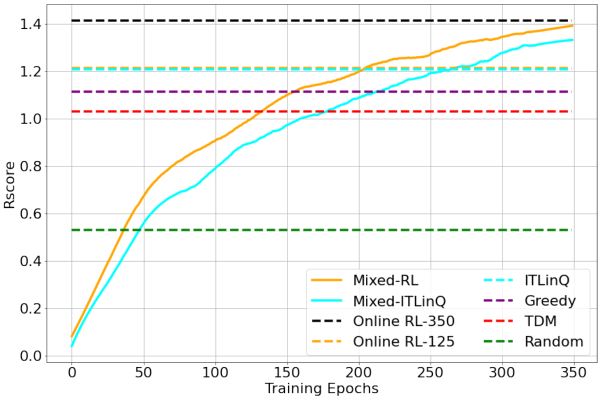

Fig. 4. Testing offline RL with a mixture dataset. Results are validated across the same 10 validation environments as before.

VII. Conclusion

We have introduced offline RL as a promising RL solution for the wireless RRM problem. Toward this end, we used a specific wireless user scheduling problem as an example, and evaluated several state-of-the-art offline RL algorithms in terms of their long-term performance and convergence rate. We observed that the performance of offline RL is largely constrained by the quality of the behavior policy that is adopted to collect the dataset, and further proposed a novel offline RL solution that leverages heterogeneous datasets collected by various behavior policies. We showed that this solution can produce a near-optimal RL policy even when all involved behavior policies are highly suboptimal, and provided some possible explanations. Our work adopted a centralized offline RL framework to produce the RRM policy, which does not scale well. It would be interesting to develop a multi-agent offline RL solution for this problem.

[1] Note that online RL-125 and ITLinQ have very similar performances; hence their allocations are set to be the same.

References

[1] S. Levine, A. Kumar, G. Tucker, and J. Fu, "Offline reinforcement learning: Tutorial, review, and perspectives on open problems," *arXiv preprint arXiv:2005.01643*, 2020.

[2] S. Fujimoto, D. Meger, and D. Precup, "Off-policy deep reinforcement learning without exploration," in *International Conference on Machine Learning*. PMLR, 2019, pp. 2052–2062.

[3] A. Kumar, A. Zhou, G. Tucker, and S. Levine, "Conservative Q-learning for offline reinforcement learning," *Advances in Neural Information Processing Systems*, vol. 33, pp. 1179–1191, 2020.

[4] I. Kostrikov, A. Nair, and S. Levine, "Offline reinforcement learning with implicit Q-learning," *arXiv preprint arXiv:2110.06169*, 2021.

[5] K. I. Ahmed and E. Hossain, "A deep Q-learning method for downlink power allocation in multi-cell networks," *arXiv preprint arXiv:1904.13032*, 2019.

[6] F. Meng, P. Chen, and L. Wu, "Power allocation in multi-user cellular networks with deep Q learning approach," in *IEEE International Conference on Communications (ICC)*. IEEE, 2019, pp. 1–6.

[7] G. Zhao, Y. Li, C. Xu, Z. Han, Y. Xing, and S. Yu, "Joint power control and channel allocation for interference mitigation based on reinforcement learning," *IEEE Access*, vol. 7, pp. 177 254–177 265, 2019.

[8] K. Yang, C. Shen, and T. Liu, "Deep reinforcement learning based wireless network optimization: A comparative study," in *IEEE INFOCOM Workshop on Data Driven Intelligence for Networks*, Toronto, Canada, Jul. 2020, pp. 1248–1253.

[9] Y. S. Nasir and D. Guo, "Multi-agent deep reinforcement learning for dynamic power allocation in wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 10, pp. 2239–2250, 2019.

[10] N. Naderializadeh, J. J. Sydir, M. Simsek, and H. Nikopour, "Resource management in wireless networks via multi-agent deep reinforcement learning," *IEEE Transactions on Wireless Communications*, vol. 20, no. 6, pp. 3507–3523, 2021.

[11] P. Rashidinejad, B. Zhu, C. Ma, J. Jiao, and S. Russell, "Bridging offline reinforcement learning and imitation learning: A tale of pessimism," *Advances in Neural Information Processing Systems*, vol. 34, pp. 11 702–11 716, 2021.

[12] T. Xie, N. Jiang, H. Wang, C. Xiong, and Y. Bai, "Policy finetuning: Bridging sample-efficient offline and online reinforcement learning," *Advances in Neural Information Processing Systems*, vol. 34, pp. 27 395–27 407, 2021.

[13] A. Kumar, J. Hong, A. Singh, and S. Levine, "When should we prefer offline reinforcement learning over behavioral cloning?" *arXiv preprint arXiv:2204.05618*, 2022.

[14] 3GPP, "Simulation assumptions and parameters for FDD HeNB RF requirements," Tech. Rep. R4-092042.

[15] N. Mastronarde and M. van der Schaar, "Joint physical-layer and system-level power management for delay-sensitive wireless communications," *IEEE Transactions on Mobile Computing*, vol. 12, no. 4, pp. 694–709, 2012.

[16] Z. Lin and M. van der Schaar, "Autonomic and distributed joint routing and power control for delay-sensitive applications in multi-hop wireless networks," *IEEE Transactions on Wireless Communications*, vol. 10, no. 1, pp. 102–113, 2010.

[17] T. Haarnoja, A. Zhou, K. Hartikainen, G. Tucker, S. Ha, J. Tan, V. Kumar, H. Zhu, A. Gupta, P. Abbeel *et al.*, "Soft actor-critic algorithms and applications," *arXiv preprint arXiv:1812.05905*, 2018.

[18] N. Naderializadeh and A. S. Avestimehr, "ITLinQ: A new approach for spectrum sharing in device-to-device communication systems," *IEEE Journal on Selected Areas in Communications*, vol. 32, no. 6, pp. 1139–1151, 2014.

[19] D. Li, R. Huang, C. Shen, and J. Yang, "Near-optimal conservative exploration in reinforcement learning under episode-wise constraints," in *International Conference on Machine Learning*, Jul. 2023.