sensors

MDPI

Article

Joint Beamforming, Power Allocation, and Splitting Control for SWIPT-Enabled IoT Networks with Deep Reinforcement Learning and Game Theory

JainShing Liu [1], Chun-Hung Richard Lin [2,*], Yu-Chen Hu [3] and Praveen Kumar Donta [4]

[1] Department of Computer Science and Information Engineering, Providence University, Taichung 43301, Taiwan; chhliu@pu.edu.tw
[2] Department of Computer Science and Engineering, National Sun Yat-sen University, Kaohsiung 80424, Taiwan
[3] Department of Computer Science and Information Management, Providence University, Taichung 43301, Taiwan; ychu@pu.edu.tw
[4] Research Unit of Distributed Systems, TU Wien, 1040 Vienna, Austria; pdonta@dsg.tuwien.ac.at
* Correspondence: lin@cse.nsysu.edu.tw

Abstract: Future wireless networks promise immense increases on data rate and energy efficiency while overcoming the difficulties of charging the wireless stations or devices in the Internet of Things (IoT) with the capability of simultaneous wireless information and power transfer (SWIPT). For such networks, jointly optimizing beamforming, power control, and energy harvesting to enhance the communication performance from the base stations (BSs) (or access points (APs)) to the mobile nodes (MNs) served would be a real challenge. In this work, we formulate the joint optimization as a mixed integer nonlinear programming (MINLP) problem, which can be also realized as a complex multiple resource allocation (MRA) optimization problem subject to different allocation constraints. By means of deep reinforcement learning to estimate future rewards of actions based on the reported information from the users served by the networks, we introduce single-layer MRA algorithms based on deep Q-learning (DQN) and deep deterministic policy gradient (DDPG), respectively, as the basis for the downlink wireless transmissions. Moreover, by incorporating the capability of data-driven DQN technique and the strength of noncooperative game theory model, we propose a two-layer iterative approach to resolve the NP-hard MRA problem, which can further improve the communication performance in terms of data rate, energy harvesting, and power consumption. For the two-layer approach, we also introduce a pricing strategy for BSs or APs to determine their power costs on the basis of social utility maximization to control the transmit power. Finally, with the simulated environment based on realistic wireless networks, our numerical results show that the two-layer MRA algorithm proposed can achieve up to 2.3 times higher value than the single-layer counterparts which represent the data-driven deep reinforcement learning-based algorithms extended to resolve the problem, in terms of the utilities designed to reflect the trade-off among the performance metrics considered.

Keywords: joint optimization; deep reinforcement learning; game theory; multi-resource allocation; beamforming; power control; energy harvesting; IoT

Citation: Liu, J.; Lin, C.-H.R.; Hu, Y.-C.; Donta, P.K. Joint Beamforming, Power Allocation, and Splitting Control for SWIPT-Enabled IoT Networks with Deep Reinforcement Learning and Game Theory. *Sensors* **2022**, *22*, 2328. https://doi.org/10.3390/s22062328

Academic Editor: Rebeca P. Díaz Redondo

Received: 28 January 2022
Accepted: 15 March 2022
Published: 17 March 2022

1. Introduction

The tremendous growth in wireless data transmission would be a result from the introduction of fifth generation of wireless communications (5G) and will continue in the wireless networks beyond 5G (B5G). In particular, the collaboration between 5G enabled Internet of Things (5G-IoT) and wireless sensor networks (WSNs) will extend the connections between the Internet and the real world and widen the scope of IoT services. In such collective networks, by uploading part of or all of the computing tasks to the edge computing, a mobile edge computing (MEC) technique is developed to reduce the enormous data traffic and huge energy consumption brought by a great number of IoT devices

and sensors [1,2]. Even given that, realizing 5G or B5G IoT networks is still challenging due to the limited energies for the IoT devices equipped with batteries. To alleviate this problem, simultaneous wireless information and power transfer (SWIPT) are proposed to effectively and conveniently extend the lifetime of IoT devices, and employed in many related works [3–7]. In fact, SWIPT is a key technique in 5G and B5G because power allocation and interference management are still the crucial issues to be addressed in the communication networks [8,9]. In the border ground, the techniques of power control along with beamforming and interference coordination are usually adopted to increase the signal for data transmissions and improve the data rates received by end-users. However, these techniques by default treat the interference as a harmful impact to data transmissions, and ignore its potential to increase the communication capacity. By contrast, SWIPT opens up the potential by harvesting energy from the ambient electromagnetic sources including the interference signals. Consequently, not only would the benefits be obtained in which devices with SWIPT can transfer the interference into a useful resource, but also there is an advantage that can be taken with the signal-to-noise and interference ratio (SINR) to be increased by SWIPT for the residual energy of IoT devices.

In this work for the scenario that multiple BSs or APs can simultaneously transmit data and energy to their mobile nodes (MNs) in edge, we further show that, when the power control and interference management meet SWIPT, an overall system utility reflecting data rate, energy harvesting, and power consumption at the same time can be conduced to lead the system to an optimal trade-off on these performance metrics. Given that, how to allocate the transmit power, select the beamforming vector, and decide the power splitting ratio for the system will be a complex multiple resource allocation (MRA) problem, and can be formulated as a mixed integer nonlinear programming (MINLP) problem or even a non-convex MINLP problem. In general, MINLP problems are NP-hard and no efficient global optimal algorithm is available. Thus, apart from traditional optimization programming programs [10–17], research efforts usually resort to game theory [18–21], graph theory [22,23], and heuristic algorithms [24,25] to reduce the complexity.

More recently, inspired by the success of deep reinforcement learning (DRL) [26] on the application of computer science in various important fields, using DRL to solve the network problems, such as power control [27–29], joint resource allocation [30,31], and energy harvesting [32], becomes one of the main trends in the communication society. Although DRL is a useful tool to resolve these problems, the data-driven approaches that resulted usually treat a given resource optimization problem as a black box to learn its input/output relationship via various DRL techniques, which do not explicitly take the advantages from the model-based counterparts, such as game theory, graph theory, and heuristic algorithms mentioned previously. By noticing this fact, in this work, we first show how to design DRL-based approaches operated in a single layer to (1) jointly solve for power control, beamforming selection, and power splitting decision, and (2) approach the optimal trade-off among the performance metrics without exhaustive search in the action space. Next, we show how to incorporate a data-driven DRL-based technique and a model-driven game-theory-based algorithm to form a two-layer iterative approach to resolve the NP-hard MRA problem. By taking benefits from both data-driven and model-driven methods, the proposed two-layer MRA approach is shown to outperform the single-layer counterparts which rely only on the data-driven DRL-based algorithms.

1.1. Related Work

As a related work for LTE, the almost blank subframe (ABS) method was proposed in the standard [33] to resolve the co-channel inter-cell interference problem caused by two LTE base stations interfering with each other. Although ABS works well in fixed beam patterns, it was shown in [34] that ABS would be inefficient due to the dynamic nature of beamforming. Apart from the standard's solution, particular attention has also been paid to the efforts on resolving different resource allocation (RA) problems. In this work, these efforts would be classified into two categories, namely *model-driven methods* and *data-driven*

methods. According to our subjects, the former includes optimization methods and game theory methods while the latter simply denotes machine learning methods. As expected, a lot of previous works would be classified into the former, including graph theory [35,36], optimization decomposition [10,11,13–15,17], and dual Lagrangian method [12,16], in addition to game theory.

As a kind of data-driven method in the latter, which requires no model-oriented analysis and design, DRL would play a key role in solving RA problems. For example, the work in [37] proposed an inter-cell interference coordination and cell range expansion technique in heterogeneous networks, wherein dynamic Q-learning-based methods were introduced to improve user throughput. In addition, the previous works [29,38,39] introduced different deep Q-learning-based power control methods to maximize their objectives. Apart from Q-learning, in [40,41], actor–critic reinforcement learning (ACRL) algorithms were developed to reduce energy consumption. Recently, with deep deterministic policy gradient (DDPG), an algorithm was proposed in [32] that can be applicable for continuous states to realize continuous energy management, getting rid of the curse of dimensionality due to discrete action space from Q-learning.

Apart from the above, game-theory-based methods also received a lot of attention. For example, non-cooperative interference-aware RA has been proposed in [19] to improve the resource utilization efficiency of OFDMA networks. In [42], an interference coordination game was introduced, and the Nash equilibrium was found to reduce its computational complexity. Similarly, a joint transmit power and subchannel allocation problem was considered in [20], and a distributed non-cooperative game-based RA algorithm and a linear pricing technique were introduced therein to find the solutions. In addition, a power control problem for self-organizing small cell networks was formulated as a non-cooperative game in [21], which can then be solved by using the distributed energy efficient power control scheme proposed. Recently, by introducing a time-varying interference pricing with SWIPT, the authors in [18] modeled the power allocation problem as a non-cooperative game, and, by minimizing the total interferences experienced, they modeled the subchannel allocation problem as a non-cooperative potential game. Then, they proposed iterative algorithms to obtain the Nash equilibrium points corresponding to these games for the solutions.

More recently, there are different learning-based approaches proposed to resolve various problems in IoT networks. For example, a beamforming design for SWIPT-enabled networks was introduced in [43], where the rate-splitting scheme and the power-splitting energy harvesting receiver are adopted for secure information transfer and energy harvesting, respectively. This work formulates an energy efficiency (EE) maximization problem and properly addresses the beamforming design issue. However, such an issue is not our focus. In [44], an EE maximization problem is considered for the SWIPT enabled heterogeneous networks (HetNets). To resolve this problem, the authors introduced a min-max probability machine and an interactive power allocation/splitting scheme based on convex optimization methods. In the latter, the Lagrange multipliers for the optimization problem involved are obtained by using the subgradient method, which could be time-consuming to converge. Despite the different design aim, our work instead develops a game-based interactive method additionally controlled by a threshold to meet our time constraint. In [45], a sum rate maximization problem was formulated for SWIPT enabled HetNets, which jointly optimizes transmit beamforming vectors and power splitting ratios. With the multi-agent DDPG method for the user equipment (UE) without mobility, this work exhibits a notable performance gain when compared with the fixed beamforming design. When UE is mobile and not in the same location vicinity, the wireless channel is not constant and varies with UE's location. Taking this into account, the work in [46] resolved the dynamic problem with a multi-agent formulation to learn its optimization policy. Specifically, the authors resorted to the majorization–minimization (MM) technique and Dinkelbach algorithm to find the locally optimal solution using the convex optimization method for solving the power and time allocation problem involved. As a complement to these works, our approach considers

single agent-based reinforcement learning to comply with the fact noted in [47] that, when a multi-agent setting is modified by the actions of all agents, the environment becomes non-stationary, and the effectiveness of most reinforcement learning algorithms would not hold in non-stationary environments [48]. In addition, by further collaborating with the game-based iterative algorithms, our approach would reduce the overhead resulting from, e.g., the MM approach to resolve a complex optimization problem such as that in [46].

1.2. The Motivations and Characteristics of This Work

In recent years, advances in artificial intelligence are further helped by the neural networks such as generative adversarial networks [49] which use advanced game theory techniques to deep learn information and could converge to the Nash equilibrium of the game involved. In general, these advances can be reflected by the notion that a machine (computer) can learn about the outcomes of the game involved and teaches itself to do better based on the probabilities, strategies, and previous instances of the game and other players under the ground of game theory. By extending the advanced notation to the optimization framework, in this work, we further exhibit the possibility of applying learning-based methods, model-based methods, or both to resolve the joint beamforming, power control, and energy harvesting problem in the SWIPT-enabled wireless networks that can alleviate the hardness of finding an optimal solution with an optimization tool required to be completed in time. In particular, in this scenario, apart from BS i serving the user or MN needing to decide its transmit power, beamforming vector, and power splitting ratio, the other BSs $j \neq i$ would make their own decisions at the same time, which can affect the user or MN served by BS i simultaneously. Here, by leveraging the scenario, we conduct our approach to make a good trade-off between information decoding and energy harvesting, which can be deployed in an actual SWIPT-enabled IoT network as one of the various SWIPT applications surveyed in [50]. Specifically, by using the UE coordinates as that in [51] sent to BS, it can align with the industry specification [33] through the slight modification to reduce the original signal overhead of [33] on the channel state information to be sent by UE with a report to have its length equal to the number of antenna elements at least. As a summary, we list the characteristics of this work as follows:

- We introduce two single-layer algorithms based on the conventional DRL-based models, DQN and DDPG, to solve the joint optimization problem formulated here as a non-convex MINLP problem, and realized as an MRA problem subject to the different allocation constraints.
- We propose further a two-layer iterative approach that can incorporate the capability of data-driven DQN technique and the strength of non-cooperative game theory model to resolve the NP-hard MRA problem.
- For the two-layer approach, we also introduce a pricing strategy to determine the power costs based on the social utility maximization to control the transmit power.
- With the simulated environment based on realistic wireless networks, we show the results that, by means of both learning-based and model-based methods, the two-layer MRA algorithm proposed can outperform the single-layer counterparts introduced which rely only on the data-driven DRL-based models.

The rest of this paper is structured as follows. In Section 2, we introduce the network and channel models for this work. Next, we present the single-layer learning-based approaches in Section 3, followed by the two-layer hybrid approach based on game theory and deep reinforcement learning in Section 4. These approaches are then numerically examined in Section 5 to show their performance differences. Finally, conclusions are drawn in Section 6.

2. Network and Channel Models

2.1. Network Model

As shown in Figure 1, an orthogonal frequency division multiplexing (OFDM) multi-access network with $\widetilde{L}$ base stations (BSs) (or access points (APs)) is considered for downlink

transmission, in which a serving BS would associate with one mobile node (MN). The distance between two neighbor BSs is R and the cell radius (or transmission range) of BS is $\hat{r} > R/2$ to allow overlap. Here, unlike the conventional coordinated multipoint Tx/Rx (CoMP) system applied to the scenario in which a MN could receive data from multiple BSs, we apply the SWIPT technique to the network so that an MN can simultaneously receive not only wireless information but also energy from different BSs. In addition, although mmWave brings many performance benefits as an essential part of 5G, it is also known to have high propagation losses due to higher mmWave frequency bands to be adopted. Thus, analog beamforming for the downlink transmission is considered to alleviate these losses.

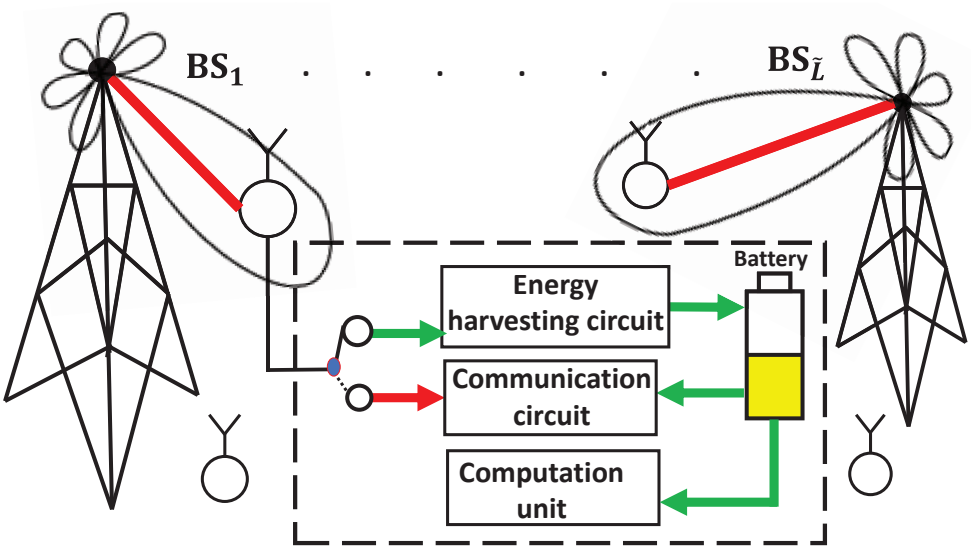

Figure 1. A system model with respect to the joint beamforming, power allocation, and splitting control for SWIPT-enabled IoT networks. In this model, each mobile node has a power split mechanism to split the received signal into two streams, one sent to the energy harvesting circuit for harvesting energy and the other to the communication circuit for decoding information.

Next, for more flexibly constructing a beampattern toward MN, each BS adopts a two-dimensional array of M antennas while each MN has a single antenna for transmission. Given that, the received signal at the MN associated with i-th BS would be

$$y_i = h_{i,i} f_i x_i + \sum_{j \neq i} h_{i,j} f_j x_j + n_i \tag{1}$$

In the above, $x_i, x_j \in \mathbb{C}$ are the transmitted signals form the i-th and j-th BSs, complying with the power constraint $\mathbb{E}\{|x_i|\} = P_i$ and $\mathbb{E}\{|x_j|\} = P_j$, where P_i and P_j are the transmit powers of the i-th and j-th BSs. In addition, $h_{i,i}, h_{i,j} \in \mathbb{C}^{M \times 1}$ are the channel vectors from the i-th and j-th BSs to the MN at the i-th BS, and $f_i, f_j \in \mathbb{C}^{M \times 1}$ denote the downlink beamforming vectors adopted at the i-th BS and j-th BSs, respectively. As the last term, n_i represents the noise at the receiver sampled from a complex normal distribution with zero mean and variance σ_n^2.

Beamforming: As mentioned previously, for the high propagation loss, analog beamforming vectors are assumed for transmission, and each $f_i, i = 1, 2, \cdots, |\mathcal{F}|$, consists of the beamforming weights for a two-dimensional (2D) planar array steered towards MN. More specifically, let each BS have a 2D array of antennas in the x–y plane, in which the antenna m is located at

$$d_m = (a_m \lambda, b_m \lambda) \tag{2}$$

where λ is the wavelength. Given the elevation direction ψ_d and the azimuthal direction ϕ_d, the phased weights for the 2D array steered towards the angle (ψ_d, ϕ_d) in the polar coordinates can be given by $e^{-j2\pi \sin\psi_d (a_m \cos\phi_d + b_m \sin\phi_d)}$. If the target is located on the

x–y plane, $\sin\psi_d$ will be 1 and the weights can be simplified as $e^{-j2\pi(a_m\cos\phi_d+b_m\sin\phi_d)}$. Given that, we consider every beamforming vector to be selected from a steering-based beamforming codebook $\mathcal{F}$ with $|\mathcal{F}|$ elements, wherein the n-th element or the array steering vector in the direction ϕ_n is given by

$$f_n \triangleq a(\phi_n) = \frac{1}{\sqrt{M}}\left[1, e^{-j2\pi(a_1\cos\phi_n+b_1\sin\phi_n)}, \cdots, e^{-j2\pi(a_{M-1}\cos\phi_n+b_{M-1}\sin\phi_n)}\right] \quad (3)$$

2.2. Channel Model

With the beamforming vector introduced above, we consider a narrow-band geometric channel model which is widely used for mmWave networks [52–54]. Specifically, the channel from BS i to the MN in BS j is formulated here as

$$h_{i,j} = \frac{\sqrt{M}}{\rho_{i,j}}\sum_{p=1}^{N_{i,j}^p}\alpha_{i,j}^p a(\phi_{i,j}^p) \quad (4)$$

where $\rho_{i,j}$ represents the path-loss between BS i and the MN associated with BS j. $\alpha_{i,j}^p$ is the complex path gain. $a(\phi_{i,j}^p)$ denotes the array response vector with respect to $\phi_{i,j}^p$, which is the angle of departure (AoD) of the p-th path. $N_{i,j}^p$ is the number of channel paths, and when compared with those for sub-6G, the number for mmWave is usually a small number [55,56]. Next, let the received power measured by the MN associated with BS i over a set of resource blocks (RBs) on the channel from BS j to the MN be $P_j|h_{i,j}f_j|^2$. Given that, the received signal to noise and interference ratio (SINR) for the MN associated with BS i can be obtained by

$$\frac{P_i|h_{i,i}f_i|^2}{\sum_{j\neq i}P_j|h_{j,i}f_j|^2+\sigma_n^2} \quad (5)$$

As shown above, each BS i uses P_i to transmit to its user with beamforming vector f_i. When incorporating SWIPT into power allocation, the use of beamforming on the mmWave MIMO system provides a new solution to resolve both interference and energy problems [57–59]. To this end, each MN in the network is installed with a power splitting unit to split the received signal for information decoding and energy harvesting simultaneously. Given that, the beamforming would provide a dedicated beam for MN through which power control and power splitting for energy harvesting can be realized at the same time. More specifically, in the power splitting architecture for downlink, the received signal at the MN associated with BS i which transmits with its beamforming vector f_i, and transmit power P_i is split into two separate signal streams according to the power split ratio θ_i, which will be determined in the sequel to maximize the system utility. In addition, when the technology of successive interference cancellation (SIC) is employed to mitigate the interference for data decoding, the stronger signal would be decoded first, and the weaker signals remaining could contribute to the interferences for decoding. With $\mathbf{P}$ and $\mathbf{F}$ to denote the sets for the transmit power and the power split ratio, respectively, in addition to the above, the SINR at the received MN i with SWIPT and SIC could be obtained by

$$\gamma_i(\mathbf{P},\theta_i,\mathbf{F}) = (1-\theta_i)\frac{P_i|h_{i,i}f_i|^2}{\sum_{j\neq i,P_i|h_{i,i}f_i|^2>P_j|h_{j,i}f_j|^2}P_j|h_{j,i}f_j|^2+\sigma_n^2} \quad (6)$$

As shown above, $1-\theta_i$ denotes the fraction of signal for the data transmission of SWIPT. In addition, with SIC [60], when there are multiple signals received by the MN associated with BS i concurrently, it will decode the stronger signal, and treat the weaker signals as interference. Here, if there are stronger signals from some BSs, they would be decoded and deleted first. Then, the desired signal will be obtained by treating the weaker

signals from the other BSs if they exist, noted here by $j \neq i$, $P_i|h_{i,i}f_i|^2 > P_j|h_{j,i}f_j|^2$, as the interference for decoding in addition to the noise σ_n^2.

2.3. Problem Formulation

Providing these essential models, our aim is to jointly optimize beamforming vectors, transmit powers, and power split ratios at the BSs to make the best trade-off between data rates, harvested energies, and power consumption from all MNs served in the SWIPT-enabled network with SIC, which is formulated as a complex multiple resource allocation (MRA) optimization problem subject to different allocation constraints that resulted from the different types of resources involved, shown as follows:

$$(P1) \max_{P_i,\theta_i,f_i,\forall i} \quad \sum_i U_i(\mathbf{P},\theta_i,\mathbf{F}) \tag{7a}$$

$$\textbf{subject to} \quad P_{min} \leq P_i \leq P_{max}, \quad \forall i \tag{7b}$$

$$0 \leq \theta_i \leq 1, \quad \forall i \tag{7c}$$

$$f_i \in \mathcal{F}, \quad \forall i \tag{7d}$$

where $U_i(\mathbf{P},\theta_i,\mathbf{F})$ in (7a) denotes the utility function for the trade-off to be introduced in (19). (7b) specifies the constraint that the transmit power, P_i, should be ranged between the minimum transmit power, P_{min}, and the maximum transmit power, P_{max}. (7c) requires θ_i to be a nonnegative ratio number no larger than 1. Finally, (7d) says that the vector, f_i, should be selected from its codebook $\mathcal{F}$.

Clearly, if U_i in the objective involves γ_i in (6), (P1) will be a mixed integer nonlinear programming (MINLP) problem. It would be even a non-convex MINLP problem due to the non-convexity of the objective function and the allocation constraints involving discrete values, and its solution is hard to find even using an optimization tool. To resolve this hard problem efficiently, we propose two kinds of innovative approaches based on deep reinforcement learning, game theory, or both, resulting in data-driven, model-driven, or hybrid iterative algorithms which could be operated in a single layer or two different layers, as introduced in the following. In addition, for clarity, we summarize the import symbols for the approaches to be introduced in Table A1 located in Appendix A due to its size.

3. Single-Layer Learning-Based Approaches

Determining an exact state transition model for (P1) through a model-based dynamic programming algorithm is challenging because the MRA problem on transmit power, power split ratio, and beamforming vector is location dependent. It is not trivial to list all the state–action pairs to be found in a state transition model predefined. Therefore, we design two single-layer learning-based algorithms derived from Markov decision process (MDP) to resolve this problem.

3.1. Q-Learning Approach

The Q-learning algorithm is based on the MDP that can be defined as a 4-tuple $<\widetilde{\boldsymbol{S}}, \widetilde{\boldsymbol{A}}, \widetilde{\boldsymbol{R}}, \widetilde{\boldsymbol{P}}>$, where $\widetilde{\boldsymbol{S}} = \{s_1, s_2, \cdots, s_{\mathtt{m}}\}$ is the finite set of states, and $\widetilde{\boldsymbol{A}} = \{a_1, a_2, \cdots, a_{\mathtt{n}}\}$ is the set of discrete actions. $\widetilde{R}(s,a,s')$ is the function to provide reward $\mathfrak{r}$ defined at state $s \in \widetilde{\boldsymbol{S}}$, action $a \in \widetilde{\boldsymbol{A}}$, and next state s'. $\widetilde{P}_{ss'(a)} = p(s'|s,a)$ is the transition probability of the agent at state s taking action a to migrate to state s'. Given that, reinforcement learning is conducted to find the optimal policy $\pi^*(s)$ that can maximize the total expected discounted reward. Among the different approaches to this end, Q-learning is widely considered, which adopts a value function $V^\pi(s) \rightarrow \mathfrak{r}$ for the expected value to be obtained by policy π from each $s \in \widetilde{S}$. Specifically, based on the infinite horizon discounted MDP, the value function in the following is formulated to show the goodness of π as

$$V^\pi(s) = \mathbb{E}_\pi \left\{ \sum_{k=0}^{\infty} \zeta^k \mathfrak{r}_i^{k+1} | s^0 = s \right\} \tag{8}$$

where $0 \leq \zeta \leq 1$ denotes the discount factor, and $\mathbb{E}$ is the expectation operation. Here, the optimal policy is defined to map the states to the optimal action in order to maximize the expected cumulative reward. In particular, the optimal action at each sate s can be obtained with the Bellman equation [61]:

$$V^*(s) = V^{\pi^*} = \max_{a \in \widetilde{A}} \left\{ \mathbb{E}\big(\mathfrak{r}(s,a) + \zeta \sum_{s' \in \widetilde{S}} P_{ss'} V^*(s')\big) \right\} \tag{9}$$

Given that, the action–value function is in fact the expected reward of this model starting from state s which takes action a according to policy π; that is,

$$Q^{\pi}(s,a) = \mathbb{E}\big(\mathfrak{r}(s,a)\big) + \zeta \sum_{s' \in \widetilde{S}} P_{ss'} V(s') \tag{10}$$

Let the optimal policy $Q^*(s,a)$ be Q^{π^*}. Then, we can obtain

$$V^*(s) = \max_{a \in \widetilde{A}} Q^*(s,a) \tag{11}$$

The strength of Q-learning can now be revealed as it can learn π^* without knowing the environment dynamics or $P_{ss'}(a)$, and the agent can learn it by adjusting the Q value with the following update rule:

$$Q(s_t,a_t) = (1-\alpha) Q(s_t,a_t) + \alpha \left[\mathfrak{r}_t + \zeta \max_{a' \in \widetilde{A}} Q(s',a') \right] \tag{12}$$

where $\alpha \in [0,1)$ denotes the learning rate.

Given this strength, the application of Q-learning is, however, limited because the optimal policy can be obtained only when the state-action spaces are discrete and the dimension is relatively small. Fortunately, after considerable investigations on the deep learning techniques, reinforcement learning has made significant progress to replace a Q-table with the neural network, leading to DQN that can approximate $Q(s_t,a_t)$. In particular, in DQN, the Q value in time t is rewritten as $Q(s_t,a_t,\omega)$ wherein ω is the weight of a deep neural network (DNN). Given that, the optimal policy $\pi^*(s)$ in DQN can be represented by $\pi^*(s) = \arg\max_{a'} Q^*(s_t,a',\omega)$, where Q^* denotes the optimal Q value obtained through DNN. The goal of this approach is then to choose the approximated action $a_{t+1} = \pi^*(s_{t+1})$, and the approximated Q value is given by

$$\hat{Q}(s_t,a_t,\omega') = \mathfrak{r}(s_t,a_t,\omega') + \zeta \max_{a' \in \widetilde{A}} \big[Q(s_{t+1},a',\omega) \big] \tag{13}$$

In the above, ω will be updated by minimizing the loss function:

$$\hat{L} = \mathbb{E}\Big[\big(\hat{Q}(s_t,a_t,\omega') - Q(s_{t+1},a_{t+1},\omega') \big)^2 \Big] \tag{14}$$

Deep Q learning elements: Following the Q-learning design approach, we next define state, action, and reward function specific for solving (P1) as follows:

(1) **State:** First, if there are n links in the network, the state at time t is represented in the sequel by using the capital notations for their components and using the superscript such as "(t)" for the time index as follows:

$$s^{(t)} = \left\{ \mathbf{L}^{(t)}, \mathbf{P}^{(t)}, \mathbf{\Theta}^{(t)}, \mathbf{F}^{(t)} \right\} \tag{15}$$

where $\mathbf{L}^{(t)} = \left\{ L_1^{(t)}, \cdots, L_n^{(t)} \right\}$, $\mathbf{P}^{(t)} = \left\{ P_1^{(t)}, \cdots, P_n^{(t)} \right\}$, $\mathbf{\Theta}^{(t)} = \left\{ \theta_1^{(t)}, \cdots, \theta_n^{(t)} \right\}$, and $\mathbf{F}^{(t)} = \left\{ f_1^{(t)}, \cdots, f_n^{(t)} \right\}$. In the above, $L_i^{(t)} = (X_i^{(t)}, Y_i^{(t)})$ denotes the Cartesian coor-

dinates of MN in link i at time t, while the others, i.e., $P_i^{(t)}$, $\theta_i^{(t)}$, and $f_i^{(t)}$, denote the transmit power, power splitting ratio, and beamforming vector for link i at time t, respectively.

Among these variables, the transmit power is usually the only parameter to be considered in many previous works [27,62]. In the complex MRA problem also involving other types of resources, it is still a major factor affecting the system performance based on SINR in (5) that would be significantly impacted by the power, and thus we consider two different state formulations for $\mathbf{P}^{(t)}$ as follows.

- *Power state formulation 1 (PSF1):* First, to align with the industry standard [33] which chooses integers for power increments, we consider a ± 1 dB offset representation similar to that shown in [51], as the the first formulation for the power state. Specifically, given an initial value P_i^0, the transmit power $P_i, \forall i$ (despite t), will be chosen from the set

$$\mathcal{P}_i^1 \triangleq \left\{10^{-0.1\cdot K_{min}} P_i^0, \cdots, 10^{-0.1} P_i^0, P_i^0, 10^{0.1} P_i^0, \cdots, 10^{0.1\cdot K_{max}} P_i^0\right\} \quad (16)$$

  where K_{min}=$\lfloor$-10 $\log_{10}(\frac{P_{min}}{P_i^0})\rfloor$ and K_{max}=$\lfloor 10 \log_{10}(\frac{P_{max}}{P_i^0})\rfloor$.
- *Power state formulation 2 (PSF2):* Next, as shown in [27], the performance of a power-controllable network can be improved by quantizing the transmit power through a logarithmic step size instead of linear step size. Given that, the transmit power $P_i, \forall i$ could be selected from the set

$$\mathcal{P}^2 \triangleq \left\{ P_{min}\left(\frac{P_{min}}{P_{max}}\right)^{\frac{j}{|\mathcal{P}^2|-2}} \middle| j = 0, \cdots, |\mathcal{P}^2| - 2 \right\} \quad (17)$$

Apart from the above, the other parameters, such as $\theta_i, \forall i$, can be chosen from the splitting ratio set Θ with linear step size, and $f_i, \forall i$ can be selected from the predefined codebook $\mathcal{F}$ with $|\mathcal{F}|$ finite vectors or elements.

(2) **Action:** The action of this process at time t, $a^{(t)}$ is selected from a set of binary decisions on the variables

$$\hat{\mathbf{A}} = \left\{\hat{\mathbf{A}}_P, \hat{\mathbf{A}}_\Theta, \hat{\mathbf{A}}_F\right\} \quad (18)$$

where $\hat{\mathbf{A}}_P = \hat{A}_{p_1} \times \hat{A}_{p_2} \cdots \times \hat{A}_{p_n} \in \{\pm 1\}^n$, $\hat{\mathbf{A}}_\Theta = \hat{A}_{\theta_1} \times \hat{A}_{\theta_2} \cdots \times \hat{A}_{\theta_n} \in \{\pm 1\}^n$, and $\hat{\mathbf{A}}_F = \hat{A}_{f_1} \times \hat{A}_{f_2} \cdots \times \hat{A}_{f_n} \in \{\pm 1\}^n$ denote all the possible binary decisions on the three types of variables involved, respectively. That is, the agent can decide each link i to increase or decrease each of the variables to the next quantized value according to $\hat{A}_{p_i}^{(t)}$, $\hat{A}_{\theta_i}^{(t)}$ and $\hat{A}_{f_i}^{(t)}$ in $a^{(t)}$, respectively.

Note that, as the number of values of a variable is limited, when reaching the maximum or minimum value with a binary action chosen from $\hat{\mathbf{A}}$, a modulo operation is used to decide the index for the next quantized value in the state space. For example, in PSF2, if $P_i^{(t)} = P_{min}\left(\frac{P_{min}}{P_{max}}\right)^{\frac{j}{|\mathcal{P}^2|-2}}$ with $j = 0$, and $j + \hat{A}_{p_i}^{(t)} < 0$, then the modulo operation will lead to $P_i^{(t+1)} = P_{min}\left(\frac{P_{min}}{P_{max}}\right)^{\frac{j'}{|\mathcal{P}^2|-2}}$ with $j' = |\mathcal{P}^2| - 2$ in $\mathcal{P}^2$. As another example, with $f_{min} = 1$ and $f_{max} = |\mathcal{F}|$ to denote the first and the last vector in the codebook $\mathcal{F}$, respectively, the action of increasing or decreasing $f_{min} \le f_i^{(t)} \le f_{max}$ by 1 will choose the previous or the next vector of $f_i^{(t)}$ in $\mathcal{F}$ as $f_i^{(t+1)}$, and a similar modulo operation will also be applied to keep $f_i^{(t+1)}$ within $[f_{min}, f_{max}]$.

(3) **Reward:** To reduce the power consumption for green communication while maintaining the desired trade-off among the data rate and the energy harvesting, we introduce

a reward function that can represent a trade-off among the three metrics properly normalized for link i with parameters λ_i, μ_i, and ν_i, at time t, as

$$U_i(\mathbf{P}^{(t)}, \theta_i^{(t)}, \mathbf{F}^{(t)}) = \lambda_i r_i(\mathbf{P}^{(t)}, \theta_i^{(t)}, \mathbf{F}^{(t)}) + \mu_i E_i(\mathbf{P}^{(t)}, \theta_i^{(t)}, \mathbf{F}^{(t)}) - \nu_i P_i^{(t)} \tag{19}$$

where $r_i(\mathbf{P}^{(t)}, \theta_i^{(t)}, \mathbf{F}^{(t)})$ denotes the data rate of link i obtained at time t, which can be represented by

$$r_i(\mathbf{P}^{(t)}, \theta_i^{(t)}, \mathbf{F}^{(t)}) = \log(1 + \gamma_i(\mathbf{P}^{(t)}, \theta_i^{(t)}, \mathbf{F}^{(t)})) \tag{20}$$

In addition, $E_i(\mathbf{P}^{(t)}, \theta_i^{(t)}, \mathbf{F}_i^{(t)})$ is the energy harvested at MN of link i at time t, represented in the log scale as

$$E_i(\mathbf{P}^{(t)}, \theta_i^{(t)}, \mathbf{F}^{(t)}) = \log\left(e_i(\mathbf{P}^{(t)}, \theta_i^{(t)}, \mathbf{F}^{(t)})\right) \tag{21}$$

wherein the harvested energy in its raw form is given by

$$e_i(\mathbf{P}^{(t)}, \theta_i^{(t)}, \mathbf{F}^{(t)}) = \theta_i \delta\left(P_i^{(t)} |h_{i,i}^{(t)} f_i^{(t)}|^2 + \sum_{j \neq i} P_j^{(t)} |h_{j,i}^{(t)} f_j^{(t)}|^2 + \sigma_n^2\right) \tag{22}$$

In the above, δ is the power conversion efficiency, and ν_i is the price or cost for the power consumption $P_i^{(t)}$ to be paid for link i's transmission. Note that the log representation is considered here to accommodate a normalization process in deep learning similar to the batch normalization in [63]. Otherwise, the data rate $r_i(\mathbf{P}^{(t)}, \theta_i^{(t)}, \mathbf{F}^{(t)})$ obtained with a log operation and the raw energy harvesting $e_i(\mathbf{P}^{(t)}, \theta_i^{(t)}, \mathbf{F}^{(t)})$ without the (log) operation may be directly combined in the utility function. If so, with the metric values lying in very different ranges, such a raw representation could cause problems in the training process. Note also that, although λ_i and μ_i could be set to compensate the scale differences, a very high energy obtained in certain case can still happen to significantly vary the utility function and impede the learning process. By taking these into account, the system utility at time t can be represented by the sum of these link rewards as

$$U^{(t)} = U(\mathbf{P}^{(t)}, \mathbf{\Theta}^{(t)}, \mathbf{F}^{(t)}) = \sum_i U_i(\mathbf{P}^{(t)}, \theta_i^{(t)}, \mathbf{F}^{(t)}) \tag{23}$$

Policy selection: In general, Q-learning is an off-policy algorithm that can find a suboptimal policy even when its actions are obtained from an arbitrary exploratory selection policy [64]. Following that, we conduct the DQN-based MRA algorithm to have a near-greedy action selection policy, which consists of (1) exploration mode and (2) exploitation mode. On the one hand, in exploration mode, the DQN agent would randomly try different actions at every time t for getting a better state-action or Q value. On the other hand, in exploitation mode, the agent will choose at each time t an action $a^{(t)}$ that can maximize the Q value via DNN with weight ω; that is, $a^{(t)} = \arg\max_{a' \in \mathbf{A}} Q^*(s_t, a', \omega)$. More specifically, we conduct the agent to explore with a probability ϵ and to exploit with a probability $1 - \epsilon$, where $\epsilon \in (0, 1)$ denotes a hyperparameter to adjust the trade-off between exploration and exploitation, resulting in a ϵ-greedy selection policy.

Experience replay: This algorithm also includes a buffer memory D as a replay memory to store transactions $(s^{(t)}, a^{(t)}, \mathfrak{r}^{(t)}, s')$, where reward $\mathfrak{r}^{(t)} = U^{(t)}$ is obtained by (23) at time t. Given that, at each learning step, a mini-batch is constructed by randomly sampling the memory pool and then a stochastic gradient descent (SGD) is used to update ω. By reusing the previous experiences, the experience replay makes the stored samples to be exploited more efficiently. Furthermore, by randomly sampling the experience buffer, a more independent and identically distributed data set could be obtained for training.

As a summary of these key points introduced above, we formulate the single-layer DQN-based MRA training algorithm with a pseudo code representation shown in Algorithm 1 for easy reference.

Algorithm 1 The single-layer DQN-based MRA training algorithm.

(Input) $\lambda_i, \mu_i, \nu_i, \forall i$, batch size η, learning rate α, minimum exploration rate ϵ_{min}, discount factor ζ, and exploration decay rate d;
(Output) Learned DQN to decide $P_i, \theta_i, f_i, \forall i$, for (7);
Initialize action $a^{(0)}$ and replay buffer $D = \emptyset$;
for episode = 1 to $\mathcal{M}$ **do**
  Initialize state $s^{(0)}$;
  for time $t = 1$ to $\mathcal{N}$ **do**
    Observe current state $s^{(t)}$;
    $\epsilon = \max(\epsilon \cdot d, \epsilon_{min})$;
    if random number $r < \epsilon$ **then**
      Select $a^{(t)} \in \hat{\mathbf{A}}$ at random;
    else
      Select $a^{(t)} = \arg\max_{a'} Q^*(s^{(t)}, a', \omega)$;
    end if
    Observe next state s';
    Store transition $(s^{(t)}, a^{(t)}, \mathfrak{r}^{(t)}, s')$ in D, where $\mathfrak{r}^{(t)}$ is $U^{(t)}$ obtained with (23);
    Select randomly η stored samples $(s^{(j)}, a^{(j)}, \mathfrak{r}^{(j)}, s^{(j+1)})$ from D for experience;
    Obtain $\hat{Q}(s^{(j)}, a^{(j)}, \omega')$ for all j samples with (13);
    Perform SGD to minimize the loss in (14) for finding the optimal weight of DNN, ω^*;
    Update $\omega = \omega^*$ in the DQN;
    $s^{(t)} = s'$;
  end for
end for

3.2. DDPG-Based Approach

Similar to that found in the literature [28,29], as a deep reinforcement learning algorithm, DQN would be superior to the classical Q-learning algorithm because it can handle the problems with high-dimensional state spaces that can hardly be done with the former. However, DQN still works on a discrete action space, and suffers the curse of dimensionality when the action space becomes large. For this, we next develop a deep deterministic policy gradient (DDPG)-based algorithm that can find optimal actions in a continuous space to solve this MRA optimization problem without quantizing the actions that should be done for the DQN-based algorithm.

Specifically, with DDPG, we aim to determine an action a to maximize the action–value function $Q(s, a)$ for a given state s. That is, our goal is to find

$$a^*(s) = \arg\max_a Q(s, a) \tag{24}$$

as that done with DQN introduced previously. However, unlike DQN, there are two neural networks for DDPG, namely actor network and critic network, and each contains two subnets, namely online net and target net, with the same architecture. First, the actor network with the weight of DNN, ω_a, which is called "actor parameter", will take state s to output a deterministic action a, denoted by $Q_a(s; \omega_a)$. Second, the critic network with the weight of DNN, ω_c, which is called "critic parameter" will take state s and a as its inputs to produce the state–value function, denoted by $Q(s, a; \omega_c)$, to simulate a table for Q-learning or Q-table that would get rid of the curse of dimensionality. Given that, two key features of DDPG can be summarized as follows:

(1) *Exploration*: As defined, the actor network is conducted to provide solutions to the problem, playing a crucial role in DDPG. However, as it is designed to produce only

deterministic actions, additional noise, n, is added to the output so that the actor network can explore the solution space. That is,

$$a(s) = Q_a(s;\omega_a) + n \tag{25}$$

(2) *Updating the networks*: Next, with the notation $(s, a, \mathfrak{r}, s')$ to denote the transaction wherein reward $\mathfrak{r}$ is obtained by taking action a at state s to migrate to s' as that in DQN, the update procedures for the critic and actor networks can be further summarized in the following.

- As shown in (24), the actor network is updated by maximizing the state–value function. In terms of the parameters ω_a and ω_c, this maximization problem can be rewritten to find $J(\omega_a) = Q(s, a; \omega_c)|_{a=Q_a(s;\omega_a)}$. Here, as the action space is continuous and the state–value function is assumed to be differentiable, the actor parameter, ω_a, would be updated by using the gradient ascent method. Furthermore, as the gradient depends on the derivative of the objective function with respect to ω_a, the chain rule can be applied as

$$\nabla\omega_a J(\omega_a) = \nabla_a Q(s, a; \omega_c)|_{a=Q_a(s;\omega_a)} \nabla\omega_a Q_a(s;\omega_a) \tag{26}$$

  Then, as the actor network would output $Q_a(s;\omega_a)$ to be the action adopted by the critic network, the actor parameter ω_a can be updated by maximizing the critic network's output with the action obtained from the actor network, while fixing the critic parameter ω_c.
- Apart from the actor network to generate the needed actions, the critic network is also crucial to ensure that the actor network is well trained. To update the critic network, there are two aspects to be considered. First, with $Q_{a'}(s;\omega_{a'})$ from the target actor network to be an input of the target critic network, the state–value function would produce

$$y = \mathfrak{r} + \zeta\bar{Q}(s', a; \omega_c)|_{a=Q_{a'}(s;\omega_{a'})} \tag{27}$$

  Second, the output of the critic network, $Q(s, a; \omega_c)$, can be regarded as another source to estimate the state–value function. Based on these aspects, the critic network can be updated by minimizing the following loss function:

$$\hat{L} = (y - Q(s, a; \omega_c))^2 \tag{28}$$

  Given that, the critic parameter, ω_c, can be obtained by finding the parameter to minimize this loss function.
- Finally, the target nets in both critic and actor networks can be updated with the soft update parameter, τ, on their parameters ω_c' and ω_a', as follows:

$$\omega_c' = \tau\omega_c + (1-\tau)\omega_c', \quad \omega_a' = \tau\omega_a + (1-\tau)\omega_a' \tag{29}$$

Action representation for the MRA problem: As defined, the actor network outputs the deterministic action $a^* = Q_a(s;\omega_a)$. Due to the deterministic, a dynamic ϵ-greedy policy is used to determine the action by adding a noise term $n^{(t)}$ to explore the action space. Here, as the state of this work involves different types of variables, the action resulting at time t in fact consists of three parts as $a^{(t)^*} = \left\{A_P^{(t)^*}, A_\Theta^{(t)^*}, A_F^{(t)^*}\right\}$. When added with the corresponding noises, the exploration action $a^{(t)}$ would be specified as

$$a^{(t)} = \left[[A_P^{(t)^*} + n_P^{(t)}]_{P_{low}}^{P_{up}}, [A_\Theta^{(t)^*} + n_\Theta^{(t)}]_{\Theta_{low}}^{\Theta_{up}}, [A_F^{(t)^*} + n_F^{(t)}]_{F_{low}}^{F_{up}}\right] \tag{30}$$

where the different parts of $a^{(t)}$ are clipped to the intervals $[x_{up}, x_{low}]$, $x \in \{P, \Theta, F\}$, according to the different types of variables, and the added noises are obtained with a normal distribution also based on the different types as

$$n_x^{(t)} \sim \mathcal{N}\left(0, d^{(t)}(x_{up} - x_{low})\right) \tag{31}$$

where $d^{(t)}$ denotes the exploration decay rate at time t.

State normalization and quantization: As shown in the previous works [32,63,65], a state normalization to preprocess the training sample sets would lead to a much easier and faster training process. In our work, the three types of variables, $\mathbf{P}^{(t)}$, $\mathbf{\Theta}^{(t)}$, and $\mathbf{F}^{(t)}$ (shown in vector forms) in $s^{(t)}$ may have their values lying in very different ranges, which could cause problems in a training process. To prevent them, we normalize the coordinates with the cell radius, and these variables with the scale factors ς_1, ς_2, and ς_3, as

$$P_i^{(t)} = \varsigma_1 \frac{\tilde{P}_i^{(t)}}{P_{max}}, \theta_i^{(t)} = \varsigma_2 \frac{\tilde{\theta}_i^{(t)}}{\theta_{max}}, f_i^{(t)} = \varsigma_3 \frac{\tilde{f}_i^{(t)}}{f_{max}}, \forall i \tag{32}$$

In the above, $\tilde{f}_i$ is an integer variable rounded from its real counterpart to denote which element in the codebook $\mathcal{F}$ to be used because the output of DDGP is a continuous action. Specifically, given $a^{(t)}=\left\{a_P^{(t)}, a_\Theta^{(t)}, a_F^{(t)}\right\}$ where $a_{f_i}^{(t)} \in a_F^{(t)} = [A_F^{(t)^*} + n_F^{(t)}]_{F_{low}}^{F_{up}}$ is obtained by (30), its value at time t will be

$$\tilde{f}_i^{(t)} = \left[\lfloor f_i^{(t)} f_{max}/\varsigma_3 + a_{f_i}^{(t)} \rfloor\right]_{f_{min}}^{f_{max}} \tag{33}$$

Note that, after the rounding operation (represented here by the floor function), the value may still be out of its feasible range, and thus a modulo operation similar to that for DQN is also applied here to keep it in $[f_{min}, f_{max}]$. For the other types of variables, the corresponding modulo operations are required to keep them in their feasible ranges as well. Still, due to their continuous nature, a rounding operation is avoided. Specifically, with $a_{P_i}^{(t)} \in a_P^{(t)}$ and $a_{\theta_i}^{(t)} \in a_\Theta^{(t)}$, each $\tilde{P}_i^{(t)}$ and $\tilde{\theta}_i^{(t)}$ at time t would be updated by

$$\tilde{P}_i^{(t)} = \left[P_i^{(t)} P_{max}/\varsigma_1 + a_{P_i}^{(t)}\right]_{P_{min}}^{P_{max}} \tag{34}$$

$$\tilde{\theta}_i^{(t)} = \left[\theta_i^{(t)} \theta_{max}/\varsigma_2 + a_{\theta_i}^{(t)}\right]_{\theta_{min}}^{\theta_{max}} \tag{35}$$

Apart from the above, the critic network $Q(s_c, a; \omega_c)$ is conducted to transfer gradient in learning, which is not involved in action generation. In particular, the critic network evaluates the current control action based on the performance index (23) while the parameters $\mathbf{P}^{(t)}$, $\mathbf{\Theta}^{(t)}$, and $\mathbf{F}^{(t)}$ of U in (23) are obtained by the actor network. Apart from these networks, the DDPG-based algorithm also includes an experience replay mechanism as the DQN counterpart. That is, when the experience buffer is full, the current transition $(s^{(t)}, a^{(t)}, \mathfrak{r}^{(t)}, s')$ will replace the oldest one in the buffer D where reward $\mathfrak{r}^{(t)} = U^{(t)}$, and then the algorithm would randomly choose η stored transitions to form a mini-batch for updating the networks. Given these sampled transitions, the critic network can update its online net by minimizing the loss function represented by

$$\hat{L}_\eta = \frac{1}{\eta} \sum_i (y_i - Q(s_i, a; \omega_c))^2 \tag{36}$$

where $y_i = \mathfrak{r}_i + \zeta \bar{Q}(s_i', a; \omega_c)|_{a=Q_{a'}(s_i;\omega_{a'})}$. Similarly, the actor network can update its online net with

$$\nabla \omega_a J_\eta(\omega_a) = \frac{1}{\eta} \sum_i \nabla_a Q(s_i, a; \omega_c)|_{a=Q_a(s_i;\omega_a)} \nabla \omega_a Q_a(s_i; \omega_a) \tag{37}$$

Finally, we summarize the single-layer DDPG-based MRA training algorithm in Algorithm 2 to be referred to easily.

Algorithm 2 The single-layer DDPG-based MRA training algorithm.

(Input) $\lambda_i, \mu_i, \nu_i, \forall i$, batch size η , actor learning rate α_a, critic learning rate α_c, decay rate d, discount factor ζ, and soft update parameter τ;
(Output) Learned actor/critic to decide $P_i, \theta_i, f_i, \forall i$, for (7);
Initialize actor $Q_a(s;\omega_a)$, critic $Q(s,a;\omega_c)$, action $a^{(0)}$, replay buffer D, and set initial decay rate $d^{(0)} = 1$;
for episode = 1 to $\mathcal{M}$ **do**
  Initialize state $s^{(0)}$ and $\rho^{(0)}$;
  for time $t = 1$ to $\mathcal{N}$ **do**
    Normalize state $s^{(t)}$ with (32);
    Execute action $a^{(t)}$ in (30), obtain reward $\mathfrak{r}^{(t)} = U^{(t)}$ with (23), and observe new state s';
    if replay buffer D is not full **then**
      Store transition $(s^{(t)}, a^{(t)}, \mathfrak{r}^{(t)}, s')$ in D;
    else
      Replace the oldest one in buffer D with $(s^{(t)}, a^{(t)}, \mathfrak{r}^{(t)}, s')$;
      Set $d^{(t)} = d^{(t-1)} \cdot d$;
      Randomly choose η stored transitions from D;
      Update the critic online network by minimizing the loss function in (36);
      Update the actor online network with the gradient obtained by (37);
      Soft update the target networks with their parameters updated by (29);
      $s^{(t)} = s'$;
    end if
  end for
end for

4. Two-Layer Hybrid Approach Based on Game Theory and Deep Reinforcement Learning

As exhibited above, DDPG can be used for continuous action spaces as well as high-dimensional state spaces, which would overcome the difficulty of DQN which can apply only to discrete action spaces. However, the MRA problem includes both discrete and continuous variables, which requires DDPG to quantize the continuous variables involved to be their discrete counterparts as shown in (33). In addition, as a data driven approach, deep reinforcement learning does not explicitly benefit from an analytic model specific to the problem. To take the advantages from both data-driven and model-driven approaches, we propose in the following a novel approach that consists of two layers, where the lower layer is responsible for the continuous power allocation (PA) and energy harvest splitting (EHS) by using a game-theory-based iterative method, and the upper layer resolves the discrete beam selection problem (BSP) by using a DQN algorithm. That is, if $f_i, \forall i$, can be given, PA and EHS on P_i and θ_i for each link i could be decomposed from the objective. Then, we could simplify the MRA problem by reducing (P1) to a BSP sub-problem and a PA/EHS sub-problem. Specifically, the latter (PA/EHS) is given by

$$(P2) \quad \max_{P_i,\theta_i} \quad U_i(\mathbf{P}, \theta_i, \mathbf{F}) \tag{38a}$$

$$\text{subject to} \quad P_{min} \le P_i \le P_{max} \tag{38b}$$

$$0 \le \theta_i \le 1 \tag{38c}$$

Clearly, if the BSP sub-problem can be solved, the major challenge of this approach would be the PA/EHS sub-problem shown in (P2). Here, even represented by a simpler form, (P2) is still a non-convex problem whose solution for link i will depend on the other links $j \neq i$. That is, despite EHS, the PA problem still remains in (P2) that a larger P_i would increase SINR of link i while reducing those of the other links $j \neq i$ in (6), increase energy harvesting in (22), or both, at the cost ν_i for P_i in the objective function.

4.1. Game Model

To overcome this difficulty, we convert (P2) into a non-cooperative game among the multiple links which could be regarded as self-interesting players and finding its Nash equilibrium (NE) is the fundamental issue to be considered in this game model. On the one hand, a link i can be seen as a non-cooperative game player who can choose its own P_i and θ_i to make a trade-off so that a larger P_i will lead to a higher SINR value in (6) for data rate, a higher value in (22) for energy harvesting, or both on the cost of a higher power consumption, and vice versa. On the other hand, the utility given in (19) can be considered to reduce the power consumption for green communication while maintaining a desired trade-off among the data rate and the energy harvesting. The game-based pricing strategy is thus designed through which BS can require its link to pay a certain price for the power consumption on its transmission. For this, λ_i can be interpreted as the willingness of player i to pay for the data rate, and μ_i as that to pay for the energy harvesting. Given that, each link or player i can determine its P_i and θ_i based on price ν_i to maximize its own utility, and in this maximization, λ_i, μ_i, and ν_i are predetermined values for player i and unknown for the others $j \neq i$, as a basis for the non-cooperative game.

4.2. Existence of Nash Equilibrium

To ensure the outcome of the non-cooperative game to be effective, we next show this game to have at least one Nash equilibrium. As noted in [66], a Nash equilibrium point represents a situation wherein every player is unilaterally optimal and no player can increase its utility alone by changing its own strategy. Furthermore, according to the game theory fundamental [66], the non-cooperative game admits at least one Nash equilibrium point if (1) the strategy space is a nonempty, compact and convex set, and (2) the utility function is continuous quasiconcave with respect to the action space. In (P2), the utility function U_i can be verified to satisfy the above conditions. Specifically, for the first condition, we can note that the transmit power is bounded by P_{min} and P_{max}, i.e., $P_i \in [P_{min}, P_{max}]$, and the power splitting ratio, θ_i, is a real number bounded by 0 and 1. Let S_i be the set of all strategies as its strategy space. Then, the strategy space for each link i in the proposed game model can be represented by S_i=$\{(P_i, \theta_i) \in \mathbb{R}^2 | P_{min} \leq P_i \leq P_{max}, 0 \leq \theta_i \leq 1\}$, which is a compact (closed and bounded) convex set as required.

For the second condition, we can derive the partial differential of the utility function with respect to power P_i as

$$\frac{\partial U_i}{\partial P_i} = \frac{\lambda_i(1-\theta_i)|h_{i,i}f_i|^2}{R_i - \theta_i P_i |h_{i,i}f_i|^2} + \frac{\mu_i \theta_i \delta |h_{i,i}f_i|^2}{\theta_i \delta \left(P_i |h_{i,i}f_i|^2 + \sum_{j \neq i} P_j |h_{j,i}f_j|^2 + \sigma_n^2\right)} - \nu_i \tag{39}$$

where R_i is the total received power at link i, which accommodates the effect of SIC involved, as shown as follows:

$$R_i = \begin{cases} \sigma_n^2 + \sum_{j \neq i} P_j |h_{j,i}f_j|^2 + P_i |h_{i,i}f_i|^2, & \text{if } P_i |h_{i,i}f_i|^2 > \sum_{j \neq i} P_j |h_{j,i}f_j|^2 \\ \sigma_n^2 + P_i |h_{i,i}f_i|^2 & \text{, otherwise} \end{cases} \tag{40}$$

Similarly, we can obtain the partial differential of the utility with respect to θ_i by

$$\begin{aligned} \frac{\partial U_i}{\partial \theta_i} &= \frac{\lambda_i}{1+\gamma_i} \frac{\partial \gamma_i}{\partial \theta_i} + \frac{\mu_i \delta \left(P_i |h_{i,i}f_i|^2 + \sum_{j \neq i} P_j |h_{j,i}f_j|^2 + \sigma_n^2\right)}{\theta_i \delta \left(P_i |h_{i,i}f_i|^2 + \sum_{j \neq i} P_j |h_{j,i}f_j|^2 + \sigma_n^2\right)} \\ &= \frac{-\lambda_i P_i |h_{i,i}f_i|^2}{R_i - \theta_i P_i |h_{i,i}f_i|^2} + \frac{\mu_i}{\theta_i} \end{aligned} \tag{41}$$

Furthermore, from (39) and (41), the second derivative of the utility function with respect to P_i and θ_i, respectively, can be obtained by

$$\begin{aligned}\frac{\partial U_i^2}{\partial P_i^2} &= -\lambda_i\left(\frac{(1-\theta_i)|h_{i,i}f_i|^2}{R_i-\theta_i P_i|h_{i,i}f_i|^2}\right)^2-\mu_i\left(\frac{|h_{i,i}f_i|^2}{P_i|h_{i,i}f_i|^2+\sum_{j\neq i}P_j|h_{j,i}f_j|^2+\sigma_n^2}\right)^2\\ \frac{\partial U_i^2}{\partial \theta_i^2} &= -\lambda_i\left(\frac{P_i|h_{i,i}f_i|^2}{R_i-\theta_i P_i|h_{i,i}f_i|^2}\right)^2-\mu_i\left(\frac{1}{\theta_i}\right)^2\end{aligned} \tag{42}$$

It is easy to see that both $\frac{\partial U_i^2}{\partial P_i^2}$ and $\frac{\partial U_i^2}{\partial \theta_i^2}$ are less than or equal to 0, implying that the utility function is convex. In addition, U_i is continuous in P_i. Consequently, the utility functions, $U_i, \forall i$, all satisfy the required conditions for the existence of at least one Nash equilibrium.

4.3. Power Allocation and Energy Harvest Splitting in the Lower Layer

Based on the non-cooperative game model introduced, the associated BS is responsible for deciding the transmit power P_i and the power splitting ratio θ_i for link i, with the channel state information $h_{i,j}$ and the weights λ_i and μ_i, which can be done by finding its Nash equilibrium. To see this, we note that, as the utility functions $U_i, \forall i$, are concave down with respect to (P_i, θ_i), this decision can be made by using the solution to the system of equations:

$$\frac{\partial U_1}{\partial P_1}=0,\ldots,\frac{\partial U_n}{\partial P_n}=0,\frac{\partial U_1}{\partial \theta_1}=0,\ldots,\frac{\partial U_n}{\partial \theta_n}=0 \tag{43}$$

where n denotes the number of links in the network.

To solve the system of equations, we propose an iterative algorithm based on the game model, and through the fixed point iteration process, the system of Equation (43) can be solved numerically. Here, by taking the derivative with respect to P_i (resp. θ_i) and setting the result equal to 0, we can transform the system into a fixed point form for each link i that can facilitate its convergence, as follows:

$$P_i = \frac{\lambda_i}{\nu_i-\frac{\mu_i|h_{i,i}f_i|^2}{\hat{R}_i+P_i|h_{i,i}f_i|^2}}-\frac{\hat{R}_i}{(1-\theta_i)|h_{i,i}f_i|^2} \tag{44}$$

$$\theta_i = \frac{\mu_i R_i}{(1+\lambda_i)P_i|h_{i,i}f_i|^2} \tag{45}$$

where $\hat{R}_i$ is an auxiliary variable denoted by

$$\hat{R}_i=\sigma_n^2+\sum_{j\neq i}P_j|h_{j,i}f_j|^2 \tag{46}$$

To show the iterative process more clearly, we denote the transmit power, the total received power, the auxiliary variable, and the power splitting ratio, for link i at the k-th iteration, by $P_i[k]$, $R_i[k]$, $\hat{R}_i[k]$, and $\theta_i[k]$, respectively. Given that, the iterations on P_i and θ_i can be shown by the relationships between iterations k and $k-1$ with their results to be bounded by the corresponding maximum and minimum values as follows:

$$P_i[k]=\left[\frac{\lambda_i}{\nu_i-\frac{\mu_i|h_{i,i}f_i|^2}{\hat{R}_i[k-1]+P_i[k-1]|h_{i,i}f_i|^2}}-\frac{\hat{R}_i[k-1]}{(1-\theta_i[k-1])|h_{i,i}f_i|^2}\right]_{P_{min}}^{P_{max}} \tag{47}$$

$$\theta_i[k]=\left[\frac{\mu_i R_i[k-1]}{(1+\lambda_i)P_i[k-1]|h_{i,i}f_i|^2}\right]_{\theta_{min}}^{\theta_{max}} \tag{48}$$

4.4. Beam Selection in the Upper Layer and the Overall Algorithm

With the transmit powers and energy splitting ratios from the lower layer with a low cost, the two-layer hybrid approach is designed to resolve the remaining beam selection problem with a DQN-based algorithm in the upper layer, which would reduce the computational overhead when compared with the DQN approach in Section 3.1 and

the DDPG-based approach in Section 3.2. In addition, unlike the previous approaches considering either discrete action space or continuous action space solely, the two-layer approach obtains the variables in their own domains without either approximating the hybrid space by concretization or relaxing it into a continuous set. As a result, the two-layer approach would achieve higher utilities than the others, as exemplified in the experiments.

Specifically, we propose to use a DQN-based algorithm in the upper layer to resolve the beam selection problem in its own discrete action space. When compared with that given in Section 3.1, this algorithm considers locations $\mathbf{L}$ and beamforming vectors $\mathbf{F}$ only, leading to a reduced DQN model whose state at time t is represented by $s^{(t)} = \left\{\mathbf{L}^{(t)}, \mathbf{F}^{(t)}\right\}$, and the action $a^{(t)}$ is selected from $\hat{\mathbf{A}}$ (here including only $\hat{\mathbf{A}}_F$) modified to take into account also the case of no changes. That is, each $\hat{A}_{f_i}^{(t)} \in a^{(t)}$ selected from $\hat{\mathbf{A}}_F$ can now be anyone in $\{-1, 0, +1\}$ instead of ± 1, in which 0 implies no changes on the previous beam selection. When the modification integrates with the lower layer, the two-layer hybrid MRA training algorithm has results as shown in Algorithm 3 along with its flowchart shown in Figure 2. Similar to Algorithms 1 and 2, the training algorithm would take the parameters for the utility, the hyperparameters for the learning algorithm, and the parameters for the game-based method, as the input, while producing a learned DQN model as the output that can online decide P_i, θ_i, and f_i, $\forall i$, for the optimization problem in (7) afterwards. Apart from the input and output, its main steps are summarized as follows:

Algorithm 3 The two-layer hybrid MRA training algorithm.

(Input) $\lambda_i, \mu_i, \nu_i, \forall i$, batch size η, learning rate α, minimum exploration rate ϵ_{min}, discount factor ζ, exploration decay rate d, and converge threshold ϱ;
(Output) Learned DQN to decide $P_i, \theta_i, f_i, \forall i$, for (7);
(Upper-layer DQN-based learning:)
Initialize action $a^{(0)}$ and replay buffer $D = \emptyset$;
for episode = 1 to $\mathcal{M}$ **do**
  Initialize state $s^{(0)}$;
  for time $t = 1$ to $\mathcal{N}$ **do**
    Observe current state $s^{(t)} = \left\{\mathbf{L}^{(t)}, \mathbf{F}^{(t)}\right\}$;
    $\epsilon = \max(\epsilon \cdot d, \epsilon_{min})$;
    if random number $r < \epsilon$ **then**
      Select $a^{(t)}$ from $\hat{\mathbf{A}}_F$ at random;
    else
      Select $a^{(t)} = \arg\max_{a'} Q^*(s^{(t)}, a', \omega)$;
    end if
    Observe next state s';
    (Lower-layer game-theory-based iteration:)
    for each link i **do**
      for iteration $k = 1$ to $\mathcal{K}$ **do**
        Update $P_i[k]$ with (47);
        Update $\theta_i[k]$ with (48);
        if $|U_i[k] - U_i[k-1]| \leq \varrho$ **then**
          $k' = k$; break;
        end if
      end for
      $k^* = \min\{k', \mathcal{K}\}$;
      $P_i^{(t)} = P_i[k^*]$; $\theta_i^{(t)} = \theta_i[k^*]$;
    end for
    Determine $U_i^{(t)}$ based on $P_i^{(t)}$ and $\theta_i^{(t)}$ in the lower layer, and $f_i^{(t)}$ in the upper layer, $\forall i$;
    Store transition $(s^{(t)}, a^{(t)}, \mathfrak{r}^{(t)}, s')$ in D;
    Select η random samples $(s^{(j)}, a^{(j)}, \mathfrak{r}^{(j)}, s^{(j+1)})$ from D;
    Calculate $\hat{Q}(s^{(j)}, a^{(j)}, \omega')$ and perform SGD to find the optimal weight of DNN, ω^*;
    Update $\omega = \omega^*$ for DQN in the upper layer;
    $s^{(t)} = s'$;
  end for
end for

- Observe state $s^{(t)}$ at time t for beam section.
- Select an optimal action from $a^{(t)}$ at time step t.
- Given selected beamforming vectors $\mathbf{F}^{(t)}$, obtain transmit powers $\mathbf{P}^{(t)}$ and splitting ratios $\mathbf{\Theta}^{(t)}$ through the game-theory-based iterative method in the lower layer.
- Assess the impact on data rate r_i, energy harvesting E_i, and transmit power P_i, for all links i.
- Reward the action at time t as $U_i(\mathbf{P}^{(t)}, \theta_i^{(t)}, \mathbf{F}^{(t)}), \forall i$, based on the impact assessed.
- Train DQN with the system utility $U^{(t)}$ obtained.

After the training or learning period, say T, the trained DQN from Algorithm 3 would be used to observe the following state $s^{(t)} = \left\{\mathbf{L}^{(t)}, \mathbf{F}^{(t)}\right\}, t > T$, evaluate utility U_i with the given parameters λ_i, μ_i, and ν_i, and then take action $a^{(t)}$ to decide P_i, θ_i, and $f_i, \forall i$, for the system in the testing process.

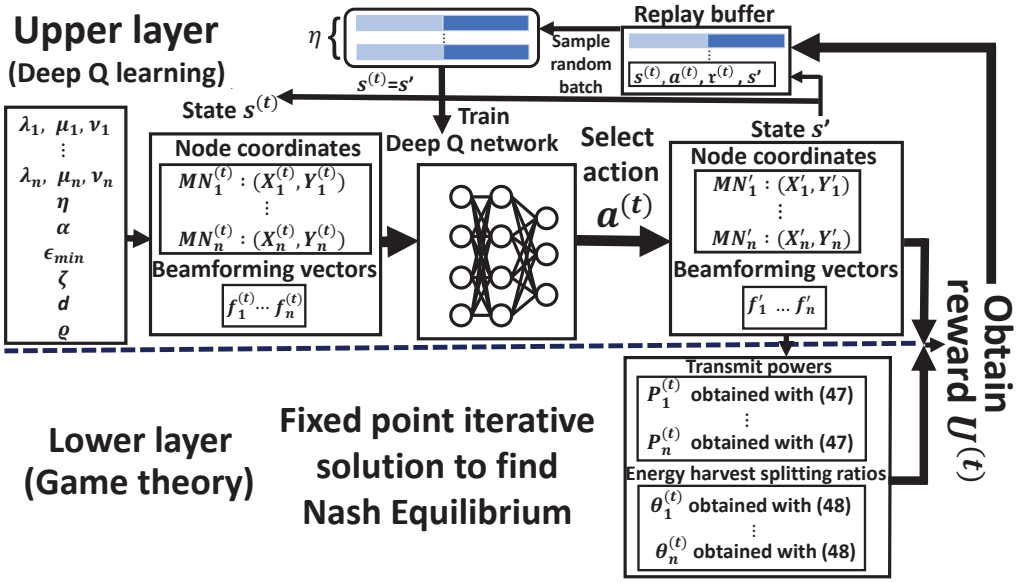

Figure 2. Flowchart of the two-layer hybrid MRA training algorithm. In the upper half, the input and the state (corresponding to line 1 and line 3 in Algorithm 3) are shown by the first box and the second box, respectively (from left to right). After selection (lines 10–14), the new state is shown by the fourth box. In the bottom half, the lower-layer iterations (lines 17–27) are exhibited with a box showing the equations involved. The two halves then cooperatively produce the reward (line 28) shown in the rightmost side toward the remaining boxes at the top denoting the following steps in Algorithm 3.

4.5. Time Complexity

Next, we show the time complexity for each of these algorithms before revealing their performance differences in the next section. Specifically, let the number of episodes be $\mathcal{M}$, and the number of time-steps per episode be $\mathcal{N}$. Assuming that the Q-learning network in Algorithm 1 has J fully connected layers, the time complexity with regard to the number of (floating point) operations in this algorithm would be $O(\sum_{j=0}^{J-1} u_j u_{j+1})$ based on the analysis in [32], where u_j denotes the unit number in the jth layer, and u_0 is the input state size. In each time-step of an episode, there may be other operations such as the random selection of an action in line 10 not involving the neural network, which could be ignored when compared with the former for the analysis. Thus, taking the nesting for loops (the outer is episode loop and the inner is time-step loop) into account, we have its worst-case time complexity as $O(\mathcal{MN}\sum_{j=0}^{J-1} u_j u_{j+1})$.

Apart from training, DDPG also involves a normalization process whose time complexity could be denoted by $\mathcal{T}(s)$, where $\mathcal{T}(s)$ is the number of the variables in the state

set. In addition, the actor and critic networks of DDPG in Algorithm 2 are assumed to have J and K fully connected layers, respectively. According to [32], the time complexity with respect to these networks in the training algorithm would be $O(\sum_{j=0}^{J-1} u_{actor,j}u_{actor,j+1} + \sum_{k=0}^{K-1} u_{critic,k}u_{critic,k+1})$, where $u_{actor,i}$ and $u_{critic,i}$ denote the unit number in the ith layer with respect to the actor network and the critic network, respectively. Then, by taking the nesting loops into account as well, we have the overall time complexity of this algorithm as $O(\mathcal{MN}(\sum_{j=0}^{J-1} u_{actor,j}u_{actor,j+1} + \sum_{k=0}^{K-1} u_{critic,k}u_{critic,k+1}))$.

Finally, let the number of links be n and the number of iterations per link be $\mathcal{K}$ in addition to $\mathcal{M}$ and $\mathcal{N}$ given previously. As the two-layer hybrid training algorithm involves the lower-layer game-theory-based iterations, the overall time complexity of Algorithm 3 would be $O(\mathcal{MN}n\mathcal{K}\sum_{j=0}^{J-1} u_{Q,j}u_{Q,j+1})$, where $u_{Q,j}$ denotes the unit number in the jth layer with respect to the DQN neural network in this algorithm. Note that, although there are additional $n\mathcal{K}$ iterations for the lower layer, the input state size $u_{Q,0}$ is $\left|\left\{\mathbf{L}^{(0)}, \mathbf{F}^{(0)}\right\}\right|$ that could be much smaller than $u_0 = \left|\left\{\mathbf{L}^{(0)}, \mathbf{P}^{(0)}, \mathbf{\Theta}^{(0)}, \mathbf{F}^{(0)}\right\}\right|$ in the single-layer Algorithm 1 with DQN, while $u_{Q,j} = u_j$, $1 \leq j \leq J$, is considered. In addition, it requires no normalization process and has the computational overhead on its neural network lower than that of $O(\sum_{j=0}^{J-1} u_{actor,j}u_{actor,j+1} + \sum_{k=0}^{K-1} u_{critic,k}u_{critic,k+1})$ on the two different types of neural networks in Algorithm 2.

5. Numerical Experiments

In this section, we conduct simulation experiments to evaluate the proposed two-layer approach and compare it with the single-layer approaches also introduced. To this end, we first present the simulation setup adopted and the parameters involved. Then, we show the performance differences between the two-layer hybrid MRA algorithm based on game theory and deep reinforcement learning, and the single-layer counterparts based on the conventional deep reinforcement learning models (DQN and DDPG).

5.1. Simulation Setup

With the network model and the channel model introduced in Section 2, we conduct MNs to be uniformly distributed in the simulated cellular network and let them move at a speed of v = 2 km/h on average with log-normal shadow fading as well as small-scale fading. In this environment, the cell radius is set to $\hat{r}$ and the distance between sites or BSs is considered to be 1.5 $\hat{r}$, in which MNs can experience a probability of line of sight, P_{los}, on the signals from BSs. For easy reference, the important parameters for the radio environment including those not shown above are summarized in Table 1.

Table 1. Important radio environment parameters.

Parameter	Value
Maximum transmit power (P_{max})	40 W (46 dBm)
Minimum transmit power (P_{min})	1 W (30 dBm)
Probability of light of sight (P_{los})	0.7
Cell radius ($\hat{r}$)	150 m
Distance between sites (BSs)	225 m
Antenna gain	3 dBi
Mobile node (MN) antenna gain	0 dBi
Number of multipaths	4
MN movement speed on average (v)	2 km/h
Number of transmit antennas of BS	$\{4, 8, 16, 32\}$
Downlink frequency band	28 GHz

Apart from the parameters for radio, the converge threshold ϱ is set to 10^{-5} for the two-layer algorithm, and the hyperparameters for the deep reinforcement learning models

are tabulated in Table 2. For example, in the DQN for the single-layer approach, the state $s^{(t)}$ at time t is denoted by

$$\left\{X_i^{(t)}, Y_i^{(t)}, X_j^{(t)}, Y_j^{(t)}, P_i^{(t)}, P_j^{(t)}, \theta_i^{(t)}, \theta_j^{(t)}, f_i^{(t)}, f_j^{(t)}\right\}$$

which corresponds to the size of state, 10, listed in this table. In addition, as introduced in Section 3.1, a $\pm$ 1 dB offset representation is considered for PSF1, and the number of power levels is set here as 9 for PSF2 to construct their power sets $\mathcal{P}^1$ and $\mathcal{P}^2$, respectively. Furthermore, a $\pm$ 0.05 offset representation, and a set of 11 values, $\{0, 0.1, \cdots, 1\}$ with step size of 0.1, are also conducted as the power splitting ratio sets $\boldsymbol{\Theta}$ for PSF1 and PSF2, respectively. Nevertheless, the size of action is 64 according to the binary decisions defined in (18), despite PSF1 or PSF2 in DQN. Apart from the above, for the two-layer approach, the DQN for the upper layer only considers the beamforming vectors $\mathbf{F}$ in addition to the locations $\mathbf{L}$, which reduces the size of state to 6. Moreover, as it considers $\{-1, 0, +1\}$ instead of ± 1 for the actions, the size of action becomes 9. Despite these differences, the other hyperparameters of DQN are the same for both single- and two-layer approaches. Finally, the hyperparameters for DDPG are chosen to reflect its performance on average with a reasonable time complexity to execute, and a codebook $\mathcal{F}$ with 4, 8, 16, and 32 elements or vectors, respectively, to correspond to the different numbers of antennas in the radio environment is considered for all the algorithms involved.

Table 2. Reinforcement learning parameters.

	Parameter	**Value**
	DQN:	
	Discount factor (ζ)	0.995
	Learning rate (α)	0.01
	Initial exploration rate (ϵ)	1.0
	Minimum exploration rate (ϵ_{min})	0.1
	Exploration decay rate (d)	0.9995
	Size of state ($\|s\|$)	10
	Size of action ($\|a\|$)	64
	Replay buffer size ($\|D\|$)	2000
	Batch size (η)	256
	DDPG:	
	Actor learning rate (α_a)	0.001
	Critic learning rate (α_c)	0.002
	Replay buffer size ($\|D\|$)	10000
	Exploration decay rate (d)	0.9995
	Batch size (η)	32
	Scale factors (ς_1, ς_2, ς_3)	1
	Discount factor (ζ)	0.9
	Soft update parameter (τ)	0.01
	DQN for two-layer:	
	Size of state ($\|s\|$)	6
	Size of action ($\|a\|$)	9
	The same parameters for the single-layer DQN	

Given that, we conduct 50 experiments with different seeds for all the algorithms under comparison. For each of these experiments, there are 400 training episodes or epochs in total. At the beginning of each episode, MNs are randomly located in the simulated network, which then move at speed v in 500 time slots per episode. Afterward, with the trained $(\mathbf{P}, \boldsymbol{\Theta}, \mathbf{F})$ from these algorithms, we conduct another 100 episodes with MNs

randomly located at the beginning as well to obtain the averaged utility, data rate, energy harvesting, and power consumption to validate the parameters obtained with the different algorithms. Specifically, each 100 testing episodes of an experiment produce a mean value, and each averaged metric shown in the following figures denotes the average of these mean values from the 50 experiments. Note that, since DDPG is trained with normalized variables as shown in (32), in the testing process, we also have to preprocess these inputs.

5.2. Performance Comparison

Given the environment, we compare the proposed two-layer MRA algorithm aided by game theory with the single-layer MRA algorithms based solely on DQN and DDPG also introduced. To see their performance differences, we conduct two sets of experiments from different aspects; the first focuses on the number of antennas, M, and the second on the power cost ν_i. Given that, in Figures 3–5 to be shown for the comparison results, the legends of "two-layer", "single-layer with DDPG", "single-layer with DQN of PSF1", and "single-layer with DQN of PSF2" exhibited therein represent the two-layer MRA algorithm, the single-layer DDPG-based MRA algorithm, the single-layer DQN-based MRA algorithm with PSF1, and the single-layer DQN-based MRA algorithm with PSF2 introduced in this work, respectively.

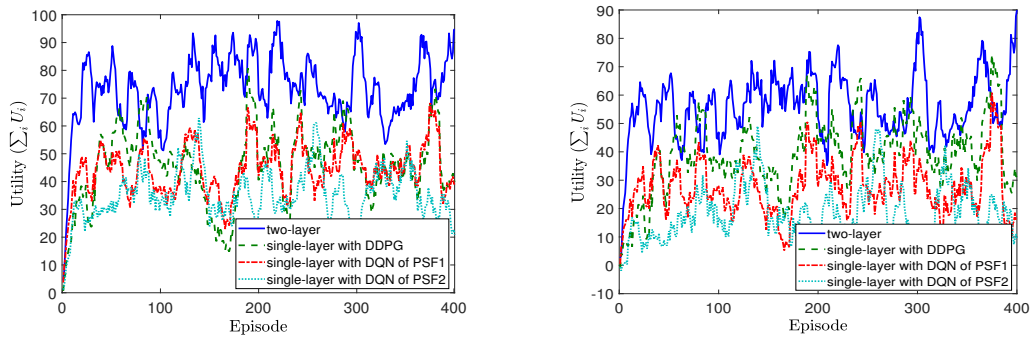

Figure 3. Utilities obtained during training periods upon (**a**) $M = 32$, and (**b**) $M = 4$.

5.2.1. Impacts of Antennas

In the first experiment set, four numbers of transmit antennas, $M \in \{4, 8, 16, 32\}$, in BS are examined while fixing $\lambda_i = 10$, $\mu_i = 1$, and $\nu_i = 1$, $\forall i$. Due to similar trends to be given, in Figure 3, we exemplify the utilities obtained during the training periods in two experiment instances with the highest and the lowest number of transmit antennas, 32 and 4, respectively. It can be seen easily from the two sub-figures that the utility that resulted from the two-layer MRA algorithm is higher than those from the single-layer counterparts during the training periods, despite the number of antennas, on average. In addition, it can also be observed that, with the continuous action space, DDPG could outperform DQN in general, despite the power state formulations (PSF1 and PSF2) of the latter. Finally, we can see that, with a $\pm$ 1 dB offset representation, PSF1 of DQN would result in a greater number of states on the transmit power than PSF2 equipped with a limited number of quantized levels, which could eventually lead to a better performance on the utility in the long term.

Next, we show the performance differences among the averaged metrics on utility, data rate, energy harvesting, and power consumption obtained by the testing process on $(\mathbf{P}, \mathbf{\Theta}, \mathbf{F})$ resulting from these algorithms. As shown in Figure 4, the two-layer MRA algorithm outperforms the single-layer counterparts on all the performance metrics except the energy harvesting, despite the number of antennas, M. In particular, in terms of the averaged utilities resulting from all different M, the two-layer MRA algorithm can achieve up to 2.3 times higher value than the single-layer DQN of the PSF2 algorithm. Despite the utility, as the resulting energy harvesting has relatively smaller values to impact the

overall utility, a lower (resp. higher) value of this metric represented in the log scale is still possible and its impact would be compensated by a higher (resp. lower) value of power consumption, data rate, or both, which eventually leads to the overall utility to increase as M increases. For example, the highest utilities which are obtained by the two-layer MRA algorithm (as shown in Figure 4a) are mainly contributed by the highest data rates (as shown in Figure 4b) and the lowest power consumption (as shown in Figure 4d), which are all resulting from the two-layer algorithm, despite the energy harvesting of this algorithm to be slightly fluctuated as M increases and lower than that from the single-layer counterparts (as shown in Figure 4c).

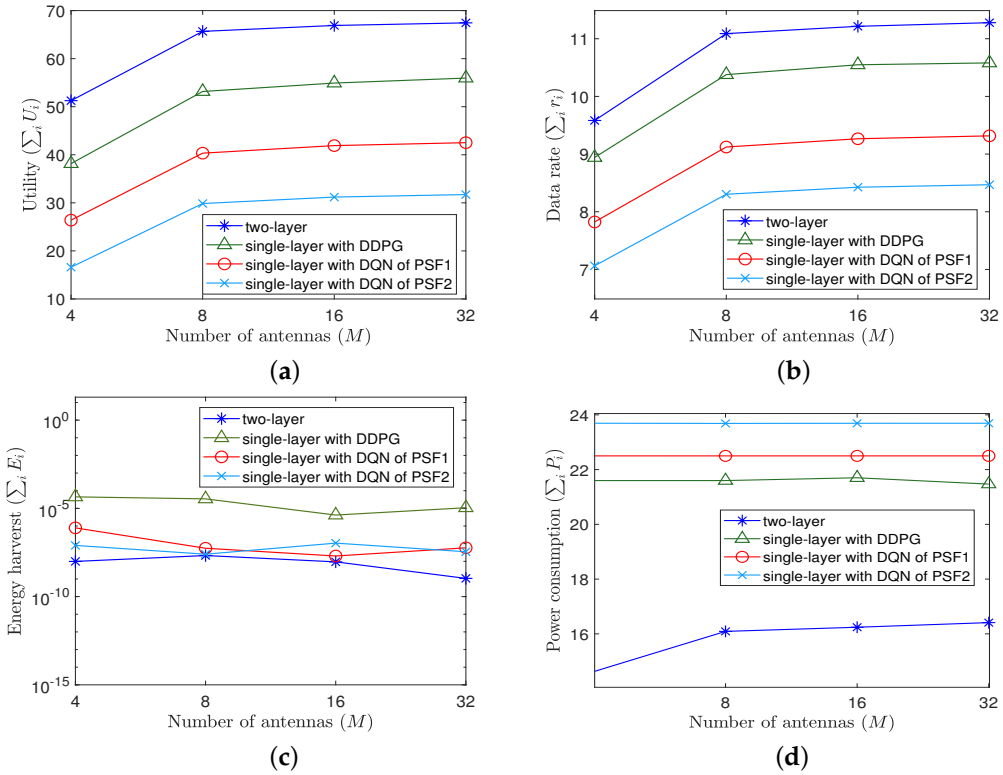

Figure 4. Impacts of varying the number of antennas (M) upon (**a**) utility, (**b**) data rate (bps), (**c**) energy harvesting (J), and (**d**) power consumption (W).

In addition, as no previous works exactly consider the same system formulations and metrics presented here, it is hard to directly compare this work with the others such as [27,51] which consider only **P**, **F**, or both, for their data transmissions without the capability of energy harvesting. However, even without the capability, we could still consider the DRL algorithm in [51] with only **P** to see the possible performance differences between ours and the conventional approaches. Specifically, with $M = 32$, the comparison results are summarized in Table 3. As shown readily, without the power split for energy harvesting, the DRL algorithm can obtain the highest data rate as an upper bound here, as expected. In comparison, the two-layer algorithm can achieve almost the same data rate while harvesting the energy with the lowest power consumption. Similarly, the single-layer algorithms can enjoy the energy harvesting with similar power consumption, but they may obtain lower data rates when splitting their powers to harvest energy and send data simultaneously.

5.2.2. Impacts of Pricing Strategy

From the utility function defined by (19), we can see that the unit power cost ν_i actually plays a crucial role in the non-cooperative game model, and would have a strong impact on the performance of joint optimization and the Nash equilibrium. Thus, in the final set of

experiments, we propose a simple pricing strategy for the base station to determine ν_i on the basis of social utility maximization and to control the transmit power of link so that its value can be located within the feasible range $[P_{min}, P_{max}]$ for the high performance of this algorithm to be realized by the social utility maximization.

Table 3. Performance comparison with $M = 32$.

Method	Data Rate	Energy Harvesting	Power Consumption
DRL	11.32910	0	22.51510
two-layer	11.26969	8.164853×10^{-9}	16.40005
single-layer with DDPG	10.58339	1.062941×10^{-5}	21.34165
single-layer with DQN of PSF1	9.31607	5.809001×10^{-8}	22.50100
single-layer with DQN of PSF2	8.46842	3.477011×10^{-8}	23.69319

Specifically, let the desired transmit power be P_i^d, and, according to the fixed point formulation in (44), we have

$$P_i^d = \frac{\lambda_i}{\nu_i - \frac{\mu_i |h_{i,i} f_i|^2}{\hat{R}_i + P_i^d |h_{i,i} f_i|^2}} - \frac{\hat{R}_i}{(1-\theta_i)|h_{i,i} f_i|^2} \tag{49}$$

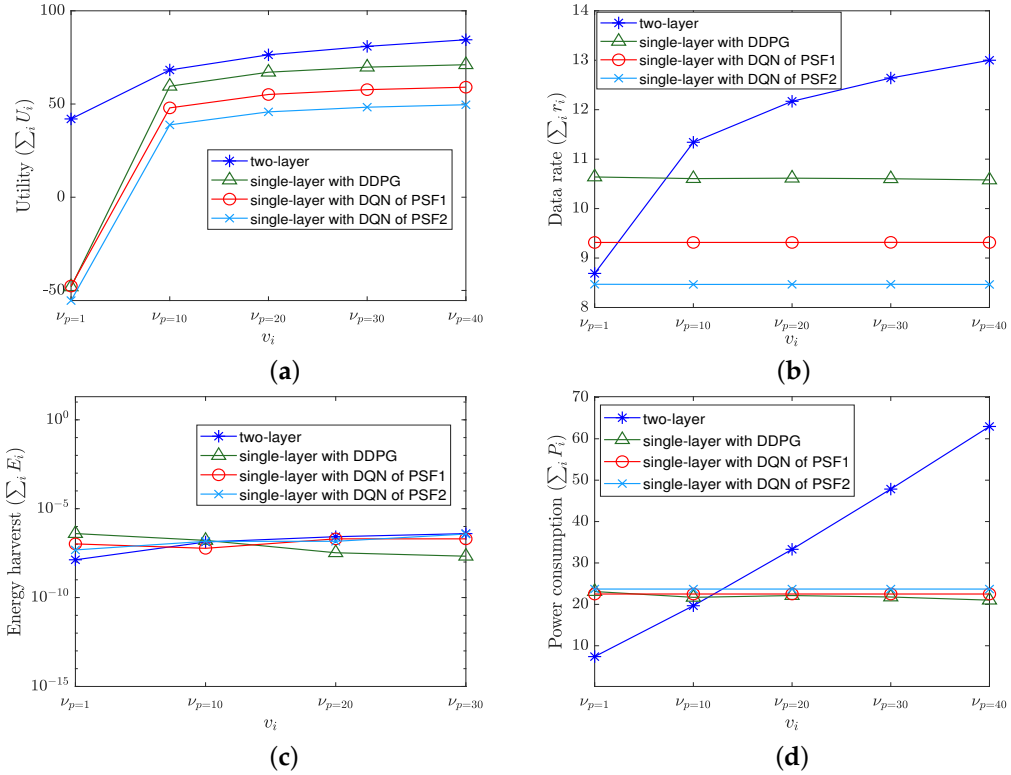

Figure 5. Impacts of the pricing strategy upon (**a**) utility, (**b**) data rate (bps), (**c**) energy harvesting (J), and (**d**) power consumption (W).

Given that, the desired power cost $\nu_{P_i^d}$ can be obtained by

$$\nu_{P_i^d} = \frac{\mu_i |h_{i,i} f_i|^2}{\hat{R}_i + P_i^d |h_{i,i} f_i|^2} + \frac{\lambda_i (1-\theta_i)|h_{i,i} f_i|^2}{P_i^d (1-\theta_i) + \hat{R}_i} \tag{50}$$

Accordingly, the two-layer hybrid MRA algorithm is slightly modified to dynamically adjust ν_i instead of using a fixed $\nu_i, \forall i$, as an input of the algorithm. To be more specific, the sketch of this modification is given in Algorithm 4, wherein the modified three statements

showing their calculations (50), (47) and (48), respectively, are highlighted with bold italic font, in addition to the fact that the input does not include ν_i now. For the comparison, the pricing strategy is also applied to Algorithms 1 and 2 by replacing the input ν_i with $\nu_{P_i^d}$ dynamically adjusted by using (50) as well after observing the next state $\boldsymbol{S}'$ carried out in the corresponding steps in these algorithms.

Algorithm 4 The two-layer hybrid MRA training algorithm with the pricing strategy.

(Input) $\lambda_i, \mu_i, \forall i, \cdots$;
...
for episode = 1 to $\mathcal{M}$ **do**
  for time $t = 1$ to $\mathcal{N}$ **do**
    ...
    Observe next state S';
    Obtain $\nu_{P_i^d}, \forall i$, **by using (50)**
    for each link i **do**
      for iteration $k = 1$ to $\mathcal{K}$ **do**
        Update $P_i[k]$ **by using (47) with** $\nu_i = \nu_{P_i^d}, \forall i$;
        Update $\theta_i[k]$ **by using (48) with** $\nu_i = \nu_{P_i^d}, \forall i$;
        ...
      end for
    end for
    ...
  end for
end for

Here, following the same setting P_{min} = 1 W and P_{max} = 40 W, we sample the feasible range at $\{1\text{ W}, 10\text{ W}, 20\text{ W}, 30\text{ W}, 40\text{ W}\}$ as P_i^d to obtain $\nu_{P_i^d}$ with (50) while fixing $\lambda = 10$, $\mu = 1$, and $M = 32$, and conduct these algorithms to output the performance metrics averaged to be compared. The results are now summarized in Figure 5, showing that the two-layer algorithm outperforms the others in terms of the utility. In particular, although it may have lower data rates when $\nu_{P=1}$ (denoting $\nu_{P_i^d}$ obtained by $P_i^d = 1$ W), and higher power consumption when $\nu_{P>10}$ (denoting $\nu_{P_i^d}$ with $P_i^d > 10$ W), the increasing trend of these resulting metrics would still lead to a utility higher than the others and the resulting utility would increase as $\nu_{P_i^d}$ increases. Similarly, as the energy harvesting has relatively smaller values to impact the system as noted before, its small fluctuations from the different algorithms do not alter the increasing trend of utility in the final experiment set as well.

6. Conclusions

In this work, we sought to maximize the utility that can make an optimal trade-off among data rate and energy harvesting while balancing the cost of power consumption in multi-access wireless networks with base stations having multi-antennas. Given the capability of selecting beamforming vectors from a finite set, adjusting transmit powers, and deciding power splitting ratios for energy harvesting, the wireless networks developed toward the future generation (beyond 5G or B5G) are expected to achieve the extreme performance requirements that can only be satisfied by an optimal solution to be possibly found through an exhaustive search.

To meet the expectation, we have shown in this work how to design DRL-based approaches operated in a single layer to jointly solve for power control, beamforming selection, and power splitting decision, and approach the optimal trade-off among the performance metrics without an exhaustive search in the action space that resulted. Furthermore, we have shown how to incorporate a data-driven DRL-based technique and a model-driven game-theory-based algorithm to form a two-layer iterative approach to resolve the NP-hard MRA problem in the wireless networks. Specifically, we have shown that, by taking benefits from both data-driven and model-driven methods, the proposed two-layer MRA algorithm can outperform the single-layer counterparts which rely only on

the data-driven DRL-based algorithms. Here, the single-layer algorithms could represent the conventional DRL methods extended to have the energy harvesting capability. As shown readily in the experiments, the conventional DRL method and the single-layer algorithms would not provide a good performance trade-off on the metrics considered. That is, the overall utilities reflecting the trade-off from the single-layer algorithms have been shown to be lower than that from the two-layer approach. In contrast, by collaborating between DRL and game theory, the two-layer approach has been shown to achieve better trade-off among the data rate and the energy harvesting while balancing the cost of power consumption, reflecting on the higher utilities obtained. Specifically, in the simulation experiments, we have exemplified the performance differences of these algorithms in terms of data rate, energy harvesting, and power consumption, verified the feasibility of the three parameters in the utility function, and examined the pricing strategy proposed that can dynamically adjust the transmit power of the link to locate its value within the feasible range for the high performance of the two-layer algorithm to be obtained by the social utility maximization.

From the viewpoint of social utility maximization, our pricing strategy had been shown to give this system the leverage to select beamforming vectors, transmit powers, and power split ratios by properly adjusting the power costs. Finally, inspired by the related works on multi-agent DRL, we would aim to develop further collaborating schemes that can reduce the overhead caused by different optimization methods even under the non-stationary environment brought by a multi-agent setting, as our future work.

Author Contributions: J.L. the main research idea, software implementation, validation, and manuscript preparation; C.-H.R.L. research idea discussion, review, and manuscript preparation; Y.-C.H. research idea discussion, edit, and manuscript preparation; P.K.D. research idea discussion and manuscript preparation. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Ministry of Science and Technology, Republic of China, under Grant MOST 110-2221-E-126-001.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

Table A1. Important symbols for the proposed approaches in this work.

Name	Description	Name	Description
$\mathbf{P}, \mathbf{\Theta}, \mathbf{F}$	sets of transmit powers, splitting ratios, and beamforming vectors, respectively	P_i, θ_i, f_i	transmit power, splitting ratio, and beamforming vector for link i, respectively
$\mathbf{L}$	set of locations	$\hat{L}$	loss function
$\widetilde{S}$	a finite set of states, $\{s_1, s_2, \cdots, s_{\mathtt{m}}\}$	$s^{(t)}$	state at time t, denoted by $\left\{\mathbf{L}^{(t)}, \mathbf{P}^{(t)}, \mathbf{\Theta}^{(t)}, \mathbf{F}^{(t)}\right\}$
$\widetilde{A}$	a finite set of actions, $\{a_1, a_2, \cdots, a_{\mathtt{n}}\}$	$\hat{\mathbf{A}}$	a set of binary variables, where $\hat{\mathbf{A}}_P$, $\hat{\mathbf{A}}_\Theta$, and $\hat{\mathbf{A}}_F$ correspond to those for $\mathbf{P}$, $\mathbf{\Theta}$, and $\mathbf{F}$, respectively.

Table A1. *Cont.*

Name	Description	Name	Description
$\widetilde{\boldsymbol{R}}$	a finite set of rewards, where $\widetilde{R}(s,a,s')$ is the function to provide reward $\mathfrak{r}$ at state $s \in \widetilde{\boldsymbol{S}}$, action $a \in \widetilde{\boldsymbol{A}}$, and next state s'	$\widetilde{\boldsymbol{P}}$	a finite set of transition probabilities, where $\widetilde{P}_{ss'(a)} = p(s'\|s,a)$ is the transition probability at state s taking action a to migrate to state s'
$\pi^*(s)$	optimal policy at state s	$V^{\pi}(s)$	value function for the expected value to be obtained by policy π from state $s \in S$
$V^*(s)$	optimal action at state s	$Q^{\pi}(s,a)$	action–value function representing the expected reward starting from state s and taking action a from policy π
Q^{π^*}	optimal policy for the (optimal) action–value function $Q^*(s,a) = \max_{\pi} Q^{\pi}(s,a)$	$Q(s_t,a_t)$	action–value (Q) function at time t
$\hat{Q}(s_t,a_t,\omega')$	approximated action–value (Q) function with the weight of DNN, ω', at time t	$\mathcal{F}$	beamforming codebook
$\mathcal{P}_i^1$	a set of transmit powers for link i in PSF1	$\mathcal{P}^2$	a set of transmit powers for all links in PSF2
$U_i(\mathbf{P}^{(t)},\theta_i^{(t)},\mathbf{F}^{(t)})$	reward for link i at time t, including data rate $r_i(\mathbf{P}^{(t)},\theta_i^{(t)},\mathbf{F}^{(t)})$ and energy harvest $E_i(\mathbf{P}^{(t)},\theta_i^{(t)},\mathbf{F}^{(t)})$	$(s^{(t)},a^{(t)},\mathfrak{r}^{(t)},s')$	transition at time t, where $\mathfrak{r}^{(t)} = U^{(t)}$ that is the system utility at this time step
α, α_a, α_c	learning rate, the (learning) rate specific to actor network, and the (learning) rate specific to critic network	ϵ, ϵ_{min}	exploration rate (probability) and its minimum requirement
ζ	discount factor	τ	soft update parameter
d	exploration decay rate	D	replay buffer
η	batch size	ϱ	converge threshold for the fixed point iteration
$Q_a(s;\omega_a)$, $Q_{a'}(s;\omega_{a'})$	output of actor network (online and target, respectively)	$Q(s,a;\omega_c)$, $\tilde{Q}(s,a;\omega_{c'})$	output of critic network (online and target, respectively)
α_i, μ_i, ν_i	parameters for data rate, energy harvesting, and power consumption, respectively, for link i	ς_1, ς_2, ς_3	scale factors for normalization of DDPG at time t
$a^{(t)^*}$	deterministic action of DDPG at time t, wherein $A_P^{(t)^*}$, $A_\Theta^{(t)^*}$, and $A_F^{(t)^*}$ correspond to those for transmit power, split ratio and beamforming vector	$\tilde{P}_i^{(t)}, \tilde{\theta}_i^{(t)}, \tilde{f}_i^{(t)}$	variables for normalization of DDPG at time t
R_i	total received power at link i for the fixed point iteration	$\hat{R}_i$	auxiliary variable at link i for the fixed point iteration
P_i^d	desired transmit power at link i for the pricing strategy	$\nu_{P_i^d}$	desired power cost at link i for the pricing strategy

References

1. Zhang, K.; Mao, Y.; Leng, S.; Zhao, Q.; Li, L.; Peng, X.; Pan, L.; Maharjan, S.; Zhang, Y. Energy-Efficient Offloading for Mobile Edge Computing in 5G Heterogeneous Networks. *IEEE Access* **2016**, *4*, 5896–5907. [CrossRef]
2. Hewa, T.; Braeken, A.; Ylianttila, M.; Liyanage, M. Multi-Access Edge Computing and Blockchain-based Secure Telehealth System Connected with 5G and IoT. In Proceedings of the GLOBECOM 2020—2020 IEEE Global Communications Conference, Taipei, Taiwan, 7–11 December 2020; pp. 1–6.

3. Chen, F.; Wang, A.; Zhang, Y.; Ni, Z.; Hua, J. Energy Efficient SWIPT Based Mobile Edge Computing Framework for WSN-Assisted IoT. *Sensors* **2021**, *21*, 4798. [CrossRef] [PubMed]
4. Chae, S.H.; Jeong, C.; Lim, S.H. Simultaneous Wireless Information and Power Transfer for Internet of Things Sensor Networks. *IEEE Internet Things J.* **2018**, *5*, 2829–2843. [CrossRef]
5. Masood, Z.A.; Choi, Y. Energy-efficient optimal power allocation for swipt based iot-enabled smart meter. *Sensors* **2021**, *21*, 7857. [CrossRef]
6. Liu, J.-S.; Lin, C.-H.R.; Tsai, J. Delay and energy trade-off in energy harvesting multi-hop wireless networks with inter-session network coding and successive interference cancellation. *IEEE Access* **2017**, *5*, 544–564. [CrossRef]
7. Tran, T.-N.; Voznak, M. Switchable Coupled Relays Aid Massive Non-Orthogonal Multiple Access Networks with Transmit Antenna Selection and Energy Harvesting. *Sensors* **2021**, *21*, 1101. [CrossRef]
8. Luo, Z.-Q.; Zhang, S. Dynamic Spectrum Management: Complexity and Duality. *IEEE J. Sel. Top. Signal Process.* **2008**, *2*, 57–73. [CrossRef]
9. Boccardi, F., Jr.; Heath, R.W.; Lozano, A.; Marzetta, T.L.; Popovski, P. Five disruptive technology directions for 5G. *IEEE Commun. Mag.* **2014**, *52*, 74–80. [CrossRef]
10. Li, Y.; Luo, J.; Xu, W.; Vucic, N.; Pateromichelakis, E.; Caire, G. A Joint Scheduling and Resource Allocation Scheme for Millimeter Wave Heterogeneous Networks. In Proceedings of the 2017 IEEE Wireless Communications and Networking Conference (WCNC), Francisco, CA, USA, 19–22 March 2017; pp. 1–6.
11. Yang, Z.; Xu, W.; Xu, H.; Shi, J.; Chen, M. User Association, Resource Allocation and Power Control in Load-Coupled Heterogeneous Networks. In Proceedings of the 2016 IEEE Globecom Workshops (GC Wkshps), Washington, DC, USA, 4–8 December 2016; pp. 1–7.
12. Saeed, A.; Katranaras, E.; Dianati, M.; Imran, M.A. Dynamic femtocell resource allocation for managing inter-tier interference in downlink of heterogeneous networks. *IET Commun.* **2016**, *10*, 641–650. [CrossRef]
13. Coskun, C.C.; Davaslioglu, K.; Ayanoglu, E. Three-Stage Resource Allocation Algorithm for Energy-Efficient Heterogeneous Networks. *IEEE Trans. Veh. Technol.* **2017**, *66*, 6942–6957. [CrossRef]
14. Liu, R.; Sheng, M.; Wu, W. Energy-Efficient Resource Allocation for Heterogeneous Wireless Network With Multi-Homed User Equipments. *IEEE Access* **2018**, *6*, 14591–14601. [CrossRef]
15. Le, N.-T.; Tran, L.-N.; Vu, Q.-D.; Jayalath, D. Energy-Efficient Resource Allocation for OFDMA Heterogeneous Networks. *IEEE Trans. Commun.* **2019**, *67*, 7043–7057. [CrossRef]
16. Zhang, Y.; Wang, Y.; Zhang, W. Energy efficient resource allocation for heterogeneous cloud radio access networks with user cooperation and QoS guarantees. In Proceedings of the 2016 IEEE Wireless Communications and Networking Conference, Doha, Qatar, 3–6 April 2016; pp. 1–6.
17. Zou, S.; Liu, N.; Pan, Z.; You, X. Joint Power and Resource Allocation for Non-Uniform Topologies in Heterogeneous Networks. In Proceedings of the 2016 IEEE 83rd Vehicular Technology Conference (VTC Spring), Nanjing, China, 15–18 May 2016; pp. 1–5.
18. Zhang, H.; Du, J.; Cheng, J.; Long, K.; Leung, V.C.M. Incomplete CSI Based Resource Optimization in SWIPT Enabled Heterogeneous Networks: A Non-Cooperative Game Theoretic Approach. *IEEE Trans. Wirel. Commun.* **2017**, *17*, 1882–1892. [CrossRef]
19. Chen, X.; Zhao, Z.; Zhang, H. Stochastic Power Adaptation with Multiagent Reinforcement Learning for Cognitive Wireless Mesh Networks. *IEEE Trans. Mob. Comput.* **2012**, *12*, 2155–2166. [CrossRef]
20. Xu, C.; Sheng, M.; Yang, C.; Wang, X.; Wang, L. Pricing-Based Multiresource Allocation in OFDMA Cognitive Radio Networks: An Energy Efficiency Perspective. *IEEE Trans. Veh. Technol.* **2013**, *63*, 2336–2348. [CrossRef]
21. Jiang, Y.; Lu, N.; Chen, Y.; Zheng, F.; Bennis, M.; Gao, X.; You, X. Energy-Efficient Noncooperative Power Control in Small-Cell Networks. *IEEE Trans. Veh. Technol.* **2017**, *66*, 7540–7547. [CrossRef]
22. Zhang, H.; Song, L.; Han, Z. Radio Resource Allocation for Device-to-Device Underlay Communication Using Hypergraph Theory. *IEEE Trans. Wirel. Commun.* **2016**, *15*, 1. [CrossRef]
23. Zhang, R.; Cheng, X.; Yang, L.; Jiao, B. Interference-aware graph based resource sharing for device-to-device communications underlaying cellular networks. In Proceedings of the 2013 IEEE Wireless Communications and Networking Conference (WCNC), Shanghai, China, 7–10 April 2013; pp. 140–145.
24. Feng, D.; Lu, L.; Yuan-Wu, Y.; Li, G.Y.; Feng, G.; Li, S. Device-to-device communications underlaying cellular networks. *IEEE Trans. Commun.* **2013**, *61*, 3541–3551. [CrossRef]
25. Jiang, Y.; Liu, Q.; Zheng, F.; Gao, X.; You, X. Energy-Efficient Joint Resource Allocation and Power Control for D2D Communications. *IEEE Trans. Veh. Technol.* **2016**, *65*, 6119–6127. [CrossRef]
26. Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A.A.; Veness, J.; Bellemare, M.G.; Graves, A.; Riedmiller, M.; Fidjeland, A.K.; Ostrovski, G.; et al. Human-level control through deep reinforcement learning. *Nature* **2015**, *518*, 529–533. [CrossRef]
27. Meng, F.; Chen, P.; Wu, L.; Cheng, J. Power Allocation in Multi-User Cellular Networks: Deep Reinforcement Learning Approaches. *IEEE Trans. Wirel. Commun.* **2020**, *19*, 6255–6267. [CrossRef]
28. Nguyen, K.K.; Duong, T.Q.; Vien, N.A.; Le-Khac, N.-A.; Nguyen, M.-N. Non-Cooperative Energy Efficient Power Allocation Game in D2D Communication: A Multi-Agent Deep Reinforcement Learning Approach. *IEEE Access* **2019**, *7*, 100480–100490. [CrossRef]

29. Zhang, Y.; Kang, C.; Ma, T.; Teng, Y.; Guo, D. Power Allocation in Multi-Cell Networks Using Deep Reinforcement Learning. In Proceedings of the 2018 IEEE 88th Vehicular Technology Conference (VTC-Fall), Chicago, IL, USA, 27–30 August 2018; pp. 1–6.
30. Choi, J. Massive MIMO With Joint Power Control. *IEEE Wirel. Commun. Lett.* **2014**, *3*, 329–332. [CrossRef]
31. Zhang, Y.; Kang, C.; Teng, Y.; Li, S.; Zheng, W.; Fang, J. Deep Reinforcement Learning Framework for Joint Resource Allocation in Heterogeneous Networks. In Proceedings of the 2019 IEEE 90th Vehicular Technology Conference (VTC2019-Fall), Honolulu, HI, USA, 22–25 September 2019; pp. 1–6.
32. Qiu, C.; Hu, Y.; Chen, Y.; Zeng, B. Deep Deterministic Policy Gradient (DDPG)-Based Energy Harvesting Wireless Communications. *IEEE Internet Things J.* **2019**, *6*, 8577–8588. [CrossRef]
33. 3GPP. *Evolved Universal Terrestrial Radio Access (E-UTRA): Physical Layer Procedures (3GPP)*; ts 36.213, dec. 2015; 3GPP: Valbonne, France , 2015.
34. Kim, R.; Kim, Y.; Yu, N.Y.; Kim, S.-J.; Lim, H. Online Learning-Based Downlink Transmission Coordination in Ultra-Dense Millimeter Wave Heterogeneous Networks. *IEEE Trans. Wirel. Commun.* **2019**, *18*, 2200–2214. [CrossRef]
35. Song, Q.; Wang, X.; Qiu, T.; Ning, Z. An Interference Coordination-Based Distributed Resource Allocation Scheme in Heterogeneous Cellular Networks. *IEEE Access* **2017**, *5*, 2152–2162. [CrossRef]
36. Trakas, P.; Adelantado, F.; Zorba, N.; Verikoukis, C. A QoE-aware joint resource allocation and dynamic pricing algorithm for heterogeneous networks. In Proceedings of the GLOBECOM 2017—2017 IEEE Global Communications Conference, Singapore, 4–8 December 2017; pp. 1–6.
37. Simsek, M.; Bennis, M.; Guvenc, I. Learning Based Frequency- and Time-Domain Inter-Cell Interference Coordination in HetNets. *IEEE Trans. Veh. Technol.* **2014**, *64*, 4589–4602. [CrossRef]
38. Ghadimi, E.; Calabrese, F.D.; Peters, G.; Soldati, P. A reinforcement learning approach to power control and rate adaptation in cellular networks. In Proceedings of the 2017 IEEE International Conference on Communications (ICC), Paris, France, 21–25 May 2017; pp. 1–7.
39. Calabrese, F.D.; Wang, L.; Ghadimi, E.; Peters, G.; Hanzo, L.; Soldati, P. Learning Radio Resource Management in RANs: Framework, Opportunities, and Challenges. *IEEE Commun. Mag.* **2018**, *56*, 138–145. [CrossRef]
40. Sharma, S.; Darak, S.J.; Srivastava, A. Energy saving in heterogeneous cellular network via transfer reinforcement learning based policy. In Proceedings of the 2017 9th International Conference on Communication Systems and Networks (COMSNETS), Bengaluru, India, 4–8 January 2017; pp. 397–398.
41. Wei, Y.; Yu, F.R.; Song, M.; Han, Z. User Scheduling and Resource Allocation in HetNets With Hybrid Energy Supply: An Actor-Critic Reinforcement Learning Approach. *IEEE Trans. Wirel. Commun.* **2017**, *17*, 680–692. [CrossRef]
42. Liang, L.; Feng, G. A Game-Theoretic Framework for Interference Coordination in OFDMA Relay Networks. *IEEE Trans. Veh. Technol.* **2011**, *61*, 321–332. [CrossRef]
43. Lu, Y.; Xiong, K.; Fan, P.; Zhong, Z.; Ai, B.; Ben Letaief, K. Worst-Case Energy Efficiency in Secure SWIPT Networks with Rate-Splitting ID and Power-Splitting EH Receivers. *IEEE Trans. Wirel. Commun.* **2021**, *21*, 1870–1885. [CrossRef]
44. Xu, Y.; Li, G.; Yang, Y.; Liu, M.; Gui, G. Robust Resource Allocation and Power Splitting in SWIPT Enabled Heterogeneous Networks: A Robust Minimax Approach. *IEEE Internet Things J.* **2019**, *6*, 10799–10811. [CrossRef]
45. Zhang, R.; Xiong, K.; Lu, Y.; Gao, B.; Fan, P.; Ben Letaief, K. Joint Coordinated Beamforming and Power Splitting Ratio Optimization in MU-MISO SWIPT-Enabled HetNets: A Multi-Agent DDQN-Based Approach. *IEEE J. Sel. Areas Commun.* **2021**, *40*, 677–693. [CrossRef]
46. Omidkar, A.; Khalili, A.; Nguyen, H.H.; Shafiei, H. Reinforcement Learning Based Resource Allocation for Energy-Harvesting-Aided D2D Communications in IoT Networks. *IEEE Internet Things J.* **2022**, *7*, 4387–4394. [CrossRef]
47. Canese, L.; Cardarilli, G.; Di Nunzio, L.; Fazzolari, R.; Giardino, D.; Re, M.; Spanò, S. Multi-Agent Reinforcement Learning: A Review of Challenges and Applications. *Appl. Sci.* **2021**, *11*, 4948. [CrossRef]
48. Sutton, R.S.; Barto, A.G. Reinforcement Learning: An Introduction, 2nd ed.; MIT Press: Cambridge, MA, USA, 2018.
49. Goodfellow, J.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial nets. In Proceedings of the Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, Montreal, QC, Canada, 8–14 December 2014; pp. 2672–2680.
50. Perera, T.D.P.; Jayakody, D.N.K.; Sharma, S.K.; Chatzinotas, S.; Li, J. Simultaneous Wireless Information and Power Transfer (SWIPT): Recent Advances and Future Challenges. *IEEE Commun. Surv. Tutor.* **2017**, *20*, 264–302. [CrossRef]
51. Mismar, F.B.; Evans, B.L.; Alkhateeb, A. Deep Reinforcement Learning for 5G Networks: Joint Beamforming, Power Control, and Interference Coordination. *IEEE Trans. Commun.* **2019**, *68*, 1581–1592. [CrossRef]
52. Alkhateeb, A.; El Ayach, O.; Leus, G.; Heath, R.W. Channel Estimation and Hybrid Precoding for Millimeter Wave Cellular Systems. *IEEE J. Sel. Top. Signal Process.* **2014**, *8*, 831–846. [CrossRef]
53. Heath, R.W., Jr.; Gonzalez-Prelcic, N.; Rangan, S.; Roh, W.; Sayeed, A.M. An Overview of Signal Processing Techniques for Millimeter Wave MIMO Systems. *IEEE J. Sel. Top. Signal Process.* **2016**, *10*, 436–453. [CrossRef]
54. Schniter, P.; Sayeed, A. Channel estimation and precoder design for millimeter-wave communications: The sparse way. In Proceedings of the 2014 48th Asilomar Conference on Signals, Systems and Computers, Pacific Grove, CA, USA, 2–5 November 2014; pp. 273–277.

55. Rappaport, T.; Gutierrez, F.; Ben-Dor, E.; Murdock, J.N.; Qiao, Y.; Tamir, J.I. Broadband Millimeter-Wave Propagation Measurements and Models Using Adaptive-Beam Antennas for Outdoor Urban Cellular Communications. *IEEE Trans. Antennas Propag.* **2013**, *61*, 1850–1859. [CrossRef]
56. Rappaport, T.S.; Heath, R.W., Jr.; Daniels, R.C.; Murdock, J.N. *Millimeter Wave Wireless Communications*; Pearson: London, UK, 2014.
57. Lu, X.; Wang, P.; Niyato, D.; Kim, D.I.; Han, Z. Wireless charging technologies: Fundamentals, standards, and network applications. *IEEE Commun. Surv. Tutor.* **2015**, *18*, 1413–1452. [CrossRef]
58. Ng, D.W.K.; Lo, E.S.; Schober, R. Multiobjective Resource Allocation for Secure Communication in Cognitive Radio Networks With Wireless Information and Power Transfer. *IEEE Trans. Veh. Technol.* **2016**, *65*, 3166–3184. [CrossRef]
59. Chang, Z.; Gong, J.; Ristaniemi, T.; Niu, Z. Energy-Efficient Resource Allocation and User Scheduling for Collaborative Mobile Clouds With Hybrid Receivers. *IEEE Trans. Veh. Technol.* **2016**, *65*, 9834–9846. [CrossRef]
60. Sen, S.; Santhapuri, N.; Choudhury, R.R.; Nelakuditi, S. Successive interference cancellation: A back-of-the-envelope perspective. In Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks, New York, NY, USA, 20–21 October 2010; pp. 17:1–17:6.
61. Bertsekas, D.P. *Dynamic Programming and Optimal Control*; Athena Scientific: Belmont, MA, USA, 1995; Volume 1.
62. Li, X.; Fang, J.; Cheng, W.; Duan, H.; Chen, Z.; Li, H. Intelligent Power Control for Spectrum Sharing in Cognitive Radios: A Deep Reinforcement Learning Approach. *IEEE Access* **2018**, *6*, 25463–25473. [CrossRef]
63. Ioffe, S.; Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Proceedings of the 32nd International Conference on Machine Learning, Lille, France, 7–9 July 2015; Volume 37, pp. 448–456.
64. Sutton, R.S.; Barto, A.G. *Introduction to Reinforcement Learning*, 1st ed.; MIT Press: Cambridge, MA, USA, 1998.
65. Alkhateeb, A.; Alex, S.; Varkey, P.; Li, Y.; Qu, Q.; Tujkovic, D. Deep Learning Coordinated Beamforming for Highly-Mobile Millimeter Wave Systems. *IEEE Access* **2018**, *6*, 37328–37348. [CrossRef]
66. Fudenberg, D.; Tirole, J. *Game Theory*; MIT Press: Cambridge, MA, USA, 1991.