SpringerBriefs in Computer Science

Alice Faisal · Ibrahim Al-Nahhal · Octavia A. Dobre · Telex M. N. Ngatched

Reinforcement Learning for Reconfigurable Intelligent Surfaces

Assisted Wireless Communication Systems

Springer

SpringerBriefs in Computer Science

SpringerBriefs present concise summaries of cutting-edge research and practical applications across a wide spectrum of fields. Featuring compact volumes of 50 to 125 pages, the series covers a range of content from professional to academic.

Typical topics might include:

- A timely report of state-of-the art analytical techniques
- A bridge between new research results, as published in journal articles, and a contextual literature review
- A snapshot of a hot or emerging topic
- An in-depth case study or clinical example
- A presentation of core concepts that students must understand in order to make independent contributions

Briefs allow authors to present their ideas and readers to absorb them with minimal time investment. Briefs will be published as part of Springer's eBook collection, with millions of users worldwide. In addition, Briefs will be available for individual print and electronic purchase. Briefs are characterized by fast, global electronic dissemination, standard publishing contracts, easy-to-use manuscript preparation and formatting guidelines, and expedited production schedules. We aim for publication 8–12 weeks after acceptance. Both solicited and unsolicited manuscripts are considered for publication in this series.

**Indexing: This series is indexed in Scopus, Ei-Compendex, and zbMATH **

Alice Faisal • Ibrahim Al-Nahhal •
Octavia A. Dobre • Telex M. N. Ngatched

Reinforcement Learning for Reconfigurable Intelligent Surfaces

Assisted Wireless Communication Systems

Springer

Alice Faisal
Faculty of Engineering and Applied Science
Memorial University
St. John's, NL, Canada

Ibrahim Al-Nahhal
Faculty of Engineering and Applied Science
Memorial University
St. John's, NL, Canada

Octavia A. Dobre
Faculty of Engineering and Applied Science
Memorial University
St. John's, NL, Canada

Telex M. N. Ngatched
Faculty of Engineering
McMaster University
Hamilton, ON, Canada

ISSN 2191-5768 ISSN 2191-5776 (electronic)
SpringerBriefs in Computer Science
ISBN 978-3-031-52553-7 ISBN 978-3-031-52554-4 (eBook)
https://doi.org/10.1007/978-3-031-52554-4

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Preface

As the wireless communication networks are advancing toward their sixth generation, the key enabling technologies need to be thoroughly investigated. Recently, reconfigurable intelligent surfaces (RISs) have emerged as a promising solution to realize the demands of future wireless communication systems. They consist of low-cost passive reflecting elements that can be independently tuned to boost the received signal quality. RISs have the ability to control, amongst others, the phase of the electromagnetic waves that are reflected, refracted, and scattered. This feature enables RISs to effectively control the randomness of the propagation environment, leading to enhanced signal quality and strength, enhanced security, increased data rates, reduced error rates, and improved coverage. Furthermore, since the RIS elements are passive (i.e., they do not require a direct power source), the RISs can be deployed at low-cost, which makes them efficient in large-scale wireless systems. The RIS reflection coefficients can be optimized along with different parameters to maximize key performance metrics, such as the sum rate, secrecy rate, energy efficiency, signal coverage, etc. These features make RISs a critical component for future wireless communication systems.

Optimizing RIS-assisted wireless systems requires powerful algorithms to cope with the dynamic propagation environment and time-frequency-space varying channel conditions. Most of the current work on optimizing the RISs relies on alternating optimization techniques. Although such approaches can provide near-optimal solutions, they rely on well-established mathematical relaxations that change depending on the wireless communication system and objective function. Furthermore, they are not scalable. Future wireless systems will be characterized by massive number of connected devices, base stations, and sensors. Designing and controlling such large-scale wireless systems under dynamic environments will be infeasible given the considered relaxations for deriving explicit and solvable mathematical formulations of the wireless systems. Therefore, developing adaptive approaches through sensing and learning is needed to efficiently optimize the RIS reflection coefficients.

Deep reinforcement learning (DRL) is envisioned as one of the key enabling techniques to exploit the full potential of dynamic RIS-assisted wireless commu-

nication environments. DRL can adapt to the wireless system requirements and achieve its target performance through learning from experience. Furthermore, the DRL frameworks can learn to select the optimal configuration of the RIS reflection coefficients based on the current conditions, such as the location of the wireless device or the presence of obstacles, by maximizing a reward function. Moreover, it is well-suited for optimizing wireless systems, as DRL approaches can solve untractable non-linear mathematical formulations, without the need for either prior relaxations or prior knowledge of the communication environment.

In this book, we provide a comprehensive overview of RL approaches, with examples of applying DRL in the optimization of RIS-assisted wireless systems. Chapter 1 presents the holistic background of RL and details some of the widely used algorithms according to the problem class (i.e., continuous and discrete). Chapter 2 focuses on presenting the RIS-assisted wireless system model and potential scenarios. Chapters 3 and 4 explain the application of DRL to solve continuous and discrete problems in RIS-assisted wireless communication systems, respectively. Finally, Chap. 5 concludes the book by discussing the challenges of DRL and potential research directions in RIS-assisted communication systems.

St. John's, NL, Canada — Alice Faisal
St. John's, NL, Canada — Ibrahim Al-Nahhal
St. John's, NL, Canada — Octavia A. Dobre
Hamilton, ON, Canada — Telex M. N. Ngatched
October 2023

Contents

Acronyms

AO	Alternating optimization
AWGN	Additive white Gaussian noise
BS	Base station
CSI	Channel state information
DDPG	Deep deterministic policy gradient
DL	Downlink
DNN	Deep neural network
DQL	Deep Q-learning
DRL	Deep reinforcement learning
FD	Full-duplex
HD	Half-duplex
HER	Hindsight experience replay
IEN	Imitation environment network
LoS	Line-of-sight
MISO	Multiple-input single-output
mmWave	Millimeter wave
NN	Neural network
OFDM	Orthogonal frequency division multiplexing
RIS	Reconfigurable intelligent surface
RL	Reinforcement learning
SAC	Soft actor-critic
SI	Self-interference
SINR	Signal-to-interference-plus-noise ratio
SWIPT	Simultaneous wireless information and power transfer
TD3	Twin delayed DDPG
UAV	Unmanned aerial vehicle
UE	User equipment
UL	Uplink

Chapter 1
Reinforcement Learning Background

1.1 Overview

Reinforcement learning (RL) is a subfield of machine learning that aims to maximize a reward function. Unlike supervised learning, it does not need a labeled training dataset to learn the optimal configuration of the problem. Instead, it learns to make an optimal decision based on the current conditions by taking actions in the environment and receiving rewards or penalties accordingly. The key terms that describe the basic components of an RL problem include the *agent*, *environment*, *state*, *action*, and *reward*. The *agent* represents the decision-maker in the environment. The *environment* represents the physical configuration that the agent operates in. It provides the agent with information about the state of the system and reward associated with each action. The *state* represents the current configuration of the environment. In response to the state of the environment, the agent chooses an *action* based on the deployed policy, which represents the method of choosing actions. The agent then receives a feedback (i.e., positive or negative) based on the chosen action in a form of a scalar *reward*. In wireless communications, the RL environment represents the wireless propagation environment, and the agent is the microcontroller that is programmed to choose actions based on the learning experience. The RL actions represent the optimization variables of the wireless system, such as power, phase shifts, user scheduling, etc. The RL states can be designed in several ways depending on the problem, but they generally include key elements of the wireless environment that the RL agent needs to know to enhance the learning process, such as transmit power, receive power, channel states, etc. Finally, the RL reward is typically related to the optimization objective function of the wireless system to lead the agent to learn successfully. Figure 1.1 illustrates the general working principle of an RL agent.

The goal of an RL agent is to learn the optimal policy that maximizes its cumulative reward. The learning of the agent evolves based on its interaction with the environment. It chooses a suitable action, a_t, at time step t based on the state,

A. Faisal et al., *Reinforcement Learning for Reconfigurable Intelligent Surfaces*,
SpringerBriefs in Computer Science, https://doi.org/10.1007/978-3-031-52554-4_1

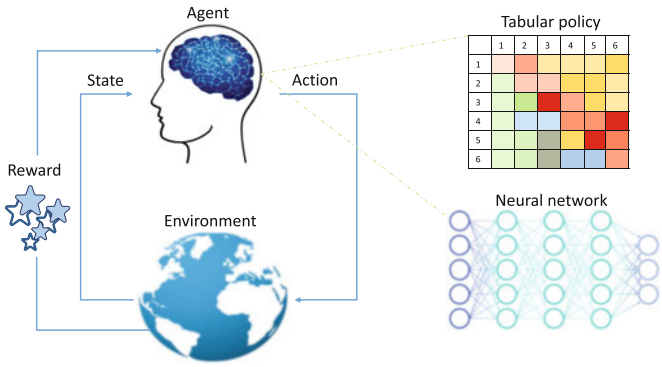


Fig. 1.1 DRL learning process

s_t, of the environment and transitions to a new state, s_{t+1}. The reward signal, r, received from the environment is used to update the agent's policy, leading to maximizing its cumulative reward. The policy differs based on the requirements of the problem and the agent's goal. There are two main types of policies in RL: *Deterministic* and *stochastic*. The former maps each state to a unique action. In other words, if the agent revisits the same state, it would always take the same action. The latter maps each state to a probability distribution over actions. This means that in each state, the agent will take an action based on the probabilities assigned to each action by the policy. One of the widely used deterministic policies is the tabular policy. The table contains the expected reward for each action-state pair, in which the agent follows the policy of choosing the action which corresponds to the maximum reward. Furthermore, for complicated systems, such as wireless systems, with a large number of state-action pairs, the tabular policy can be replaced by a neural network (NN) that takes observations as inputs and outputs the action or the probability distribution over actions, depending on whether the policy is deterministic or stochastic.

By combining RL with deep learning, deep reinforcement learning (DRL) can learn to optimize non-linear large-scale systems without the need for prior knowledge of the system dynamics. DRL algorithms have been used to solve a wide range of problems in different fields, including complex games, robots, and optimizing resource allocation in wireless communication networks. The DRL algorithms have also been used to learn optimal policies for decision-making in real-world problems, such as optimizing energy consumption in smart grids and traffic control in urban areas. The DRL algorithms can handle complicated wireless systems with high-dimensional state spaces and non-linear dynamics, making them a powerful tool for many real-world applications.

DRL is well-suited for solving wireless communication problems, specifically reconfigurable intelligent surface (RIS)-assisted systems, for many reasons. (a) *Adaptability*: DRL can learn to adapt to the dynamic changes of the wireless environment. In particular, the wireless propagation environment can be affected

by various factors such as the mobility of users, channel states, presence of new obstacles, and availability of resources. The DRL allows the RIS-assisted system to continuously learn to adapt to these changes and adjust its configuration accordingly to optimize the system's performance. (b) *Flexibility*: DRL can learn to optimize multiple objectives. To fully exploit the RIS-assisted systems in real-life applications, multiple objectives might have to be optimized jointly, such as maximizing the sum rate, minimizing the power consumption, and maximizing the number of supported devices. DRL can learn to optimize these multiple objectives simultaneously by designing the appropriate reward function. It can further be integrated with other techniques, such as supervised learning, unsupervised learning, and evolutionary algorithms to improve its performance. (c) *Complexity*: DRL can handle high-dimensional and complex state and action spaces. RIS-assisted systems can have a massive number of possible configurations under various conditions, making it difficult/complex to optimize using traditional methods. (d): *Practicality*: DRL can learn how to optimize the performance of RIS-assisted systems without prior knowledge of the wireless propagation environment, the number of users, or the distribution of channels. This makes DRL a powerful tool for designing and optimizing RISs in unknown or changing environments. Furthermore, for most of the practical wireless systems, the mathematical modeling of the environment is infeasible. Traditional optimization techniques may suffer from the excessive need for prior mathematical relaxation requirements, which might be impractical in dynamic wireless systems. In contrast, the DRL can handle the objectives of these wireless systems by learning from experience.

RL optimization spaces can be classified into two main categories: discrete and continuous. In discrete RL problems, the action space, which represents the optimization variables of a problem, is discrete. The state space is typically represented by a set of discrete finite states. A popular example of a discrete RL problem is a chess game, where the state space is the set of all possible positions of the pieces on the board, the action space is the set of all possible moves, and the rewards are positive for winning moves and negative for losing moves. In wireless communications, optimizing finite RIS phase shifts is considered a discrete problem. In contrast, in continuous RL problems, the RL agent chooses an action from an unbounded and continuous range of values rather than a finite and discrete set of values. An example of a continuous RL problem is a robot navigation problem, where the state space is the robot's position and orientation, the action space is the robot's control inputs, and the rewards are positive for reaching the goal and negative for colliding with obstacles. In comparison to discrete action spaces, continuous action spaces pose greater challenges in the design and implementation of the RL algorithms, as the continuous nature of the problem makes it harder to find an optimal RL policy. To tackle this challenge, function approximation techniques are needed, such as deploying NNs to represent the policy and gradient-based optimization methods to update it. This approach allows for a more flexible representation and enables the RL agent to learn more complex and sophisticated policies. We hereby discuss the details of the most powerful RL algorithms used to

solve discrete and continuous problems and present their applicability to the RIS-assisted wireless problems later in this chapter.

1.2 Discrete Spaces

1.2.1 Q-Learning

Q-learning is an efficient value-based model-free RL algorithm which does not require a model of the environment, making it easier to be deployed in a wide range of wireless communication problems whose dynamics may be unknown or difficult to model. Q-learning aims to learn the optimal policy by estimating the action-value function, known as the *Q-function*, which represents the expected reward for taking a discrete action in a specific state. This function is updated continuously as the RL agent interacts with the environment and observes the rewards and then is used to select the best action at each time step (i.e., representing the policy) [1]. In RL, the agent interacts with the environment in the form of episodes and steps. The cycle of state-action-reward represents a time step t. The agent continues to iterate through cycles until it reaches the desired state or the predetermined number of steps (i.e., the terminal state). This series of steps forms an episode k [2].

At each time step, the agent observes the current state, chooses an action based on its current policy, receives a reward from the environment, and updates the Q-values estimates. The Q-values can be thought of as a table that maps the states and actions to real-valued estimates of the expected future RL reward. The table is updated using the temporal difference (i.e., the difference between the expected reward and observed reward) control algorithm as follows

$$Q(s_t, a_t) = \underbrace{Q(s_t, a_t)}_{\substack{\text{current Q-value} \\ \text{estimate}}} + \alpha \Bigg[\underbrace{r_{t+1}}_{\substack{\text{observed} \\ \text{reward}}} + \gamma \underbrace{\max_a Q(s_{t+1}, a)}_{\substack{\text{maximum expected} \\ \text{future reward}}} - Q(s_t, a_t) \Bigg], \tag{1.1}$$

where α and γ denote the RL learning rate and discount factor, respectively. In Q-learning, the learning rate and discount factor are critical hyperparameters that govern the update rule of the Q-table. The learning rate dictates the extent to which the Q-values are updated in response to observed RL rewards. A high RL learning rate leads to rapid Q-value updates, which can result in unstable convergence, while a low learning rate results in a more stable but slow convergence rate. The optimum RL learning rate value strikes a balance between the speed of convergence and stability. The RL discount factor, on the other hand, controls the importance of the future rewards when estimating Q-values. The discount factor is a scalar value between 0 and 1, and reflects the degree to which future rewards are discounted. A discount factor of 1 implies that future rewards are considered to be as critical as immediate rewards, while a discount factor close to 0 implies that future rewards

are completely ignored in favor of immediate rewards. The optimum discount factor value balances the trade-off between long-term and short-term rewards. In the update rule, the discount factor governs the weight given to future rewards, enabling the agent to balance between short-term and long-term rewards. The optimum values of α and γ depend on the nature of the problem and environment, and may be determined through a trial and error approach.

Having discussed the update rule of the Q-table, now the question would be *how to choose the action in step t based on the Q-table?* The algorithm balances the exploration-exploitation trade-off by using an exploration policy, such as epsilon-greedy, to choose actions. In RL, the *exploitation process* refers to using the agent's knowledge to choose actions, while *exploration process* refers to trying new actions to gain more information about the environment. The trade-off is crucial as the agent must find a balance to maximize the reward while exploring new actions to improve its understanding of the environment. The epsilon-greedy (ϵ-greedy) policy defines the probability with which the agent selects a random action rather than exploiting its knowledge (i.e., selecting the action corresponding to the highest estimated Q-value). A high value of ϵ results in a higher rate of exploration, while a low value of ϵ results in a higher rate of exploitation. The value of ϵ is typically decreased over time, allowing the agent to gradually transition from exploring to exploiting as it gains more information about the environment, starting with $\epsilon = 1$, down to $\epsilon = 0.05$.

The agent's learning process continues (i.e., choosing an action through exploration or exploitation, measuring the reward, and updating the Q-table) until reaching convergence. Typically, the Q-learning algorithm is guaranteed to converge over time. However, the convergence depends on the tuning of α, γ, and ϵ. Furthermore, assuming that all state–action pairs are visited and updated, Q has been proved to converge with probability of 1 to the optimal solution, Q^*. The Q-learning algorithm is shown in Algorithm 1 in detail.

The main problem with Q-learning is that it is challenging to be scaled to large-scale problems with many states and actions. For example, if Q-learning was deployed to train an agent to play a chess game, the Q-table would handle around 10^{40} entries. These entries represent the state space, which takes into account the various pieces and their positions on the board, as well as other factors such as the turn, castling rights, and passant squares. Searching through the table and frequently updating it will be challenging, making it nearly impossible for the algorithm to converge to an optimal policy. Therefore, an estimation of the Q-value is required to tackle this problem.

1.2.2 Deep Q-Learning

One of the most powerful approaches to approximating the Q-function is using deep neural networks (DNNs). DNNs can handle high-dimensional input spaces and learn complex non-linear relationships efficiently, making it feasible to deploy the concept

Algorithm 1 Q-learning Algorithm.

Initialize: $\alpha \in (0, 1]$, $\gamma \in (0, 1]$, $\epsilon > 0$, the exploration threshold η, and exploration decay ρ, $Q(s, a)$ arbitrarily except $Q(\text{terminal}, .) = 0$;

repeat
  Initialize s, resetting the environment;
  repeat
    Initialize x, a random number between 0 and 1;
    if $x \leq \epsilon$ **then**
      Select a random action $a_t \in \mathcal{A}$;
    else

$$a_t = \max_a Q(s_t, a); \tag{1.2}$$

    end if
    Measure the reward r_t;
    Observe the new state, s_{t+1}, given a_t;
    Update Q using (1.1);
    $s_t \leftarrow s_{t+1}$
    if $\epsilon > \eta$ **then**
      $\epsilon \leftarrow \epsilon\rho$;
    end if
  until $t = T$;
until $k = K$;

Output: Q^*.

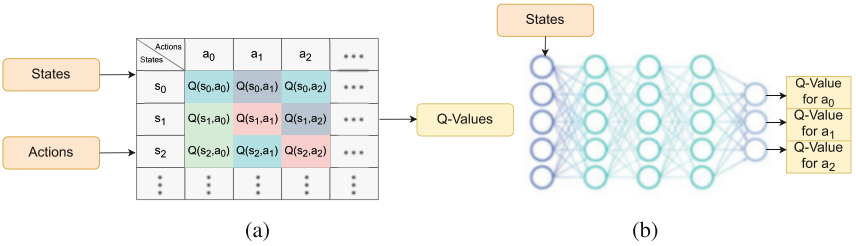


Fig. 1.2 Q-learning vs. DQL. (**a**) Q-learning. (**b**) DQL

of Q-learning in large-scale problems. Having said that, now the question would be how to train the network to approximate the Q-values? Generally, at each time step, the DNN with parameters θ takes the state as input and calculates the target, which is the estimated reward for taking action in that state, based on (1.1) [1]. The NN parameters are updated to minimize the difference between the predicted Q-value and the target one. This process is repeated for multiple iterations over which the NN continues to learn from the experience and improves its predictions. The training process can be improved using the replay buffer. Figure 1.2 highlights the difference between Q-learning and deep Q-learning (DQL).

Specifically, DQL uses the concept of a replay buffer, $\mathcal{D}$, to store experiences as the agent interacts with the environment. The size of the buffer can be denoted by its

cardinality $D = |\mathcal{D}|$. At time t, the agent's experience is defined as the tuple: $e_t = (s_t, a_t, r_t, s_{t+1})$. The use of a replay buffer is important in DRL because it allows the agent to learn from a more diverse set of experiences, including rare or infrequent events that would not be encountered if the network was updated after every single time step. By randomly sampling from the replay buffer, the network can be trained on a minibatch, $\mathcal{N}_\mathcal{B}$ of size $N_B = |\mathcal{N}_\mathcal{B}|$, containing diverse experiences, which makes the learning more stable and reduces the correlation between consecutive experiences [3]. Estimating the Q-values can be done by minimizing the following loss at each step t

$$L(\theta) = ((y_j - Q(s_j, a_j; \theta))^2, \tag{1.3}$$

where y_j is the target value defined as

$$y_j = r_j + \gamma \max_{a_{t+1}} Q(s_{t+1}, a_{t+1}; \theta). \tag{1.4}$$

Once the NN has been trained, it can be used to select actions by choosing the action corresponding to the highest predicted Q-value for a given state (i.e., greedy) or based on the exploration-exploitation trade-off (i.e., ϵ-greedy). The detailed steps are described in Algorithm 2.

Besides the basic implementation of DQL, several improvements have been proposed in game theory applications to enhance the performance and stability of the traditional algorithm [4]. Some notable improvements include the following: Double Q-learning, dueling network architecture, prioritized experience replay, distributional learning, noisy networks, and multi-step and rainbow learning. The double Q-learning addresses the issue of overestimation bias in traditional DQL. It utilizes two separate value functions to decouple the selection and evaluation of actions, mitigating the overestimation of action values. The dueling network architecture separates the estimation of state values and action expectations, allowing the agent to learn the value of being in a particular state independently of the chosen action. This architecture provides better insights into the value of different actions and enhances the learning efficiency. The prioritized experience replay assigns higher priorities to experiences with higher temporal difference errors, indicating that they are more informative for learning. By sampling experiences with higher priorities more frequently, prioritized experience replay improves sample efficiency and focuses the learning on important experiences. The distributional learning represents the action-value function as a distribution rather than a single value. This allows for a more comprehensive understanding of the uncertainty and variability in the value estimates, enabling better exploration and handling of risk-sensitive scenarios. Noisy networks introduce noise into the parameters of the network during training to encourage exploration. By adding parameter noise, the agent can explore different actions more effectively, leading to improved learning and more robust policies. The multi-step learning, also known as n-step learning, incorporates multiple future rewards into the update step. By considering

Algorithm 2 DQL Algorithm.

Initialize: θ with random weights, D, $\alpha \in (0, 1]$, $\gamma \in (0, 1]$, $\epsilon > 0$, the exploration threshold η, random number between 0 and 1, x, and exploration decay ρ;

repeat
  Initialize s, resetting the environment;
  repeat
    if $x \leq \epsilon$ **then**
      Select a random action $a_t \in \mathcal{A}$;
    else

$$a_t = \max_a Q(s_t, a; \theta); \tag{1.5}$$

    end if
    Measure the reward r_t;
    Observe the new state, s_{t+1}, given a_t;
    Store e_t in D;
    Sample N_B transitions (s_j, a_j, r_j, s_{j+1}) randomly from D when it is full;
    if $\epsilon > \eta$ **then**
      $\epsilon \leftarrow \epsilon\rho$;
    end if
    Compute the target value using (1.4);
    Perform a gradient descent step on (1.3);
    $s_t \leftarrow s_{t+1}$
  until $t = T$;
until $k = K$;

Output: Q^*.

future rewards over multiple time steps, the agent can learn faster and make more informed decisions. Rainbow is an integration of several improvements into a single framework. It combines double Q-learning, prioritized experience replay, dueling network architecture, multi-step learning, and noisy networks, resulting in a powerful and highly effective DQL algorithm. These improvements aim to address various limitations and challenges of traditional DQL, such as overestimation bias, sample efficiency, exploration-exploitation trade-off, and robustness to uncertainty. By incorporating these enhancements, the performance, stability, and learning efficiency of DQL algorithms can be significantly improved.

1.3 Continuous Spaces

Many wireless communication problems have high-dimensional real-valued action spaces, especially RIS-assisted systems whose phase shifts affect the performance greatly. While DQL can be deployed to handle problems with high-dimensional state and action spaces, it cannot be applied to handle continuous data. This is due to the way the DQL selects actions: finding the action that has the highest Q-

value. In the case of continuous action spaces, the optimization process becomes much more complex and requires an iterative optimization process at every step. An intuitive approach to adapting DQL to continuous problems is to discretize the action space. However, this solution imposes several limitations, such as the curse of dimensionality, where the number of actions grows rapidly with the number of degrees of freedom. Dealing with large action spaces makes it difficult to explore the environment efficiently. Moreover, the simple discretization of action spaces needlessly loses important information about the action domain characteristics, which may be crucial for optimizing many problems. To this end, a model-free actor-critic algorithm, named deep deterministic policy gradient (DDPG), was proposed to approximate policies for continuous action spaces [5]. The algorithm is closely connected to DQL in terms of learning goals. In DQL, if the optimal Q-function $Q(s, a)$ is known, then in any given state, the optimal action can be found by solving (1.2) [6]. In contrast, DDPG aims to learn an approximate to the optimal action space, which makes it specifically adapted for environments with continuous action spaces [7]. Furthermore, since the action space is continuous, the function $Q(s, a)$ is assumed to be differentiable with respect to the action. This allows setting an efficient gradient-based learning method for a policy, $\mu(s)$, which exploits that fact. Instead of computing $\max_a Q(s, a)$, it is approximated with $\max_a Q(s, a) \approx Q(s, \mu(s))$.

1.3.1 DDPG

DDPG is based on the *actor-critic* approach, which is comprised of two main DNN models: actor and critic NNs. The former, $\mu(s_t|\boldsymbol{\theta}_\mu)$, defines the policy network that takes the state as an input and outputs the approximated action. The latter, $Q(s_t, a_t|\boldsymbol{\theta}_q)$, defines the evaluation network that takes the state and action as an input and outputs the approximated Q-value. Similar to the DQL algorithm, DDPG incorporates the concept of replay buffer, D, to minimize the correlation between the training samples by sampling random minibatch transitions, N_B. Furthermore, the DDPG algorithm introduces the concept of target networks, which are copies of the actor and critic networks, denoted by $\mu'(s_t|\boldsymbol{\theta}_\mu)$ and $Q'(s_t, a_t|\boldsymbol{\theta}_q)$. They are used to calculate the target values as [8]

$$y_j = r_j + \gamma Q'(s_{j+1}, \mu'(s_{j+1}|\boldsymbol{\theta}_{\mu'})|\boldsymbol{\theta}_{q'}). \tag{1.6}$$

Here, the target values depend on the same parameters that are trained. Therefore, introducing the target network stabilizes the learning process by causing a delay in the network. In particular, the weights of the target networks are updated through a soft update coefficient by slowly tracking the learned actor and critic networks, rather than directly copying their weights [9]. To this end, the critic network is updated by minimizing the mean-squared loss between the updated Q-value and the original Q-value, defined as

$$L = \frac{1}{N_B} \sum_j \left(y_j - Q(s_j, a_j | \boldsymbol{\theta}_q) \right)^2 . \tag{1.7}$$

On the contrary, the actor network update function depends on the objective function, which chooses actions that maximize the expected return as follows

$$J(\theta) = \mathbb{E} \left[Q(s, a) | s = s_t, a_t = \mu(s_t) \right] . \tag{1.8}$$

The actor network is updated by taking the derivative of the objective function with respect to the policy parameter, expressed as

$$\nabla_{\boldsymbol{\theta}_\mu} = \frac{1}{N_B} \sum_j \nabla_a Q(s, a | \boldsymbol{\theta}_q)|_{s=s_j, a=\mu(s_j)} \nabla_{\boldsymbol{\theta}_\mu} \mu(s | \boldsymbol{\theta}_\mu)|_{s_j} . \tag{1.9}$$

The target networks are updated using Polyak averaging as follows

$$\boldsymbol{\theta}_{q'} \longleftarrow \tau \boldsymbol{\theta}_q + (1 - \tau) \boldsymbol{\theta}_{q'}, \tag{1.10}$$

$$\boldsymbol{\theta}_{\mu'} \longleftarrow \tau \boldsymbol{\theta}_\mu + (1 - \tau) \boldsymbol{\theta}_{\mu'}, \tag{1.11}$$

where $\tau \in (0, 1]$ is the soft update coefficient. In DQL, the exploration process was done by selecting a random action based on the ϵ-greedy policy, while in DDPG, the actor network approximates the actions directly. Therefore, to improve the learning process of the RL agent, the exploration can be enforced by adding a random process (i.e., noise) to the action (i.e., $a_t = \mu(s_t | \boldsymbol{\theta}_\mu) + \xi$). The noise type is selected based on the environment of the problem. For instance, in wireless communications, ξ can be modeled as a Gaussian process with zero-mean and a variance of 0.1. The complete algorithm steps of DDPG is summarized in Algorithm 3.

Several improvements have been proposed to enhance the performance and stability of the traditional DDPG algorithm. Some notable improvements include: *Actor-critic architectures:* DDPG can benefit from improved actor-critic architectures, such as the twin delayed DDPG (TD3) and soft actor-critic (SAC) algorithms. In particular, TD3 extends the DDPG by incorporating twin critics and delayed updates. The twin critics in the TD3 help mitigate the overestimation bias commonly found in value-based methods. By using two separate critics, the TD3 can estimate the value function more accurately and reduce the variance of value estimates. The delayed updates involve updating the target networks less frequently than the policy and value networks, which stabilizes the learning process. Delaying the updates helps to decorrelate the value estimates and mitigate issues associated with overfitting the current value estimates. On the other hand, the SAC leverages entropy regularization to encourage exploration and achieve more robust policies. In particular, the SAC is an off-policy actor-critic approach that combines the advantages of maximum entropy RL and stochastic policies. It introduces entropy regularization to encourage exploration and enable the learning of more diverse and

Algorithm 3 DDPG Algorithm.

Initialize: $\boldsymbol{\theta}_\mu$ and $\boldsymbol{\theta}_q$ with random weights, D, γ, τ, and α, **Set:** $\boldsymbol{\theta}_{\mu'} \leftarrow \boldsymbol{\theta}_\mu$ and $\boldsymbol{\theta}_{q'} \leftarrow \boldsymbol{\theta}_q$;
repeat
  Initialize s, resetting the environment;
  Initialize $\xi \sim \mathcal{N}(0, 0.1)$;
  repeat
    Obtain $a_t = \mu(s_t|\boldsymbol{\theta}_\mu) + \xi$ from the actor network;
    Observe the new state, s_{t+1}, given a_t;
    Store (s_t, a_t, r_t, s_{t+1}) in D;
    When D is full, sample a minibatch N_B transitions randomly (s_j, a_j, r_j, s_{j+1}) from D;
    Calculate the target value using (1.6);
    Update the critic by minimizing the loss using (1.7);
    Update the actor using the policy gradient as in (1.9);
    Update the target NNs through soft update using (1.10) and (1.11);
  until $t = T$;
until $k = K$;
Output: a^*.

robust policies. By maximizing the policy entropy, the SAC promotes exploration, avoids premature convergence to suboptimal solutions, and can handle tasks with continuous and high-dimensional action spaces effectively.

Other improvements of the traditional DDPG algorithm include parametric noise, distributional DDPG, hindsight experience replay (HER), batch normalization, and parameter sharing. In particular, adding parameter noise to the actor network during training can improve exploration in the DDPG. By injecting noise into the policy parameters, the agent is encouraged to explore different actions, leading to better policy learning and improved performance. Similar to distributional DQL-learning, distributional DDPG represents the action-value function as a distribution. This approach provides a more comprehensive understanding of the uncertainty in the action-value estimates, enabling better exploration and handling of sensitive scenarios. Moreover, the HER is a technique that can accelerate learning in the DDPG for tasks with sparse rewards. The key idea behind HER is to leverage hindsight knowledge to relabel unsuccessful experiences and treat them as if they had achieved the desired goal. This enables the agent to learn from a broader range of experiences and benefit from a larger set of positive rewards, even if the original attempts did not succeed. The agent thus learns from both successes and failures, improving sample efficiency and learning speed. Applying batch normalization to the NNs in the DDPG can improve stability and learning performance. By normalizing the inputs to each layer, batch normalization helps alleviate issues related to covariate shifts and enables faster and more stable training. Parameter sharing can be employed in certain scenarios to improve the efficiency of DDPG. By sharing parameters between multiple agents in a multi-agent setting, the learning process can be accelerated, and knowledge transfer can occur between agents. These improvements aim to address various challenges faced by the traditional DDPG algorithm, such as exploration-exploitation trade-off, sample efficiency, stability, and learning speed.

By incorporating these enhancements, the performance, robustness, and learning efficiency of DDPG algorithms can be significantly improved.

References

1. Sutton RS, Barto AG (2018) Reinforcement learning: an introduction. MIT Press, Cambridge
2. Watkins CJ, Dayan P (1992) Q-learning. Mach Learn 8:279–292
3. Mariano CE, Morales EF (2001) DQL: a new updating strategy for reinforcement learning based on Q-learning. In: De Raedt L, Flach P (eds) Machine learning: ECML 2001. Springer Berlin Heidelberg, Berlin/Heidelberg, pp 324–335
4. Géron A (2022) Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow. O'Reilly Media, Inc., Sebastopol
5. Lillicrap TP, Hunt JJ, Pritzel A, Heess N, Erez T, Tassa Y, Silver D, Wierstra D (2016) Continuous control with deep reinforcement learning. In: Proceedings of the International Conference on Learning Representations (ICLR), May 2016, pp 1–14
6. van Hasselt H, Guez A, Silver D (2016) Deep reinforcement learning with double q-learning. Proc AAAI Conf Artif Intell 30(1) [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/10295
7. Hou Y, Liu L, Wei Q, Xu X, Chen C (2017) A novel DDPG method with prioritized experience replay. In: 2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC), November 2017, pp 316–321
8. Mnih V, Kavukcuoglu K, Silver D, Rusu AA, Veness J, Bellemare MG, Graves A, Riedmiller MA, Fidjeland A, Ostrovski G, Petersen S, Beattie C, Sadik A, Antonoglou I, King H, Kumaran D, Wierstra D, Legg S, Hassabis D (2015) Human-level control through deep reinforcement learning. Nature 518(7540):529–533
9. Wu X, Liu S, Zhang T, Yang L, Li Y, Wang T (2018) Motion control for biped robot via DDPG-based deep reinforcement learning. In: 2018 WRC Symposium on Advanced Robotics and Automation (WRC SARA), December 2018, pp 40–45

Chapter 2
RIS-Assisted Wireless Systems

2.1 Overview

Traditional wireless radio transmission utilizes conventional reflecting surfaces, which only induce fixed phase shifts. While these surfaces can reflect signals, they lack the ability to actively control or adapt their reflection properties according to dynamic wireless environments. Thanks to the recent advancements in metamaterial science, RISs are proposed. It is composed of passive reflecting elements that can be electronically controlled to manipulate the properties of the incident waves, providing new degrees of freedom. RIS offers significant advantages over traditional reflective surfaces, which include: (a) Enhanced signal coverage and quality: By optimizing the RIS phase shifts, scattered waves and signal paths will be added constructively at the receiver side. This enables improving the signal-to-noise ratio and mitigating channel impairments such as fading or interference. (b) Flexibility and cost-effectiveness: RIS provides a promising solution for improving wireless communication performance. Compared to traditional active systems, RIS does not require complex and power-hungry electronics. The passive nature of RIS, combined with its ability to be deployed in several scenarios, allows for more economical installations. RIS can be integrated into existing infrastructure or deployed as standalone elements, offering scalability and performance improvement. (c) Security and privacy: By tuning the RIS phase shifts selectively for reflecting signals, RIS can create spatially restrict communication zones toward authorized users, thereby reducing the risk of eavesdropping or unauthorized access. (d) Other features include providing full-band response, mitigating interference, enabling dynamic reconfiguration, and enhancing the capacity [1, 2].

The passive RIS elements consist of metamaterial, which can be made of varactor diodes or other micro-electro-mechanical systems that can modify their induced phase shifts to achieve expected communication targets. Metasurfaces are characterized by their dynamic and adaptable behavior, accomplished through the use of tunable elements that can modify their electromagnetic response when

A. Faisal et al., *Reinforcement Learning for Reconfigurable Intelligent Surfaces*,
SpringerBriefs in Computer Science, https://doi.org/10.1007/978-3-031-52554-4_2

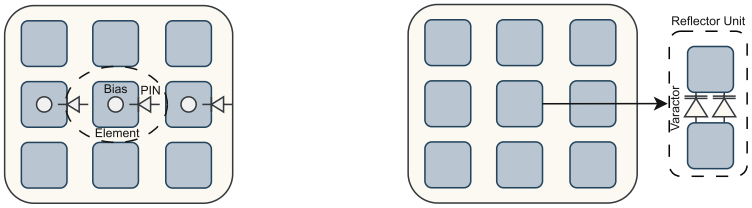


Fig. 2.1 Controlling reflections methods. (**a**) PIN diode. (**b**) Varactor-tuned resonator

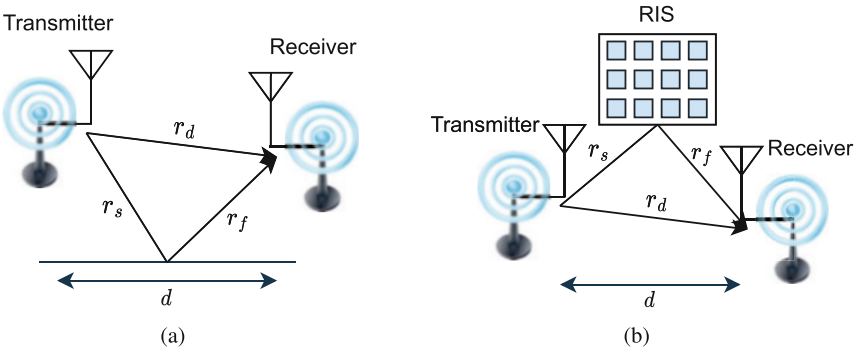


Fig. 2.2 Two-ray propagation model. (**a**) Conventional model. (**b**) RIS-assisted model

subjected to an external bias. These elements include complementary metal-oxide semiconductor or micro-electro-mechanical system switches that serve to control the meta-atoms acting as input and output antennas. When an incoming electromagnetic wave enters through an input antenna, it is routed according to the status of the switch and exits through an output antenna, enabling the RIS elements to achieve a customized reflection. These properties are made possible due to the ability of the switching elements to modify the behavior of the meta-atoms, which play a critical role in the function of metasurfaces [3].

One can control the metasurface reflective properties by applying an external bias to the PIN diodes, working as switching elements, as shown in Fig. 2.1a. When the PIN diode is switched off, the incoming signal is mostly absorbed. On the other hand, when the PIN diode is turned on, most of the incoming signal is reflected. Moreover, varactor-tuned resonators can also be used to control the signal reflection [4, 5]. In particular, by applying a bias voltage to the varactor diode, a tunable phase shift is achieved, as shown in Fig. 2.1b.

To best illustrate the RIS benefits, Fig. 2.2a shows a conventional two-ray propagation model of a wireless communication system. The received signal consists of a line-of-sight (LoS) signal and reflected signal from the ground. According to the Snell's law of reflection, the received power at distance d, P_d, is represented as

$$P_d = P_s \left(\frac{\lambda}{4\pi}\right)^2 \left|\frac{1}{r_d} + \frac{\Gamma \times e^{-j\Delta\phi}}{r_s + r_f}\right|^2, \tag{2.1}$$

where P_s is the signal transmitted power, λ is the signal wavelength, r_d is the distance between the transmit and receive antennas, r_s is the distance between the transmitter and the point of reflection, r_f is the distance between the point of reflection and the receive antenna, Γ is the ground reflection coefficient, and $\Delta\phi$ denotes the phase difference between the LoS and reflected paths of the received signal. It is shown in (2.1) that the reflection from the ground surface degrades the received signal power. However, if we consider an RIS with N elements to assist the communication between the transmitter and receiver, as shown in Fig. 2.1b, the received power relative to the i-th reconfigurable metasurface element, would be as follows

$$P_d = P_s \left(\frac{\lambda}{4\pi}\right)^2 \left|\frac{1}{r_d} + \sum_{i=1}^{N} \frac{\Gamma_i \times e^{-j\Delta\phi_i}}{r_{s,i} + r_{f,i}}\right|^2. \tag{2.2}$$

Under the assumption that the distance between the transmitter and receiver is large, and that there is no ground reflection, d would be approximately equivalent to $r_d \approx r_s + r_f$. Since each Γ_i is optimized to align the received signal phase with the LoS path, the received signal power in (2.2) can be simplified as

$$P_d \approx (N+1)^2 P_s \left(\frac{\lambda}{4\pi d}\right)^2, \tag{2.3}$$

which proves that the received signal power is directly proportional to the square of the number of the controlled RIS phases, N^2, and inversely proportional to the square of the distance between the transmitter and the receiver. This demonstrates the promising capabilities of RISs in wireless systems, as the signal power gain is directly related to the number of reflecting elements.

2.2 Scenarios

Due to the unique features of RISs, they can be deployed to various wireless communication scenarios to boost the system performance. By intelligently adjusting the phase shift, amplitude, or polarization of the reflected waves, RIS can respond to variations in channel conditions, user locations, or system requirements. This adaptability allows the RIS to optimize signal propagation and maintain high system performance even in complex scenarios [6]. The RIS phase shifts can be optimized along with other system parameters to target different problems, such as maximizing the sum rate, energy efficiency, and secrecy rate. In what follows, we present some prospective use cases of the RIS technology in future wireless networks.

2.2.1 RIS-Assisted Cognitive Radio Networks

One of the promising applications is deploying the RIS in cognitive radio systems. Since future generations of mobile communications are expected to support a massive number of connected devices, the radio frequency spectrum would mostly suffer from data congestion and spectrum scarcity. To this end, cognitive radio systems aim to increase the spectrum utilization by enabling unlicensed users (secondary network) to access the spectrum unoccupied by licensed users (primary network) while protecting the primary networks from interference problems. The secondary system has to be carefully designed to limit the performance degradation caused by the interference.

One of the key advantages of RIS-assisted cognitive radio networks is the ability to improve spectral efficiency through intelligent spectrum management. The RIS acts as an intelligent reflector that can selectively enhance or attenuate signals, allowing for improved signal quality, reduced interference, and increased network capacity. By adapting the RIS configuration based on real-time channel measurements and cognitive radio decision-making algorithms, RIS-assisted networks can efficiently reduce interference, optimize spectrum utilization, and enhance the overall system performance.

Several studies have proposed optimization frameworks that consider the coexistence of primary and secondary users, aiming to maximize the achievable rate of secondary users while satisfying interference constraints imposed by primary users [7–10]. These works have provided reliable insights into the advantages of the RIS deployment in improving the overall system performance by jointly optimizing the beamformers and power allocation. Furthermore, enhancing the security and privacy aspects of cognitive radio networks through RIS integration has also been a significant research direction. By designing a joint transmit beamforming and cooperative jamming strategy in RIS-assisted systems, the received signal quality can be significantly improved while also exploiting the jamming capabilities of the active eavesdropper, Eve.

Several studies in RIS-assisted cognitive radio networks involve spectrum sensing using machine learning techniques. Leveraging the capabilities of RISs, researchers have proposed novel approaches for signal detection and classification. These methods utilize machine learning algorithms to enhance the accuracy of spectrum sensing and achieve more reliable performance. Additionally, the utilization of the RL algorithms in resource allocation for RIS-assisted cognitive radio networks has been investigated. By formulating resource allocation as a Markov decision process, researchers have applied the RL techniques to learn optimal policies for dynamic spectrum allocation and RIS configuration. This approach enables the network to adapt to changing conditions and optimize resource utilization, improving overall system efficiency.

2.2.2 RIS-Assisted Unmanned Aerial Vehicle

The RIS has been leveraged for assisting unmanned aerial vehicle (UAV) communications, where it can be deployed on the ground or attached to UAVs to assist terrestrial communications by exploiting the RIS reflection from the sky. RISs strategically manipulate the wireless propagation environment, allowing UAVs to overcome obstacles, improve signal quality, and extend their communication range. This enables reliable and efficient communication between UAVs and ground stations, other UAVs, or wireless networks.

The UAVs are used for various services in practical scenarios, such as real-time data collection, traffic monitoring, military operations, and medical assistance. However, the UAVs suffer from fuel efficacy, environmental disturbances, and limited network capability, which makes the deployment of such powerful technology challenging. To this end, the RIS can be integrated with UAVs to enhance the system performance and combat the limitations of UAVs.

Several works have investigated RIS-assisted UAV systems, and it was proven that RIS can be efficiently integrated into different scenarios and boost the system performance [11–14]. One of the key benefits of RIS-assisted UAVs is the extension of flight range. By optimizing the RIS phase shifts, RISs enhance the received signal strength of the UAVs, enabling them to operate over longer distances. This opens up new possibilities for applications such as surveillance, monitoring, and exploration. The RISs also contribute to the energy efficiency of the UAVs by reducing the power required for communication. This results in energy savings and increased converge, enhancing the overall efficiency and operational capabilities of the UAVs.

RISs can further assist UAVs in achieving improved accuracy in navigation and localization applications. By leveraging the RISs ability to manipulate signals and reduce interference, UAVs can achieve better positioning accuracy. This enables more precise navigation, obstacle avoidance, and localization capabilities for UAVs, enhancing their overall performance in various environments.

Furthermore, RISs provide an adaptive environment for UAVs. With their reconfigurable nature, RISs can adjust the phase shifts, using appropriate algorithms, in real-time to respond to changes in the wireless channel or environmental conditions. This adaptability allows UAVs to maintain reliable and efficient communication links even in challenging scenarios, such as urban environments or areas with severe interference.

2.2.3 RIS-Assisted Simultaneous Wireless Information and Power Transfer

Simultaneous wireless information and power transfer (SWIPT) systems are wireless systems that enable the simultaneous transfer of both information and power to the intended receivers. In SWIPT, wireless signals are used not only for transmitting

data but also for harvesting energy to power the devices. SWIPT systems offer several advantages in wireless communications. They enable devices to operate without relying solely on external power sources, leading to enhanced energy sustainability. Moreover, SWIPT can be particularly beneficial in low-power or energy-constrained scenarios, such as internet of things systems, where energy harvesting can supplement or replace traditional power sources.

RIS-assisted SWIPT has emerged as a promising technology for various wireless systems environments. In RIS-assisted SWIPT, the RIS is deployed strategically to reflect and manipulate the wireless signals for both information transfer and power harvesting purposes. The RIS phase shifts can be controlled to jointly optimize the power transfer and information decoding at the receiver. Given that the RIS deployment is energy and cost-efficient, it plays a crucial role in optimizing SWIPT systems. The RIS can assist in mitigating the energy dissipation caused by propagation losses. This capability opens up possibilities for sustainable and self-powered wireless communication systems.

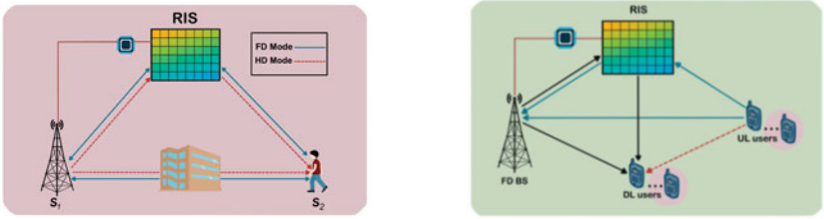
Several research efforts have been devoted to exploring RIS-assisted SWIPT in various applications [15–18]. Optimization techniques have been developed to optimize the RIS phase shifts and power allocation to maximize the signal-to-interference-plus-noise ratio (SINR) performance, considering factors such as transmit power constraints, channel conditions, and quality-of-service requirements. Furthermore, practical aspects, such as hardware impairments, have been considered to investigate the impact on SWIPT performance and devise techniques to mitigate their effects. It was shown that the RIS deployment results in a significant performance improvement in several scenarios as compared to traditional systems without RIS.

2.3 System Models

2.3.1 Half-Duplex

Current wireless communication systems use the half-duplex (HD) operation [19, 20]. In a HD system, transmission and reception occur in separate slots through time division duplex or frequency division duplex. This means that a device can either transmit or receive data at a given time but not both simultaneously over the same frequency or channel. The general form of the received signal in a traditional wireless system can be expressed as a linear combination of the transmitted signal and any added noise or interference . It can be affected by channel fading and other factors. Mathematically, the received signal can be represented as

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{n}, \tag{2.4}$$



(a) (b)

Fig. 2.3 RIS-assisted wireless communication systems. (**a**) HD-FD RIS System. (**b**) Multi-user UL DL RIS system

where $\mathbf{y}$ is a column vector of size $N_r \times 1$ representing the received signal at the receiver. $\mathbf{H}$ is a matrix of size $N_r \times N_t$ representing the channel gain matrix between the transmitter and receiver, where N_r is the number of receive antennas and N_t is the number of transmit antennas. $\mathbf{x}$ is a column vector of size $N_t \times 1$ representing the transmitted signal and $\mathbf{n}$ is a column vector of size $N_r \times 1$ representing the additive white Gaussian noise (AWGN) at the receiver. Consider deploying an RIS to assist the communication between the transmitter and receiver, as shown in Fig. 2.3a. Here, S_1 and S_2 represent the base station (BS) and user equipment (UE), respectively. The BS is equipped with M transmit antennas, while the UE is equipped with one receive antenna, representing a multiple-input single-output (MISO) system. Given $\bar{k} = 3 - k \; \forall \, k = 1, 2$, let $\mathbf{H}_{S_{\bar{k}}R} \in \mathbb{C}^{N \times M}$, $\mathbf{h}_{RS_k}^H \in \mathbb{C}^{1 \times N}$, and $\mathbf{h}_{S_{\bar{k}}S_k}^H \in \mathbb{C}^{1 \times M}$ denote the channel coefficients of the $S_{\bar{k}}$-RIS, RIS-S_k, and $S_{\bar{k}}$-S_k links, respectively.

In this case, the downlink (DL) received signal consists of both the direct and reflected links of the source and RIS as follows [21]

$$y_k = \Big(\underbrace{\mathbf{h}_{RS_k}^H \boldsymbol{\Theta} \mathbf{H}_{S_{\bar{k}}R}}_{\text{Reflected signal}} + \underbrace{\mathbf{h}_{S_{\bar{k}}S_k}^H}_{\text{Direct signal}} \Big) \mathbf{w}_{\bar{k}} x_{\bar{k}} + n, \; k = 2, \tag{2.5}$$

where $n \sim \mathcal{CN}(0, \sigma^2)$ represents the complex AWGN with zero-mean and variance σ^2. The diagonal matrix $\boldsymbol{\Theta} = \text{diag}\big(e^{j\varphi_1}, \cdots, e^{j\varphi_n}, \cdots, e^{j\varphi_N}\big) \in \mathbb{C}^{N \times N}$ denotes the phase shifts of the RIS elements, where $\varphi_n \in [-\pi, \pi)$ is the phase shift introduced by the n-th reflecting element. The source node employs an active beamforming $\mathbf{w}_k \in \mathbb{C}^{M \times 1}$ to transmit the information signal, x_k.

The following chapters will focus on explaining how DRL can be applied to optimize RIS-assisted systems. Several works leverage DRL for the RIS phase shifts optimization, while the beamformers vectors are optimized through closed-form solutions. In the HD case, the optimal beamforming vector can be obtained as follows

$$\mathbf{w}_{\bar{k}}^{\dagger} = \sqrt{P_{\max}} \frac{\left(\mathbf{h}_{RS_k}^H \boldsymbol{\Theta} \mathbf{H}_{S_{\bar{k}}R} + \mathbf{h}_{S_{\bar{k}}S_k}^H\right)^H}{\left\|\left(\mathbf{h}_{RS_k}^H \boldsymbol{\Theta} \mathbf{H}_{S_{\bar{k}}R} + \mathbf{h}_{S_{\bar{k}}S_k}^H\right)\right\|}, \; k = 2, \tag{2.6}$$

where $P_{\max}$ is the maximum transmitted power of $S_{\bar{k}}$.

2.3.2 Full-Duplex

The above communication system operates in a HD mode, where the UE only receives information from the BS, representing a one-way communication mode. In contrast, full-duplex (FD) communications enable the UE to send and receive information simultaneously on the same frequency, leading to higher throughput and more effective communications. However, The adoption of FD technology in wireless systems faces various technical challenges, such as self-interference (SI), and co-channel interference management. Ongoing research and development efforts continue to explore and address these challenges in order to realize the potential benefits of FD communication in future wireless networks. One of the promising solutions is to deploy an RIS to boost the system performance [22]. In particular, incorporating RISs into FD communications has a huge potential in combating the FD interference problems, facilitating ultra spectrum-efficient communication systems. Consider an RIS-assisted two-way communication system, as shown in Fig. 2.3a. The received signal can be expressed as [21]

$$y_i = \Big(\underbrace{\mathbf{h}_{RS_k}^H \boldsymbol{\Theta} \mathbf{H}_{S_{\bar{k}}R}}_{\text{Reflected signal}} + \underbrace{\mathbf{h}_{S_{\bar{k}}S_k}^H}_{\text{Direct signal}} \Big) \mathbf{w}_{\bar{k}} x_{\bar{k}} + \underbrace{\mathbf{h}_{S_kS_k}^H \mathbf{w}_k x_k}_{\text{Residual SI}} + n, \; k = 1, 2, \tag{2.7}$$

where $\mathbf{h}_{S_kS_k}^H \in \mathbb{C}^{1\times M}$ denotes the SI channels induced by the BS transmit and receive antennas. Here, the optimal beamforming vectors can be found using an approximate solution as follows

$$\mathbf{w}_{\bar{k}}^{\dagger} = (\delta \mathbf{h}_{S_{\bar{k}}S_{\bar{k}}} \mathbf{h}_{S_{\bar{k}}S_{\bar{k}}}^H + v^{\dagger}\mathbf{I})^{-1} \boldsymbol{\mathcal{B}}, \; k = 1, 2, \tag{2.8}$$

where $\mathbf{I}$ is the identity matrix and $v^{\dagger}$ is the Lagrangian variable associated with the power constraint. It can be obtained by performing a bisection search over the interval $\left[0, \sqrt{\boldsymbol{\mathcal{B}}^T \boldsymbol{\mathcal{B}}}/\sqrt{P_{\max}}\right]$, where $\boldsymbol{\mathcal{B}}$ and δ are given as [21]

$$\boldsymbol{\mathcal{B}} \triangleq \frac{1}{\tilde{b}_k} \left(1 + \frac{b_k}{|\mathbf{h}_{S_{\bar{k}}S_{\bar{k}}}^H \tilde{\mathbf{w}}_{\bar{k}}|^2 + \sigma^2}\right) \mathbf{h}_{\bar{k}} \mathbf{h}_{\bar{k}}^H \tilde{\mathbf{w}}_{\bar{k}}, \tag{2.9}$$

and

$$\delta \triangleq \frac{b_k \left(|\mathbf{h}_{\bar{k}}^H \tilde{\mathbf{w}}_{\bar{k}}|^2 + \tilde{b}_k \right)}{\tilde{b}_k \left(|\mathbf{h}_{S_{\bar{k}} S_{\bar{k}}}^H \tilde{\mathbf{w}}_{\bar{k}}|^2 + \sigma^2 \right)^2}. \tag{2.10}$$

Here, $b_k \triangleq |\mathbf{h}_k^H \mathbf{w}_k|^2$, $\tilde{b}_k \triangleq |\mathbf{h}_{S_k S_k}^H \mathbf{w}_k|^2 + \sigma^2$, $\mathbf{h}_{\bar{k}} \triangleq \mathbf{H}_{S_{\bar{k}} R}^H \boldsymbol{\Theta}^H \mathbf{h}_{R S_k} + \mathbf{h}_{S_{\bar{k}} S_k}$, and $\tilde{\mathbf{w}}_{\bar{k}}$ is a given feasible point. The work in [23] further introduced exact beamforming derivations to find the optimal solution, which can be expressed as

$$\mathbf{w}_k^* = (v^* + f_{\bar{k}} \boldsymbol{\alpha}_{\bar{k}} \boldsymbol{\alpha}_{\bar{k}}^H)^{-1} \boldsymbol{\beta}_{\bar{k}}, \tag{2.11}$$

where $f_{\bar{k}}^*$ and $b_{\bar{k}}$ are obtained as

$$f_{\bar{k}}^* = \frac{b_{\bar{k}}}{b_{\bar{k}}^2 + |\mathbf{h}_{S_{\bar{k}} S_{\bar{k}}}^H \mathbf{w}_{\bar{k}}|^2}, \tag{2.12}$$

$$b_{\bar{k}} = \left| \left(\sum_{r \in \Lambda} \mathbf{h}_{R_r S_{\bar{k}}}^H \boldsymbol{\Theta}_r \mathbf{H}_{S_k R_r} + \mathbf{h}_{S_k S_{\bar{k}}}^H \right) \mathbf{w}_k \right|^2, \tag{2.13}$$

and

$$\boldsymbol{\alpha}_{\bar{k}} = \sum_{r \in \Lambda} \mathbf{h}_{R_r S_{\bar{k}}}^H \boldsymbol{\Theta}_r \mathbf{H}_{S_k R_r} + \mathbf{h}_{S_k S_{\bar{k}}}^H, \tag{2.14}$$

$$b_{\bar{k}} = |\boldsymbol{\alpha}_{\bar{k}} \mathbf{w}_k|^2. \tag{2.15}$$

Other FD systems consider a FD-BS and multi-HD DL and uplink (UL) users. As illustrated in Fig. 2.3b, the RIS assists the communication from the FD-BS to a set $\mathcal{K} \triangleq \{1, \cdots, K\}$ of $K = |\mathcal{K}|$ DL users and from a set $\mathcal{L} \triangleq \{1, \cdots, L\}$ of $L = |\mathcal{L}|$ UL users to the FD-BS. The signals received by DL users and the FD-BS are respectively given as

$$y_{\mathrm{DL},k} = \left(\mathbf{h}_{\mathrm{R},k}^H \boldsymbol{\Theta} \mathbf{H}_{\mathrm{BR}}^H + \mathbf{h}_{\mathrm{B},k}^H \right) \sum_{i \in \mathcal{K}} \mathbf{w}_i x_i + \sum_{\ell \in \mathcal{L}} \left(g_{\ell k} + \mathbf{h}_{\mathrm{R},k}^H \boldsymbol{\Theta} \mathbf{g}_{\mathrm{R},\ell}^H \right) \sqrt{p_\ell} \tilde{x}_\ell + n, \tag{2.16}$$

and

$$y_{\mathrm{BS}} = \sum_{\ell \in \mathcal{L}} \left(\mathbf{g}_{\mathrm{B},\ell}^H + \mathbf{H}_{\mathrm{BR}} \boldsymbol{\Theta} \mathbf{g}_{\mathrm{R},\ell}^H \right) \sqrt{p_\ell} \tilde{x}_\ell + \left(\rho \mathbf{H}_{\mathrm{SI}} + \mathbf{H}_{\mathrm{BR}} \boldsymbol{\Theta} \mathbf{H}_{\mathrm{BR}}^H \right) \sum_{k \in \mathcal{K}} \mathbf{w}_k x_k + n. \tag{2.17}$$

Here, p_ℓ denotes the transmit power of the UL user. The k-th DL user and the ℓ-th UL user are denoted by $\mathsf{U}_k^{\mathtt{DL}}$, $\forall k \in \mathcal{K}$ and $\mathsf{U}_\ell^{\mathtt{UL}}$, $\forall \ell \in \mathcal{L}$, respectively. $\mathbf{H}_{\mathtt{BR}} \in \mathbb{C}^{N\times M}$, $\mathbf{h}_{\mathrm{B},k} \in \mathbb{C}^{N\times 1}$, $\mathbf{h}_{\mathrm{R},k} \in \mathbb{C}^{M\times 1}$, $\mathbf{g}_{\mathrm{B},\ell} \in \mathbb{C}^{1\times N}$, $\mathbf{g}_{\mathrm{R},\ell} \in \mathbb{C}^{1\times M}$ and $g_{\ell k} \in \mathbb{C}$ denote the channel matrices/vectors of BS-RIS, BS-$\mathsf{U}_k^{\mathtt{DL}}$, RIS-$\mathsf{U}_k^{\mathtt{DL}}$, $\mathsf{U}_\ell^{\mathtt{UL}}$-BS, $\mathsf{U}_\ell^{\mathtt{UL}}$-RIS and $\mathsf{U}_\ell^{\mathtt{UL}}$-$\mathsf{U}_k^{\mathtt{DL}}$ links, respectively. The SI channel matrix at the FD-BS is $\mathbf{H}_{\mathtt{SI}} \in \mathbb{C}^{N\times N}$. $\rho \in [0, 1)$ is the residual imperfect SI suppression level. Having explained the basic RIS-assisted system models, we provide applications of applying DRL into similar systems to optimize various communication targets in the following chapters.

References

1. ElMossallamy MA, Zhang H, Song L, Seddik KG, Han Z, Li GY (2020) Reconfigurable intelligent surfaces for wireless communications: principles, challenges, and opportunities. IEEE Trans Cogn Commun Netw 6(3):990–1002
2. Al-Nahhal I, Dobre OA, Basar E (2021) Reconfigurable intelligent surface-assisted uplink sparse code multiple access. IEEE Commun Lett 25(6):2058–2062
3. Alghamdi R, Alhadrami R, Alhothali D, Almorad H, Faisal A, Helal S, Shalabi R, Asfour R, Hammad N, Shams A, Saeed N, Dahrouj H, Al-Naffouri TY, Alouini M-S (2020) Intelligent surfaces for 6G wireless networks: a survey of optimization and performance analysis techniques. IEEE Access 8:202795–202818
4. Basar E, Di Renzo M, Rosny J, Debbah M, Alouini M-S, Zhang R (2019) Wireless communications through reconfigurable intelligent surfaces. IEEE Access 7:116 753–116 773
5. Liaskos C, Nie S, Tsioliaridou A, Pitsillides A, Ioannidis S, Akyildiz I (2018) Realizing wireless communication through software-defined hypersurface environments. In: Proceedings of the IEEE 19th International Symposium on "A World of Wireless, Mobile and Multimedia Netw." (WoWMoM), June 2018, pp 14–15
6. Faisal KM, Choi W (2022) Machine learning approaches for reconfigurable intelligent surfaces: a survey. IEEE Access 10:27 343–27 367
7. Allu R, Taghizadeh O, Singh SK, Singh K, Li C-P (2023) Robust beamformer design in active RIS-assisted multiuser MIMO cognitive radio networks. IEEE Trans Cogn Commun Netw 9(2):398–413
8. Wu X, Ma J, Xue X (2022) Joint beamforming for secure communication in RIS-assisted cognitive radio networks. J Commun Netw 2(5):518–529
9. Ge Y, Fan J (2023) Active reconfigurable intelligent surface assisted secure and robust cooperative beamforming for cognitive satellite-terrestrial networks. IEEE Trans Veh Technol 72(3):4108–4113
10. Allu R, Singh SK, Taghizadeh O, Singh K, Lit C-P (2022) Energy-efficient precoder design in RIS-assisted multiuser MIMO cognitive radio networks. In: GLOBECOM 2022 - 2022 IEEE Global Communications Conference, December 2022, pp 3338–3343
11. Agrawal N, Bansal A, Singh K, Li C-P (2022) Performance evaluation of RIS-assisted UAV-enabled vehicular communication system with multiple non-identical interferers. IEEE Trans Intell Trans Syst 23(7):9883–9894
12. Zhang Q, Zhao Y, Li H, Hou S, Song Z (2022) Joint optimization of star-RIS assisted UAV communication systems. IEEE Wirel Commun Lett 11(11):2390–2394
13. Zhang H, Huang M, Zhou H, Wang X, Wang N, Long K (2023) Capacity maximization in RIS-UAV networks: a DDQN-based trajectory and phase shift optimization approach. IEEE Trans Wirel Commun 22(4):2583–2591
14. Yu Y, Liu X, Liu Z, Durrani TS (2023) Joint trajectory and resource optimization for RIS assisted UAV cognitive radio. IEEE Trans Veh Technol 1–6

15. Ren H, Zhang Z, Peng Z, Li L, Pan C (2023) Energy minimization in RIS-assisted UAV-enabled wireless power transfer systems. IEEE Internet Things J 10(7):5794–5809
16. Ren J, Lei X, Peng Z, Tang X, Dobre OA (2023) RIS-assisted cooperative NOMA with SWIPT. IEEE Wirel Commun Lett 12(3):446–450
17. Chen Z, Tang J, Zhao N, Liu M, So DKC (2023) Hybrid beamforming with discrete phase shifts for RIS-assisted multiuser SWIPT system. IEEE Wirel Commun Lett 12(1):104–108
18. Lyu W, Xiu Y, Zhao J, Zhang Z (2023) Optimizing the age of information in RIS-aided SWIPT networks. IEEE Trans Veh Technol 72(2):2615–2619
19. Wu Q, Zhang R (2020) Beamforming optimization for wireless network aided by intelligent reflecting surface with discrete phase shifts. IEEE Trans Commun 68(3):1838–1851
20. Zhou G, Pan C, Ren H, Wang K, Renzo MD, Nallanathan A (2020) Robust beamforming design for intelligent reflecting surface aided MISO communication systems. IEEE Wirel Commun Lett 9(10):1658–1662
21. Faisal A, Al-Nahhal I, Dobre OA, Ngatched TMN (2021) Deep reinforcement learning for optimizing RIS-assisted HD-FD wireless systems. IEEE Commun Lett 25(12):3893–3897
22. Peng Z, Zhang Z, Pan C, Li L, Swindlehurst AL (2021) Multiuser full-duplex two-way communications via intelligent reflecting surface. IEEE Trans Signal Process 69:837–851
23. Faisal A, Al-Nahhal I, Dobre OA, Ngatched TMN (2022) Deep reinforcement learning for RIS-assisted FD systems: single or distributed RIS? IEEE Commun Lett 26(7):1563–1567

Chapter 3
Applications of RL for Continuous Problems in RIS-Assisted Communication Systems

3.1 Application 1: Maximizing Sum Rate

The work in [1] investigated the sum rate maximization problem of an RIS-assisted two-way MISO system while optimizing the continuous RIS phase shifts and beamformers. Two deployment schemes, namely single and distributed RIS, are explored based on the quality of the links. The problem is formulated as follows

$$\text{(P1)} \quad \max_{\mathbf{w}_k, \bar{\mathbf{\Theta}}} \quad \sum_{k=1}^{2} \mathcal{R}_k \tag{3.1a}$$

$$\text{s.t.} \quad -\pi \leq \varphi_{rn} \leq \pi, \; n = 1, \cdots, N_r \tag{3.1b}$$

$$||\mathbf{w}_k||^2 \leq P_{\max}, \; k = 1, 2. \tag{3.1c}$$

where the sum rate is expressed as

$$\mathcal{R}_k = \log_2 (1 + \gamma_k), \tag{3.2}$$

and γ_k is given by

$$\gamma_k = \frac{\left| \left(\sum_{r \in \Lambda} \mathbf{h}_{R_r S_k}^H \mathbf{\Theta}_r \mathbf{H}_{S_{\bar{k}} R_r} + \mathbf{h}_{S_{\bar{k}} S_k}^H \right) \mathbf{w}_{\bar{k}} \right|^2}{|\mathbf{h}_{S_k S_k}^H \mathbf{w}_k|^2 + \sigma^2},$$

$$k = 1, 2, \quad \Lambda = \begin{cases} 1 & \text{Single RIS} \\ 2 & \text{Distributed RIS.} \end{cases} \tag{3.3}$$

A. Faisal et al., *Reinforcement Learning for Reconfigurable Intelligent Surfaces*, SpringerBriefs in Computer Science, https://doi.org/10.1007/978-3-031-52554-4_3

Here, $\mathbf{w}_k$ denotes the beamforming vector and $\bar{\boldsymbol{\Theta}} = \text{diag}\big(\boldsymbol{\Theta}_1, \boldsymbol{\Theta}_2\big)$ is a matrix that contains the phase shifts of the two RISs for the distributed RIS. In the single RIS scenario, $\bar{\boldsymbol{\Theta}} = \text{diag}\big(\boldsymbol{\Theta}_1\big)$. The DDPG algorithm was deployed to optimize the RIS phase shifts for two deployment schemes (i.e., single and distributed RIS). On the other hand, the beamformers optimization was addressed using closed-form solutions as in (2.8) since it can be derived for single-user systems. The RL formulation is given as

3.1.1 Action Space

The action space represents the decision parameters. In this case, the aim is to maximize the sum rate, $\mathcal{R}_k$, by optimizing the continuous RIS phase shifts and beamforming vectors. Therefore, the action space contains the RIS phase shifts for both RISs and is defined as

$$a_t = \left[\varphi_{r1}^{(t)}, \cdots, \varphi_{rn}^{(t)}, \cdots, \varphi_{rN_r}^{(t)}\right]. \tag{3.4}$$

On the other hand, the beamforming vectors optimization was handled using closed-form solutions.

3.1.2 State Space

The state space should be carefully designed to provide the DDPG agent with the information it needs to learn the optimal decisions. In this case, the state space includes $\varphi_{rn} \forall\, n = 1, \cdots, N_r$ and the corresponding sum rate, $\mathcal{R}_k$, at time step $t-1$, expressed as

$$s_t = \left[\sum_{k=1}^{2} \mathcal{R}_k^{(t-1)}, \varphi_{r1}^{(t-1)}, \cdots, \varphi_{rn}^{(t-1)}, \cdots, \varphi_{rN_r}^{(t-1)}\right]. \tag{3.5}$$

3.1.3 Reward

The reward space was simply designed to represent the objective function, which is maximizing the sum rate, expressed as

$$r_t = \sum_{k=1}^{2} \mathcal{R}_k^{(t)}. \tag{3.6}$$

Furthermore, as explained in Chap. 1, searching for the action that maximizes the Q-function can be computationally expensive. Therefore, the action value is approximated directly by the actor network, while the critic network evaluates the chosen action by approximating the Q-values. To this end, the actor network output should be rescaled to reflect on the actual range of RIS phase shifts (i.e., $\varphi_{rn} \in [-\pi, \pi)$. Figure 3.1 shows the structure of the deployed DDPG algorithm to optimize the continuous RIS phase shifts based on Algorithm 3. The simulation results demonstrated the performance of the two deployment schemes based on the link quality. Figure 3.2 illustrates that the DDPG algorithm provides a significant improvement in the sum rate for the single and distributed RIS schemes compared to the random RIS phase shifts, which proves the credibility of applying the DDPG to optimize the continuous action space in RIS-assisted wireless systems.

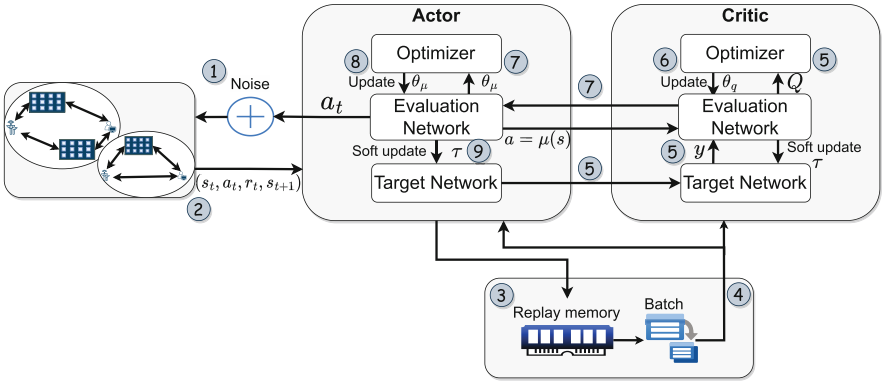


Fig. 3.1 DDPG algorithm structure

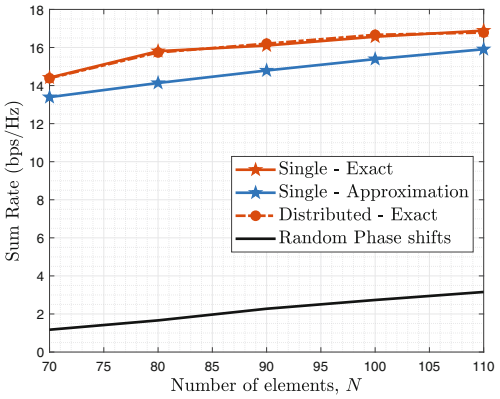


Fig. 3.2 The impact of varying N on the system performance

3.2 Application 2: Maximizing the Weighted Sum Rate

The work in [2] considered an extended problem where the aim was to maximize the weighted sum rate of a multi-user FD distributed RIS-assisted system (FD-BS, UL, and DL users). Given the considered practical system model, the optimization problem contains many continuous decision parameters. The work considered the joint optimization of RIS phase shifts, transmit and receive beamforming vectors, respectively, $\mathbf{w}_T$ and $\mathbf{w}_R$, and transmit power of BS and UL, p_B and p_u, respectively. To this end, the optimization problem is formulated as follows

$$\text{(P2)} \quad \max_{\bar{\mathbf{\Theta}}, \mathbf{w}_T, \mathbf{w}_R, p_b, p_u} \quad \rho \log_2(1+\gamma_{\text{BS}}) + (1-\rho)\log_2(1+\gamma_{\text{DL}}) \tag{3.7a}$$

$$\text{s.t.} \quad |\varphi_{rn}| = 1,\ r = 1, 2,\ n = 1, \cdots, N_r \tag{3.7b}$$

$$p_b \geq 0,\ p_u \geq 0, \tag{3.7c}$$

where γ_{BS} and γ_{DL} represent the SINR at the BS and DL user, respectively, and ρ regulates the weights of UL and DL rates in the optimization objective. The RL formulation is given as follows

3.2.1 Action Space

The decision parameters include the transmit and receive beamforming vectors, the RIS phase shifts, and transmit power at the BS and UL user. Since the NN can only handle real values, the beamformers values are separated into real and imaginary parts as independent input ports. The action space is presented as

$$\begin{aligned} a_t = [&\varphi_{r1}^{(t)}, \cdots, \varphi_{rn}^{(t)}, \cdots, \varphi_{rN_r}^{(t)}, \text{real}(w_{T1}^{(t)}, \cdots, w_{TM}^{(t)}), \text{imag}(w_{T1}^{(t)}, \cdots, w_{TM}^{(t)}), \\ &\text{real}(w_{R1}^{(t)}, \cdots, w_{RM}^{(t)}), \text{imag}(w_{R1}^{(t)}, \cdots, w_{RM}^{(t)}), p_b^{(t)}, p_u^{(t)}]. \end{aligned} \tag{3.8}$$

The actor network output is scaled according to the actions range. In particular, the RIS phase shifts are shifted to take values in the range $\varphi_{rn} \in [0, 2\pi)$ before calculating the reward. Furthermore, the in-phase, I, and quadrature, Q, parts of the beamformers are followed by the *tanh* activation function. Therefore, they can be represented as

$$\begin{aligned} \mathbf{a}_I &= \tanh(\mathbf{W}_{2,I}\text{ReLu}(\mathbf{W}_{1,I}\mathbf{x} + \mathbf{b}_{1,I}) + \mathbf{b}_{2,I}), \\ \mathbf{a}_Q &= \tanh(\mathbf{W}_{2,Q}\text{ReLu}(\mathbf{W}_{1,Q}\mathbf{x} + \mathbf{b}_{1,Q}) + \mathbf{b}_{2,Q}), \end{aligned} \tag{3.9}$$

where $\mathbf{W}$, $\mathbf{b}$ and $\mathbf{x}$ denote the NN weights, biases, and features passed via the subnetworks, respectively. The output is then normalized to have a unit norm.

Finally, the transmit power output from the sub-network is also shifted and scaled to the range $[0, P_B]$ and $[0, P_u]$, where P_B and P_u respectively denote the maximum transmit power of the BS and UL user, before using them in the environment.

3.2.2 State Space

The state of the environment includes the decision parameters of (P2) along with their evaluation (i.e., SINR of BS and DL user), at time step $t-1$, and is formulated as

$$s_t = [\gamma_{\text{BS}}^{(t-1)}, \gamma_{\text{DL}}^{(t-1)}, \varphi_{r1}^{(t-1)}, \cdots, \varphi_{rn}^{(t-1)}, \cdots, \varphi_{rN_r}^{(t-1)}, \text{real}(w_{T1}^{(t-1)}, \cdots, w_{TM}^{(t-1)}), \text{imag}(w_{T1}^{(t-1)}, \cdots, w_{TM}^{(t-1)}), \text{real}(w_{R1}^{(t-1)}, \cdots, w_{RM}^{(t-1)}), \text{imag}(w_{R1}^{(t-1)}, \cdots, w_{RM}^{(t-1)}), p_b^{(t-1)}, p_u^{(t-1)}]. \quad (3.10)$$

3.2.3 Reward

Since the optimization objective is to maximize the weighted sum of the UL and DL data rate, the reward function is assigned as the objective in (3.7a). It is showed that the DDPG algorithm is capable of predicting the RIS phase shifts, beamformers, and transmit power in a FD scenario efficiently without the need for channel state information (CSI). The DDPG algorithm nearly achieved the performance of semi-oracle DRL methods, where the beamformers were optimized using methods that require CSI.

3.3 Application 3: Maximizing the Location-Based Achievable Rate

The work in [3] investigated the joint design of beamformers and RIS phase shifts in RIS-assisted millimeter wave (mmWave) multiple-input-multiple-output wireless communications. The paper exploited a DRL algorithm that utilizes the location-aware imitation environment network (IEN). It considered a practical simulation setting, where the DDPG algorithm only uses the readily-available user location information, without the need for accurate CSI.

The aim of the paper is to maximize the achievable rate of RIS-assisted mmWave system. The problem is formulated as follows

$$\text{(P3)} \quad \max_{\bar{\boldsymbol{\Theta}},\mathbf{Q}} \quad R = \log_2 \det(\mathbf{I} + \frac{\bar{\mathbf{H}}\mathbf{Q}\bar{\mathbf{H}}^H}{\sigma^2}) \tag{3.11a}$$

$$\text{s.t.} \quad -\pi \leq \varphi_{rn} \leq \pi,\ n = 1, \cdots, N_r \tag{3.11b}$$

$$tr(\mathbf{Q}) \leq p, \mathbf{Q} \geq 0, \tag{3.11c}$$

where $\mathbf{Q}$, p, and $\mathbf{H}$ denote the transmit signal covariance matrix, transmit power, and reflected channel, respectively. To address the non-convex problem, a DNN is first built to imitate the actual transmission environment. It is worth noting that the aim of the IEN is to learn the composite reflected channel from the location information. After training the network successfully, the predicted reflected channel can be used to obtain the predicted achievable rate. Based on the predicted achievable rate, the DDPG is developed to predict the RIS phase shifts and beamforming values. The RL formulation is expressed as follows

3.3.1 Action Space

In this problem, the action is determined by the optimization variables, which include the RIS phase shifts and transmit signal covariance matrix. The real and imaginary parts of both variables are passed independently to the DNN. To this end, the action space at time step t is expressed as

$$a_t = \Big[\text{vec}\left(\text{real}(\mathbf{Q}^{(t)}), \text{imag}(\mathbf{Q}^{(t)})\right), \text{real}(\varphi_1^{(t)}, \cdots, \varphi_n^{(t)}, \cdots, \varphi_N^{(t)}), \\ \text{imag}(\varphi_1^{(t)}, \cdots, \varphi_n^{(t)}, \cdots, \varphi_N^{(t)})\Big]. \tag{3.12}$$

3.3.2 State Space

The state is determined by the information that the agent needs in the learning process. It contains the RIS phase shifts, the transmit covariance matrix, the achievable rate, and the location of the BS, RIS, and UE, and is formulated as follows

$$s_t = \Big[\text{vec}\left(\text{real}(\mathbf{Q}^{(t-1)}), \text{imag}(\mathbf{Q}^{(t-1)})\right), \text{real}(\varphi_1^{(t-1)}, \cdots, \varphi_n^{(t-1)}, \cdots, \varphi_N^{(t-1)}), \\ \text{imag}(\varphi_1^{(t)}, \cdots, \varphi_n^{(t-1)}, \cdots, \varphi_N^{(t-1)}), \text{loc(BS)}, \text{loc(RIS)}, \text{loc(UE)}\Big]. \tag{3.13}$$

Note that loc(BS), loc(RIS), and loc(UE) consist of 3D coordinates tuple of the BS, RIS, and UE locations, respectively.

3.3.3 Reward

The reward reflects the objective of the optimization problem, and is represented by the achievable sum rate given in (3.11a). The results established that the DDPG algorithm interacts with the IEN effectively and saves more time slots for data transmission, while conventional algorithms require separate signal transmission slots to interact with the actual environment.

3.4 Application 4: Maximizing the Energy Efficiency

The work in [4] considered an RIS-assisted DL multi-UAV wireless system, where each UAV is serving a cluster of UEs. The main goal of the paper is to maximize the energy efficiency by jointly optimizing the power allocation of the UAVs and the phase shift matrix of the RIS. To achieve this, the paper introduced a DRL approach that can handle time-varying channels in a centralized fashion. Additionally, a parallel learning approach is suggested to reduce latency in information transmission. In particular, most of the previous works in RIS-assisted UAV communications assumed unrealistic conditions, such as perfect CSI and static users. Additionally, the delay introduced by mathematical models and centralized learning is impractical for real-time applications. To address these limitations, the proposed DRL algorithms optimize power allocation and phase shifts in a joint manner and in real-time, which enhances the energy efficiency performance efficiently. The flexible and autonomous capabilities of UAVs and RIS, facilitated by trained NN, enable prompt decision-making and continuous learning to adapt to dynamic environments. Since the power and RIS phase shifts are considered to be continuous, a DDPG algorithm is developed to optimize the energy efficiency. Furthermore, parallel learning is used for training the model to reduce the delay when communicating the action between UAV and the RIS.

To this end, the problem is formulated as follows

$$\text{(P4)} \quad \max_{\boldsymbol{\Theta},\mathbf{P}} \quad \frac{\sum_{k=1}^{K}\sum_{m=1}^{M} R_{km}^{t}}{\sum_{k=1}^{K} P_k + P_n + P_c} \tag{3.14a}$$

$$\text{s.t.} \quad -\pi \leq \varphi \leq \pi,\ n = 1, \cdots, N \tag{3.14b}$$

$$0 \leq P_k \leq P_{\max}, \forall k \in K, \tag{3.14c}$$

where K, M, and N represent the number of UAVs, UEs, and RIS elements, respectively. The denominator of the objective function represents the total power consumption of the system. R_{km}^{t} denotes the throughput at the mth UE in the kth cluster at time step t, and is expressed as

$$R_{nm}^{t} = B \log_2 \left(1 + \gamma_{km}^{t}\right), \tag{3.15}$$

where B is the bandwidth and γ_{km}^t is the received SINR at the mth UE in the cluster l at time step t, expressed as

$$\gamma_{km}^t = \frac{P_k^t \left| H_{k,\mathrm{RIS}}^t \boldsymbol{\Theta}^t h_{\mathrm{RIS},km}^t \right|^2}{\sum_{i \neq k}^K P_i^t \left| H_{i,\mathrm{RIS}}^t \boldsymbol{\Theta}^t h_{\mathrm{RIS},im}^t \right|^2 + \alpha^2}. \tag{3.16}$$

3.4.1 Action Space

The action space contains the optimization parameters. Since the work aims to jointly optimize the RIS phase shifts and power allocation at the UAVs, the action space is expressed as follows

$$a_t = \left[\varphi_1^{(t)}, \cdots, \varphi_n^{(t)}, \cdots, \varphi_N^{(t)}, P_1, P_2, \cdots, P_K\right]. \tag{3.17}$$

3.4.2 State Space

The state space is designed in a more practical fashion, where it only contains the information available to the agent. The local information represents the reflected channel gains, defined as follows

$$s_t = \left[H_{1,\mathrm{RIS}} \boldsymbol{\Theta} h_{\mathrm{RIS},11}, H_{1,\mathrm{RIS}} \boldsymbol{\Theta} h_{\mathrm{RIS},12}, \ldots, H_{n,\mathrm{RIS}} \boldsymbol{\Theta} h_{\mathrm{RIS},nm}, \ldots, H_{N,\mathrm{RIS}} \boldsymbol{\Theta} h_{\mathrm{RIS},NM}\right]. \tag{3.18}$$

3.4.3 Reward

The reward function is designed to maximize the energy efficiency of the system. Therefore, the reward is expressed as (3.14a). With the above formulation, the DDPG is applied to optimize the system performance. Numerical results demonstrated the effectiveness of the DRL in solving joint optimization problems with dynamic environmental conditions and time-varying CSI, which proves the practicality of the DRL approaches in different scenarios.

3.5 Application 5: Maximizing the Secrecy Rate

Given the broadcast nature of wireless communication, the transmitted information is vulnerable to eavesdropping. Hence, ensuring physical layer security is crucial in wireless communication and has gained significant attention in recent years. RISs have emerged as an effective approach with low power consumption to enhance the system security. By leveraging the RIS, the signal toward the intended receivers can be strengthened while simultaneously weakening the signal to unauthorized eavesdroppers. The work in [5] considered an RIS-assisted FD communication system with multiple legitimate users and multiple eavesdroppers. The target is to maximize the secrecy success rate while jointly optimizing the transmit beamforming at the BS and phase shifts at the RIS. Since this optimization problem is complex and non-convex, an DRL approach is proposed to efficiently solve the optimization problem without relying on complex mathematical formulations. The work further considered a practical simulation setting, where both the transceiver and the RIS hardware impairments are assumed. DRL provides the flexibility to dynamically reconfigure the RIS phase shifts in real-time. This dynamic adaptation allows the RIS to counteract evolving security threats or changing environmental conditions. By continuously optimizing the phase shifts based on the CSI, RIS can adaptively enhance the security rate by adjusting the signal paths and mitigating potential vulnerabilities.

To this end, the optimization problem is formulated as follows

$$\begin{aligned} \text{(P5)} \quad & \max_{\boldsymbol{\Theta},\mathbf{W}} \quad C(\mathbf{W}, \boldsymbol{\Theta}, \boldsymbol{\varphi}, \mathcal{H}) && \text{(3.19a)} \\ & \text{s.t.} \quad -\pi \leq \varphi \leq \pi,\ n = 1, \cdots, N && \text{(3.19b)} \\ & \qquad \mathrm{Tr}(\mathbf{W}\mathbf{W}^H) \leq P_{\max}, && \text{(3.19c)} \end{aligned}$$

where $C(\mathbf{W}, \boldsymbol{\Theta}, \boldsymbol{\varphi}, \mathcal{H})$ is the secrecy success rate. It is worth noting that conventional optimization algorithms, such as alternating optimization (AO) and block coordinate descent are suitable for single-time slot optimization problems. However, they disregard historical data and the long-term advantages of the system, often leading to suboptimal solutions or performances akin to greedy-search. Consequently, applying traditional optimization techniques to attain satisfactory secure beamforming in uncertain and dynamic environments is generally impractical. To this end, the DDPG technique is leveraged to handle the optimization problem since the action space is continuous. In order to fulfill the constraints of (P5), a batch normalization layer is employed after the output of the actor network to output the actions in the feasible range. The RL formulation is given as follows

3.5.1 Action Space

The action space is composed of the continuous optimization variables, including the transmit beamforming matrix and the RIS phase shifts, and is given as

$$a_t = \left[\varphi_1^{(t)}, \cdots, \varphi_n^{(t)}, \cdots, \varphi_N^{(t)}, \mathbf{w}_1^{(t)}, \mathbf{w}_2^{(t)} \cdots, \mathbf{w}_k^{(t)}\right]. \tag{3.20}$$

3.5.2 State Space

The state space contains the cascaded channels (i.e., BS-RIS-legitimate users channel and BS-RIS-eavesdroppers channel), phase noise, transmit power of the BS, received power of the legitimate users, actions, sum rate at the legitimate users, sum rate at the BS, sum rate at the eavesdroppers, and the secrecy success rate. It is formulated as

$$\begin{aligned} s_t = \Big[\varphi_1^{(t-1)}, \cdots, \varphi_n^{(t-1)}, \cdots, \varphi_N^{(t-1)}, \mathbf{w}_1^{(t-1)}, \mathbf{w}_2^{(t-1)} \cdots, \mathbf{w}_k^{(t-1)}, \\ ||\mathbf{w}_k^{(t-1)}||^2, |\mathbf{G}_{1d,k}|^2, \mathbf{G}_{1d}, \mathbf{G}_{2d}, \delta\varphi\Big]. \end{aligned} \tag{3.21}$$

where $||\mathbf{w}_k^{(t-1)}||^2$ and $|\mathbf{G}_{1d,k}|^2$ represent the transmit power of the BS and the received power of the legitimate users, respectively. $\mathbf{G}_{1d}$, $\mathbf{G}_{2d}$, and $\delta\varphi$ denote the BS-RIS-legitimate users channel, BS-RIS-eavesdroppers channel, and phase noise, respectively.

3.5.3 Reward

The target of the optimization problem is to maximize the secrecy success rate. Thus, the reward function is designed as in (3.19a). Extensive simulation results demonstrated the significant effectiveness of the proposed DDPG algorithm in improving the secrecy success rate. The agent gradually enhances its action policy based on the reward received from the environment to achieve near-optimal transmit beamforming and phase shifts. Additionally, since these results were achieved while assuming several impairments, it proves that the DRL can be conveniently deployed in communication systems with various settings.

References

1. Faisal A, Al-Nahhal I, Dobre OA, Ngatched TMN (2022) Deep reinforcement learning for RIS-assisted FD systems: single or distributed RIS? IEEE Commun Lett 26(7):1563–1567
2. Nayak N, Kalyani S, Suraweera HA (2022) A DRL approach for RIS-assisted full-duplex UL and DL transmission: beamforming, phase shift and power optimization. December 2022 [Online]. Available: https://arxiv.org/abs/2212.13854
3. Xu W, An J, Huang C, Gan L, Yuen C (2022) Deep reinforcement learning based on location-aware imitation environment for RIS-aided mmWave MIMO systems. IEEE Wirel Commun Lett 11(7):1493–1497
4. Nguyen KK, Khosravirad SR, da Costa DB, Nguyen LD, Duong TQ (2022) Reconfigurable intelligent surface-assisted multi-UAV networks: efficient resource allocation with deep reinforcement learning. IEEE J Sel Top Signal Process 16(3):358–368
5. Peng Z, Zhang Z, Kong L, Pan C, Li L, Wang J (2022) Deep reinforcement learning for RIS-aided multiuser full-duplex secure communications with hardware impairments. IEEE Internet Things J 9(21):21 121–21 135.

Chapter 4
Applications of RL for Discrete Problems in RIS-Assisted Communication Systems

4.1 Application 6: Maximizing Sum Rate

To fully realize the RIS capabilities and overcome the probabilistic propagation environment, the RIS phase shifts should be optimized efficiently. The majority of the literature works assume an ideal phase shift model (i.e., continuous values). However, this assumption is infeasible to implement due to hardware limitations. Furthermore, optimizing continuous phase shifts for a large number of RIS elements can be computationally intensive, requiring significant processing power and memory. Therefore, current practical approaches for RIS optimization consider discrete phase shifts values instead of continuous. Discrete phase shifts refer to the case where the phase of each RIS element can take only a limited number of predetermined values. These values are typically chosen from a fixed set of discrete phases depending on a predefined resolution, which is a function of the number of quantization bits used. For example, if the considered resolution uses 2 quantization bits, the RIS phase shifts will have four possible values from the following set $\left(\text{i.e.,}\{-\pi, -\frac{\pi}{2}, 0, \frac{\pi}{2}\}\right)$. This discrete nature of phase shifts simplifies the design and implementation of RIS systems, as it requires fewer bits to represent the phase states of the elements.

The work in [1] considered optimizing the finite-level phase shifts using an efficient DQL approach. It investigated a distributed RIS-assisted two-way MISO system, where the aim was to maximize the system sum rate, formulated as

$$\text{(P6)} \quad \max_{\mathbf{w}_k, \bar{\boldsymbol{\Theta}}} \quad \sum_{k=1}^{2} \mathcal{R}_k \tag{4.1a}$$

$$\text{s.t.} \quad \varphi_{rn} \in \Upsilon,\ r = 1, 2,\ n = 1, \cdots, N_r \tag{4.1b}$$

$$||\mathbf{w}_k||^2 \leq P_{\max},\ k = 1, 2, \tag{4.1c}$$

A. Faisal et al., *Reinforcement Learning for Reconfigurable Intelligent Surfaces*, SpringerBriefs in Computer Science, https://doi.org/10.1007/978-3-031-52554-4_4

where $\bar{\boldsymbol{\Theta}} = \left(\text{diag}\left(\boldsymbol{\Theta}_1, \boldsymbol{\Theta}_2\right)\right) \in \mathbb{C}^{N_r \times N_r}$, contains the phase shifts of the two RISs, $P_{\max}$ is the maximum transmitted power of $S_{\bar{k}}$, and $\mathcal{R}_k$ is the sum rate in bit per second per Hertz (bps/Hz), expressed as

$$\mathcal{R}_k = \log_2 \left(1 + \frac{\left| \left(\sum_{r=1}^{2} \mathbf{h}_{R_r S_k}^H \boldsymbol{\Theta}_r \mathbf{H}_{S_{\bar{k}} R_r} + \mathbf{h}_{S_{\bar{k}} S_k}^H \right) \mathbf{w}_{\bar{k}} \right|^2}{|\mathbf{h}_{S_k S_k}^H \mathbf{w}_k|^2 + \sigma^2} \right), \bar{k} = 3 - k \, \forall \, k \in \{1, 2\}. \tag{4.2}$$

Since the discrete phase shift model is assumed, the RIS phase shifts are chosen only from the following set

$$\Upsilon = \{0, \Delta\varphi, \cdots, \Delta\varphi(K-1)\}, \Delta\varphi = 2\pi/K, K = 2^b. \tag{4.3}$$

Here, b is the number of quantization bits that represent the RIS phase shift values. For example, if $b = 2$, then there are only four phase shift values available for each RIS element $\left(\text{i.e.,}\{-\pi, -\frac{\pi}{2}, 0, \frac{\pi}{2}\}\right)$.

The beamformers are optimized using closed-form derivations, presented in (2.8) and (2.11). The DQL is deployed to optimize the discrete RIS phase shifts, since it is a powerful algorithm that can effectively learn to make optimal decisions in discrete action spaces. It uses an NN to approximate the action-value function, which estimates the expected reward for each possible action in a given state. By selecting the action that maximizes the estimated reward, the agent can learn an optimal policy that leads to the highest total reward over time. To this end, the action, state spaces, and reward are formulated as

4.1.1 Action Space

The action space contains the optimization variable, which is the discrete phase shifts values.

$$a_t = \left[\varphi_{r1}^{(t)}, \cdots, \varphi_{rn}^{(t)}, \cdots, \varphi_{rN_r}^{(t)} \right]. \tag{4.4}$$

4.1.2 State Space

The state space includes $\varphi_{rn} \, \forall \, n = 1, \cdots, N_r$ and the corresponding sum rate, $\mathcal{R}_k$, at time step $t-1$, expressed as

$$s_t = \left[\sum_{k=1}^{2} \mathcal{R}_k^{(t-1)}, \varphi_{r1}^{(t-1)}, \cdots, \varphi_{rn}^{(t-1)}, \cdots, \varphi_{rN_r}^{(t-1)} \right]. \tag{4.5}$$

4.1.3 Reward

The reward represents the objective function, which is maximizing the sum rate, expressed as

$$r_t = \sum_{k=1}^{2} \mathcal{R}_k^{(t)}. \tag{4.6}$$

Figure 4.1 shows the structure of the deployed DQL algorithm to optimize the discrete RIS phase shifts, and illustrates the action selection criteria based on Algorithm 2. The DQL network outputs the Q-values for all the possible actions. Then, the maximum Q-values are selected to be mapped to the corresponding value of the discrete RIS phase shift set. This process is repeated until the agent reaches a convergence point.

Figure 4.2 demonstrates the performance of the DQL algorithm in terms of sum rate versus the number of quantization bits. The results are compared with the optimization of the continuous phase shift model using two beamforming mathematical solutions, as well as a scenario where no RIS is utilized (referred to as no-RIS). The findings indicate that as the number of quantization bits increases, the DQL algorithm achieves a performance level close to the upper bound represented

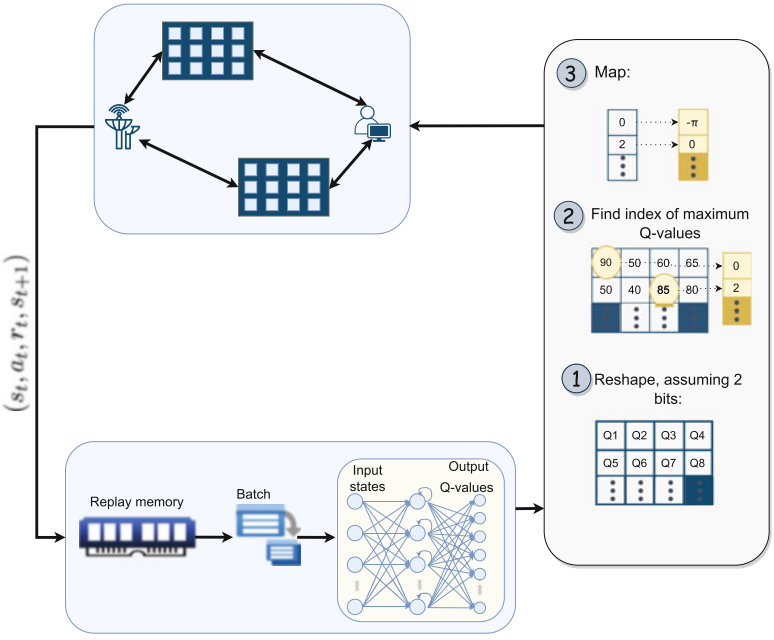


Fig. 4.1 DQL algorithm structure

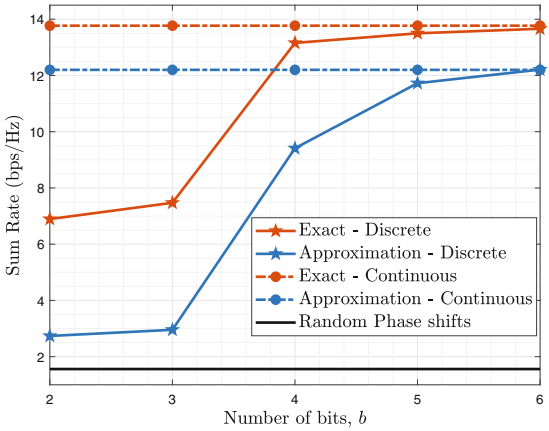


Fig. 4.2 The effect of the number of bits on the system performance for $N = 60$

by the continuous phase shift model, thereby demonstrating its practicality. This holds true for both approximate and closed-form beamforming solutions. It is worth noting that near-optimal solutions can be obtained using only 6 bits, proving the validity of the RL design.

4.2 Application 7: Minimizing System Resources

In some more complicated environments, the reward design might need to be scaled or balanced to contain positive and negative values, facilitating a simpler learning process for the agent. In particular, the work in [2] considered a practical distributed RIS-assisted FD wireless system. It jointly optimizes two discrete variables along with the beamformers to minimize the system resources. Optimizing the RIS states yields a remarkable improvement in the power consumption and overall complexity of the system. Furthermore, it enhances the system flexibility, where the RIS system can be reconfigured to adapt to environmental changes or system requirements. To this end, the aim was to minimize the system resources by jointly optimizing the discrete RIS phase shifts and their states (i.e., ON or OFF). The reflection coefficient matrix is expressed as

$$\boldsymbol{\Theta}_r = \text{diag}\left(\beta_{r1} e^{j\varphi_{r1}}, \cdots, \beta_{rn} e^{j\varphi_{rn}}, \cdots, \beta_{rN_r} e^{j\varphi_{rN_r}}\right), \tag{4.7}$$

where $\beta_{rn} \in \{0, 1\}$ is the amplitude reflection coefficient. If $\beta_{rn} = 0$, the n-th element of R_r is turned OFF. Otherwise, the n-th element is turned ON. The work considered the same system model presented in (P6), and the formulated problem is defined as

$$\text{(P7)} \quad \max_{\mathbf{w}_k, \bar{\mathbf{\Theta}}, \boldsymbol{\beta}} \quad \frac{\sum_{k=1}^{2} \mathcal{R}_k(\mathbf{w}_k, \bar{\mathbf{\Theta}})}{\sum_{i=0}^{N} \beta_i} \tag{4.8a}$$

$$\text{s.t.} \quad \mathcal{R}_k = \mathcal{R}_k^{\text{Target}}, \ k = 1, 2 \tag{4.8b}$$

$$\varphi_{rn} \in \Upsilon, \ \ n = 1, \cdots, N_r | r = 1, 2 \tag{4.8c}$$

$$\beta_{rn} \in \{0, 1\}, \ \ n = 1, \cdots, N_r | r = 1, 2 \tag{4.8d}$$

$$\mathbf{w}_k^T \mathbf{w}_k \leq P_{\max}, \ k = 1, 2, \tag{4.8e}$$

where $\boldsymbol{\beta} = \left[\beta_{r1}, \cdots, \beta_{rn}, \cdots, \beta_{rN_r}^{(t)}\right]$. Constraint (4.8b) ensures that the system is satisfying the target sum rate, $\mathcal{R}_k^{\text{Target}}$, while minimizing the system resources. Looking at the problem formulation, one can see that the action, state spaces, and reward should be approached differently than (P6). Specifically, the RL formulation is presented as

4.2.1 Action Space

Since (P7) aims to optimize the discrete decision parameters jointly, the action space includes the discrete RISs phase shifts and their states, and is expressed as

$$a_t = \left[\varphi_{r1}^{(t)}, \cdots, \varphi_{rn}^{(t)}, \cdots, \varphi_{rN_r}^{(t)}, \beta_{r1}^{(t)}, \cdots, \beta_{rn}^{(t)}, \cdots, \beta_{rN_r}^{(t)}\right]. \tag{4.9}$$

4.2.2 State Space

The state space is designed to describe the current configuration of the environment. It includes β_{rn} and φ_{rn} $\forall\, n = 1, \cdots, N_r$ and their evaluation, at time step $t-1$, and is expressed as

$$s_t = \left[\frac{\sum_{k=1}^{2} \mathcal{R}_k^{(t-1)}}{\sum_{i=0}^{N} \beta_i^{(t-1)}}, \varphi_{r1}^{(t-1)}, \cdots, \varphi_{rn}^{(t-1)}, \cdots, \varphi_{rN_r}^{(t-1)}, \beta_{r1}^{(t-1)}, \right.$$
$$\left. \cdots, \beta_{rn}^{(t-1)}, \cdots, \beta_{rN_r}^{(t-1)}\right]. \tag{4.10}$$

4.2.3 Reward

Since the goal is to minimize the system resources while satisfying the target sum rate, the reward function can not simply be designed as the objective in (4.8a). From the DQL agent's perspective, it aims to choose actions that maximize the cumulative reward, which could be guaranteed by turning most elements ON and exceeding the target sum rate. Therefore, the target rate constraint, (4.8b), should be reflected on the reward to guarantee finding the optimal number of activated elements that satisfy the system target rate. To this end, the reward is designed based on the punishment and encouragement mechanism. The agent receives negative rewards if it fails to satisfy the target rate constraint and improve the rate based on the previous time step feedback. The negative reward is weighted to prioritize the agent's learning responsibilities. In contrast, the agent receives a significant positive reward if it satisfies the target rate constraint and improves the rate based on the previous feedback. Furthermore, constraint (4.8b) is relaxed by introducing an upper bound, as it is nearly impossible to bound the agent to a restricted value only (i.e., $\mathcal{R}_k = \mathcal{R}_k^{\text{Target}}$). The reward design is summarized in Algorithm 4.

Algorithm 4 Reward design.

if $\mathcal{R}_k^{\max} > \mathcal{R}_k \geq \mathcal{R}_k^{\text{Target}}$ **then**
  $r = \frac{\sum_{k=1}^{2} \mathcal{R}_k^{(t)}}{\sum_{i=0}^{N} \beta_i^{(t)}} - \frac{\sum_{k=1}^{2} \mathcal{R}_k^{(t-1)}}{\sum_{i=0}^{N} \beta_i^{(t-1)}}$
  if $r > 0$ **then**
    $r_t = 1000$
  else
    $r_t = -100$
  end if
else
  $r_t = -1000$
end if

Figure 4.3 shows that the DQL algorithm effectively satisfies the target rate constraint with the reward function in Algorithm 4. Two target rate values are investigated to illustrate the flexibility of the deployed algorithm. In particular, in the early stages of learning, the agent is exploring lower sum rates through activating some of the RIS elements and receiving a negative reward based on Algorithm 4. After $K = 200$, it learns to converge to the target sum rate, reaching $\mathcal{R}_k = 8$ bps/Hz and $\mathcal{R}_k = 10.5$ bps/Hz. It further shows that the agent is able to save a considerable number of active RIS elements, and this reduction percentage has a huge positive impact on the system complexity in real scenarios.

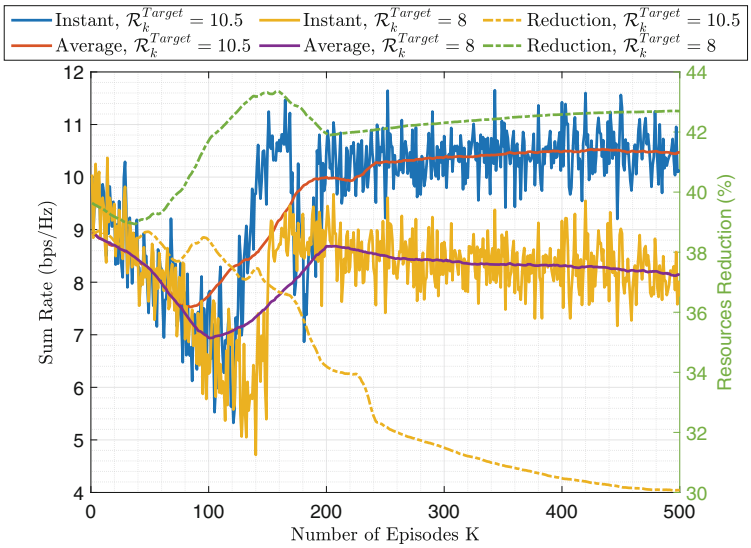


Fig. 4.3 Sum rate versus the number of episodes, K, at $N = 80$

4.3 Application 8: Maximizing the Energy Efficiency

As DQL uses NNs to approximate the optimal action-value function, it allows for considering larger action spaces, more efficient storage and retrieval of Q-values. The work in [3] further considered DQL to optimize three discrete decision parameters to maximize the average energy efficiency of an DL MISO RIS-assisted multi-user system. The parameters include the discrete RIS phase shifts, their states (i.e., ON or OFF), and the transmit power. The optimization problem is formulated as follows

$$\text{(P8)} \quad \max_{\boldsymbol{\sigma},\boldsymbol{\Theta},\boldsymbol{p}} \quad \lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T} \frac{R(t)}{P(t)} \tag{4.11a}$$

$$\text{s.t.} \quad S(t) \leq E(t), \forall t \tag{4.11b}$$

$$\sum_{k=1}^{K} p_k(t) \leq P_{\max}, \forall t. \tag{4.11c}$$

Here, $S(t)$ is the energy consumption at time slot t, $E(t)$ is the available amount of energy stored in the system at time slot t, $p_k(t)$ is the transmission power of the BS for user k at time t, and the power consumption for the wireless link between the BS and K users is denoted by $P(t)$. R represents the achievable sum rate of the RIS-assisted multi-user MISO system, which is given as follows

$$R = \sum_{k=1}^{K} \log_2 \left(1 + \frac{p_k \left| \boldsymbol{h}_k \hat{\boldsymbol{h}}_k^H \right|^2}{\sum_{\substack{j=1 \\ j \neq k}}^{K} p_j \left| \boldsymbol{h}_k \hat{\boldsymbol{h}}_j^H \right|^2 + \sigma_n^2} \right). \tag{4.12}$$

To this end, the action, state spaces, and reward are formulated as

4.3.1 Action Space

The action space includes the discrete RIS phase shifts, their states, and the transmit power, and is expressed as

$$a_t = \left[\boldsymbol{\Theta}^{(t)}, \boldsymbol{\beta}^{(t)}, \mathbf{P}^{(t)} \right]. \tag{4.13}$$

4.3.2 State Space

The state space includes the precoding vectors from the users and the energy level of the RIS, presented as follows

$$s_t = \left[\hat{\boldsymbol{h}}_k^{(t-1)}, E^{(t-1)} \right].$$

4.3.3 Reward

Since the goal is to maximize the energy efficiency of the RIS-assisted communication system, the reward is designed as the objective function, presented as

$$r_t = \sum_{k=1}^{k=K} \frac{\mathcal{R}_k^{(t)}}{\mathbf{P}^{(t)}}. \tag{4.14}$$

Using this formulation, the BS can determine its energy consumption, $P^{(t)}$, based on the system rate feedback. Practically, the reward value can be influenced by multiple unknown factors, including uncertainties regarding the users' channel conditions. To address this, the DNN leverages feedback from the uncertain environment and updates the weights of the expected reward for state-action pairs by computing the difference between expected and actual rewards. This results in the improvement of the DNN weights, allowing the BS to better estimate the expected reward for each state-action pair.

4.4 Application 9: Maximizing the Spectral Efficiency

The DQL can be applied to a wide range of applications, including advanced transmission schemes in wireless communications. In particular, the work in [4] focused on optimizing the discrete RIS phase shifts of an DL orthogonal frequency division multiplexing (OFDM) transmission system. To minimize hardware costs, a 1-bit resolution and column-wise controllable RIS are considered. Since finite resolution phase shifts are considered, the DQL is developed for the optimization problem.

OFDM is a widely used modulation scheme in modern digital communication systems. It is particularly suited for transmitting data over wireless channels that are prone to frequency-selective fading and interference. OFDM divides the available frequency band into multiple orthogonal subcarriers, allowing simultaneous transmission of multiple data streams. By enabling orthogonal subcarriers, the individual symbols do not interfere with each other, allowing for efficient utilization of the available bandwidth. OFDM provides several advantages, which make it favorable for various communication systems. Firstly, it is robust against frequency-selective fading caused by multipath propagation. Since the subcarriers are spaced closely together, the individual subcarriers experience different fading conditions. This allows for effective equalization at the receiver to mitigate the effects of multipath fading. Secondly, OFDM is resistant to narrowband interference. Since each subcarrier is orthogonal to others, narrowband interference affects only a small subset of the subcarriers, resulting in minimal degradation of the overall system performance. Furthermore, OFDM supports high data rates by allowing efficient spectrum utilization. By dividing the available bandwidth into numerous narrowband subcarriers, each subcarrier can transmit data at a lower symbol rate, enabling higher overall data rates.

Incorporating the RIS into OFDM systems enhances their overall performance. Specifically, the RIS can compensate for the frequency-selective fading and multipath propagation by tuning the phase shifts to add the received signals constructively for specific subcarriers. This helps equalize the channel and mitigate the effects of multipath fading. To this end, the work in [4] aims to maximize the broadband transmission system sum spectral efficiency of RIS-assisted OFDM systems by optimizing the discrete RIS phase shifts. The problem can be formulated as

$$\text{(P9)} \quad \max_{\boldsymbol{\Theta}} \quad \sum_{k=1}^{K} \sum_{i \in V_k} \log_2 \left(1 + p_i \frac{\left| \mathbf{h}_{k,i}^r \varphi + h_{k,i}^d \right|^2}{\kappa \sigma^2} \right), \tag{4.15a}$$

$$\text{s.t.} \quad \varphi_n = 0 \text{ or } \pi, n = 1, 2, \ldots, N. \tag{4.15b}$$

Here, V and K represent the number of subcarriers and users, respectively. $p_i = \frac{P_t}{V}$, where it is assumed that the BS distributes the subcarrier power evenly. $\kappa \geq 1$ is used to denote the difference in channel capacity with respect to the actual

system. The direct and reflected channels are denoted as h^d and $\mathbf{h}^r$, respectively. An efficient DQL algorithm is developed to optimize the RIS phase shifts with a reduced calculation delay as compared to traditional methods. To this end, the RL formulation is designed as follows

4.4.1 Action Space

The action space contains the discrete RIS phase shifts for each column of the RIS, given as

$$a_t = \left[\varphi_1^{(t)}, \cdots, \varphi_n^{(t)}, \cdots, \varphi_N^{(t)}\right]. \tag{4.16}$$

4.4.2 State Space

The state space is designed as the combination of the frequency domain reflection channel. By incorporating these channel characteristics, the DQL method can effectively capture the system structure.

$$s_t = \left[\text{real}(\mathbf{h}_{k,i}^{r(t-1)}), \text{imag}(\mathbf{h}_{k,i}^{r(t-1)}), \text{real}(h_{k,i}^{d(t-1}), \text{imag}(h_{k,i}^{d(t-1)})\right]_{i=0,1,\cdots,V-1}$$

4.4.3 Reward

Note that the objective of the developed system is to optimize the total spectral efficiency. Thus, the reward is set as the objective in (4.15a).

As the transmission scheme is advanced, a couple of improvement techniques are applied to the DQL algorithm. First, a target network is introduced to stabilize the learning performance. The DQL agent utilizes a soft update to update the target Q-network (i.e., similar to DDPG). The implementation of the soft update can effectively address the instability of the state-action network and expedite the convergence of the algorithm. Second, a search method is applied within a small range of potential phase shift vectors based on the output of the trained model. In particular, instead of choosing the action with the maximum Q-value directly, L actions are selected based on the maximum Q-value potential. Then, the agent computes the spectral efficiency for the selected L actions and chooses the action that leads to the maximum output. L is a hyperparameter that can effectively improve the spectral efficiency of the network at a relatively low extra time consumption. The improved structure of the DQL algorithm is shown in Fig. 4.4. The numerical results confirmed that the proposed algorithm achieves

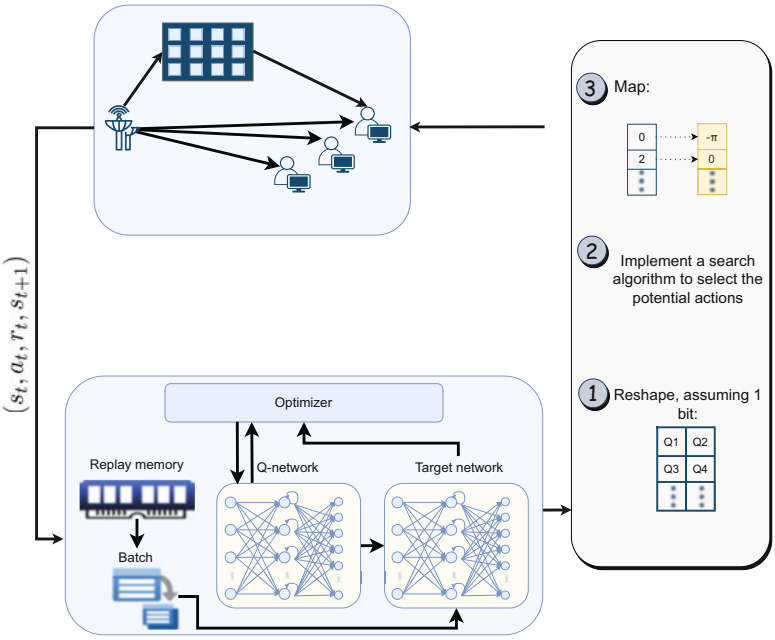


Fig. 4.4 Improved DQL algorithm structure

comparable spectral efficiency to the optimal approach (i.e., the exhaustive search) while requiring minimal time. Furthermore, the performance of the DQL algorithm is compared with traditional optimization algorithms, such as the discrete successive convex approximation algorithm, in terms of online running time and performance. It was proved that the discrete successive convex approximation algorithm exhibits significantly higher running times compared to the DQL algorithm. The proposed algorithm achieves better spectral efficiency, reaching the upper bound solution. Additionally, the proposed DQL algorithm can dynamically adjust the parameter L to strike a balance between time complexity and performance. This is of great importance and plays a key role in optimizing advanced RIS-assisted wireless communication systems.

4.5 Application 10: Maximizing the Minimum User Spectral Efficiency

The work in [5] further investigated a different objective function in an RIS-assisted DL OFDM transmission system. The target is to maximize the minimum user spectral efficiency to ensure fairness by jointly optimizing the beamformers and RIS discrete phase shifts. To achieve this, the BS employs maximum ratio transmission

to optimize the beamformers, while the dueling DQL framework is utilized for optimizing the RIS phase shifts. The optimization problem is formulated as follows

$$\begin{aligned} (\text{P10}) \quad & \max_{\boldsymbol{\Theta},\mathbf{W}} \quad \min_k \mathbf{R}_k && (4.17a) \\ & \text{s.t.} \quad \varphi_n = 0 \text{ or } \pi, n = 1, 2, \ldots, N && (4.17b) \\ & \quad\quad ||\mathbf{w}_v||^2 = 1, v = 1, 2, \ldots, V. && (4.17c) \end{aligned}$$

Here, $\mathbf{R}_k$ represents the spectral efficiency of user k. The dueling DQL (discussed in Chap. 1) is used rather than the traditional DQL algorithm to enable faster convergence and a stable learning process. The RL formulation is designed as follows

4.5.1 Action Space

The action space contains the phase shift vector of the RIS, which belong to a predefined discrete set

$$a_t = \left[\varphi_1^{(t)}, \cdots, \varphi_n^{(t)}, \cdots, \varphi_N^{(t)}\right]. \tag{4.18}$$

4.5.2 State Space

The state space represents the CSI, including direct and reflection channels, indicating the channel sample rate (i.e., q). Including a sample of the CSI reduces the input dimension of the network effectively, while still allowing the agent to learn channel characteristics. The state space is defined as

$$s_t = \left[\mathbf{h}_{i,q}^{(t-1)}, \mathbf{h}_{i,k}^{(t-1)}\right]_{i=0,1,\cdots,V-1}.$$

4.5.3 Reward

The reward is set as the minimum user spectral efficiency since the agent will aim to maximize this scalar value

$$r_t = \min_k \mathbf{R}_k^{(t)}.$$

The simulation results demonstrated that the proposed DRL-based algorithm achieves nearly optimal performance while requiring significantly less computational resources.

References

1. Faisal A, Al-Nahhal I, Dobre OA, Ngatched TMN (2022) Distributed RIS-assisted FD systems with discrete phase shifts: A reinforcement learning approach. In: GLOBECOM 2022 - 2022 IEEE Global Communications Conference, December 2022, pp 5862–5867
2. Faisal A, Al-Nahhal I, Dobre OA, Ngatched TMN (2023) On discrete phase shifts optimization of RIS-aided FD systems: are all RIS elements needed? In: 2023 IEEE International Conference on Communications Workshops (ICC Workshops), May 2023, pp 1–6
3. Lee G, Jung M, Kasgari ATZ, Saad W, Bennis M (2020) Deep reinforcement learning for energy-efficient networking with reconfigurable intelligent surfaces. In: ICC 2020 - 2020 IEEE International Conference on Communications (ICC), June 2020, pp 1–6
4. Chen P, Li X, Matthaiou M, Jin S (2023) DRL-based RIS phase shift design for OFDM communication systems. IEEE Wirel Commun Lett 12(4):733–737
5. Chen P, Zhang H, Li X, Jin S (2023) Drl-based max-min fair RIS discrete phase shift optimization for MISO-OFDM systems. J Inf Intell 1(3):281–293

Chapter 5
Challenges and Future Work

5.1 Challenges

Although DRL is promising in solving optimization problems in RIS-assisted systems, deploying DRL to these systems presents several unique challenges that need to be carefully addressed. We hereby discuss some of the common challenges, which include hyperparameter tuning, problem design, and complexity analysis.

5.1.1 Hyperparameter Tuning and Problem Design

DRL has shown outstanding performance in various RIS-assisted wireless applications. However, designing a proper DRL algorithm is a complex task that involves tuning several parameters carefully. These parameters play a crucial role in determining performance, convergence speed, and efficiency. The selection of appropriate parameters, such as the learning rate, discount factor, noise factor, and activation function can enhance the performance of DRL algorithms and accelerate the convergence to the near-optimal solution. However, the proper values of these parameters may vary depending on the problem design and the complexity of the wireless communication system. Automated hyperparameter tuning methods, such as grid search, random search, or more advanced methods like Bayesian optimization, can help, but these methods can also be computationally expensive [1].

Besides the extensive experimentation requirement, designing an appropriate reward function for the DRL agent is a crucial part of the problem design. The reward function must accurately reflect the goals of the system and encourage the agent to learn a useful policy. However, in RIS-assisted systems, defining an appropriate reward function can be challenging due to the complexity of the system and the various trade-offs involved (e.g., throughput, complexity, energy efficiency,

A. Faisal et al., *Reinforcement Learning for Reconfigurable Intelligent Surfaces*,
SpringerBriefs in Computer Science, https://doi.org/10.1007/978-3-031-52554-4_5

target rate, latency, etc.). Furthermore, the design of the state and action spaces is another critical aspect of the problem design. The state space should include all relevant information for decision-making, but including such information for RIS-assisted systems can make the state space too large and the learning problem more difficult [2]. Specifically, in complex RIS-assisted systems, including the CSI in the state space can improve the performance, but this assumption is not practical as the agent does not have access to accurate CSI all the time. Similarly, the action space should contain all relevant optimization variables. However, considering a large number of RIS reflecting elements might increase the time complexity of the DRL. In the action space design, DRL agents must balance exploration (i.e., testing new actions to gain information) and exploitation (i.e., employing the information they have to select the best action). The balance between exploration and exploitation is controlled by a hyperparameter, and finding the right balance can be tricky. Large exploration factor can lead to sub-optimal performance, while too small exploration can prevent the agent from learning a good policy. Balancing these trade-offs is a significant challenge.

5.1.2 Complexity Analysis

The DQL and DDPG are both powerful DRL algorithms often used for dealing with complex systems. When applied to RIS-assisted systems, there are several challenges related to the computational complexity that might arise. The DRL formulation of RIS-assisted systems might have large state and action spaces. The state might include CSI, power, location of users, RIS configuration, and any relevant information for the optimization problem. The action space includes the optimization variables and can be rapidly enlarged in the case of joint optimization. For DQL, which is typically used for discrete action spaces, a large action space can pose a significant challenge, as the algorithm needs to estimate the Q-value for every action in every state, which can be computationally expensive or infeasible for very large action spaces. The DDPG, on the other hand, is designed for continuous action spaces. However, it may still struggle with the complexity of the action space, as it relies on learning a policy network that can accurately map states to optimal actions. If the relationship between states and actions is very complex, learning this mapping can be a difficult task.

Furthermore, DRL algorithms such as the DQL and the DDPG often require a large number of samples (i.e., state, action, reward, and next state experience tuple) to learn effectively. In a complex RIS-assisted system, gathering these samples can be time-consuming, and the algorithm may require a large number of episodes to converge to an optimal policy. Additionally, both the DQL and the DDPG involve training an DNN, which can be computationally expensive. The time complexity of each training step depends on the size of the networks, the size of the minibatch used for training, and the complexity of the optimization algorithm used for training. In

the case of the DDPG, four DNN are used to train the agent, which involves more complex operations than the DQL.

Despite these challenges, the DQL and the DDPG have shown promise in handling complex optimization problems such as those in RIS-assisted systems. Improvements in techniques such as function approximation and experience replay can help manage the computational complexity and improve the learning efficiency. However, further research is needed to fully exploit the potential of these algorithms in this context and to develop methods that can handle the specific challenges posed by RIS-assisted systems. Despite these challenges, the potential benefits of the DRL for these systems make it an exciting area for future research.

5.2 Future Work

Although several papers studied the application of the DRL to optimization problems in RIS-assisted wireless communication systems, there are still many promising research directions to explore. Having thoroughly discussed the DRL application to RIS-assisted systems, we present some open challenges in this chapter.

5.2.1 Hybrid RL

One of the directions is to investigate the integration of DRL with other machine learning techniques, such as transfer learning and meta-learning, to improve generalization and adaptability across different wireless environments and scenarios. Hybrid DRL can combine DRL with supervised learning, unsupervised learning, or other RL techniques to leverage the strengths of multiple algorithms and tackle complex optimization problems in RIS-assisted wireless communication systems [3]. In this sub-section, we will explore two scenarios: Integrating DRL with a supervised learning algorithm and combining two DRL algorithms.

5.2.1.1 DRL and Supervised Learning

Supervised learning is a class of machine learning, where an algorithm learns based on a labeled training dataset to make predictions for unseen samples. In supervised learning, the training data consists of input-output pairs, where each input is associated with a label. The algorithm learns a function that approximates the relationship between the input features and output labels based on the provided training data. During the training phase, the supervised learning algorithm iteratively adjusts its internal parameters to minimize a loss function between the predicted outputs and the actual labels. Once the model is trained, it can be used

to make predictions or classifications for new, unseen data. Supervised learning has a wide range of applications in various wireless communication domains, such as channel estimation, source and channel coding, waveform design, and signal detection. Combining the DRL with supervised learning algorithms can create a sort of controlled environment, allowing the agent to learn in a guided approach, hence, speeding up the learning process and reducing the complexity.

One example of integrating the DRL with a supervised learning algorithm is to improve channel decoding in wireless communication systems. Channel decoding aims to recover the original transmitted information from the received signals corrupted by noise and interference. Traditionally, supervised learning algorithms such as convolutional NNs have been employed for channel decoding. However, these algorithms may struggle with complex channel conditions or limited training data. By integrating the DRL with supervised learning, such limitations can be addressed. The DRL can learn to adaptively adjust the decoding strategy based on the current channel conditions and optimize the decoding process. The supervised learning component can provide a controlled environment and help enhance the DRL learning process. For instance, the supervised learning algorithm can be used to pre-train the DRL model using a large dataset of labeled channel samples. The DRL agent can then fine-tune the model using RL techniques and further adapt it to different channel conditions. This hybrid DRL approach can enhance channel decoding performance by leveraging the generalization capabilities of supervised learning and the adaptability of DRL. It enables the DRL agent to learn optimal decoding policies based on real-time observations and feedback, leading to improved decoding accuracy in diverse channel environments.

5.2.1.2 Combined DRL Algorithms

Combining two DRL algorithms can be beneficial when solving complex action space problems with both continuous and discrete actions [4]. One example is the joint optimization of power control and discrete RIS phase shifts in multi-user wireless communication systems. The DDPG and the DQL can be combined effectively to solve such problems. For the continuous parameter, the DDPG can be used. The actor network can approximate the continuous action space directly and learn a policy that maximizes the expected cumulative reward for those actions. On the other hand, the DQL algorithm can be used to optimize the discrete actions. The input of the network can include the approximated continuous actions of the DDPG, to enhance the learning process of the DQL agent. The DQL network can be trained to estimate the Q-values for the discrete actions. To reduce the complexity of the hybrid algorithm, a smooth transition factor can be introduced, where the factor determines when to update the NNs of the DDPG and the DQL algorithms. This hybrid approach leverages the strengths of each algorithm and allows for optimal decision-making in complex environments with diverse action requirements.

5.2.2 *Exploiting Multi-Agent DRL*

Multi-agent DRL refers to the application of the DRL techniques in scenarios involving multiple agents interacting in a shared environment. In traditional DRL, a single agent learns to make sequential decisions in an environment to maximize a cumulative reward signal. On the other hand, in multi-agent DRL, multiple agents learn simultaneously and interact with each other and their shared environment. Each agent has its own observations of the environment and can take actions independently. The agents can work on optimizing different objectives to achieve a final common goal. The interactions among agents and with the environment can be competitive, cooperative, or a combination of both. Multi-agent DRL can be applied to a wide range of scenarios in wireless communications, and more. It enables agents to learn effectively through different strategies, coordination mechanisms, and adaptive behaviors that can optimize the overall system performance [5].

Multi-agent DRL has emerged as a promising approach for addressing optimization challenges in RIS-assisted wireless communications. In the realm of RIS-assisted wireless communications, multi-agent DRL offers several opportunities for improvement. For example, multi-agent DRL can be applied in RIS-assisted systems for joint radio resource allocation problems. By combining the RIS phase shift optimization with resource allocation decisions, agents representing RISs, BSs, or UEs can learn to make collaborative decisions. Through the interaction among agents, multi-agent DRL can optimize transmit power, user association, and RIS phase shifts, achieving efficient and robust resource allocation. Furthermore, interference management is a crucial challenge in RIS-assisted FD wireless communication scenarios, and multi-agent DRL can play a significant role in mitigating interference. Agents, representing RISs or BSs, can learn to coordinate their actions to improve signal quality for targeted users while minimizing interference for others. By adaptively adjusting the optimization parameters, agents can respond to changing interference conditions differently and collectively optimize their operations to reduce interference for neighboring agents or users. Intelligent coordination allows for interference-aware decision-making and improved interference management.

Despite the progress made in multi-agent DRL for RIS-assisted wireless communications, several aspects still require further investigation. One of the main aspects is scalability. In particular, existing works focused on a limited number of agents and simplified scenarios, while multi-agent DRL was developed to handle more than 40 agents in game theory applications [6]. Developing scalable multi-agent DRL algorithms to handle complex RIS-assisted environments is needed. Furthermore, efficient collaboration and information exchange among the DRL agents are crucial for achieving optimal system performance. By addressing these gaps, multi-agent DRL can further advance the DRL capabilities and practical implementation of large-scale RIS-assisted wireless communication systems.

5.2.3 *Incorporating Transfer Learning into DRL*

In recent years, there has been growing interest in leveraging transfer learning techniques in combination with DRL to enable agents to adapt efficiently to complex tasks with limited knowledge of the environment. Transfer learning enables the transfer of knowledge and representations learned from a source task to improve learning and performance on a target task. This paradigm can be particularly beneficial in RIS-assisted systems, where the availability of labeled data and system-specific training is limited [7].

By incorporating transfer learning into DRL algorithms, RIS-assisted wireless systems can benefit from the knowledge gained from previously accomplished or relative tasks. The pre-trained models or representations from the source tasks can be used to initialize the DRL models for the target tasks. This initialization helps in capturing useful features and reducing the learning time required for convergence on future unseen tasks.

One approach is to pre-train an DNN using model-based DRL approaches (i.e., for optimization purposes), or a dataset from a relevant task (i.e., for channel estimation or interference modeling purposes). Then, the DRL techniques can be deployed to optimize RIS-assisted wireless systems, starting with partial knowledge of the nature of the problem. The pre-trained DNN serves as a feature extractor, capturing important patterns and representations that can be generalized to the target task. In this case, the RIS-assisted wireless systems can benefit from improved sample efficiency, faster convergence, and enhanced performance. The transfer of knowledge from related tasks or environments allows for better generalization, especially when labeled data or extensive training on the target task is limited.

However, it is important to carefully design the pre-trained models and transfer strategies to ensure effective learning in RIS-assisted systems. The choice of transfer learning techniques, network architectures, and fine-tuning procedures should be considered based on the requirements of the RIS-assisted wireless system. Further research and exploration in this area are crucial to fully realize the potential of DRL with transfer learning in advancing RIS-assisted wireless communication systems.

5.3 Concluding Remarks

With the recent advancements in wireless communications, researchers have been intensively investigating machine learning based techniques to optimize the networks. Furthermore, due to the increased complexity of wireless systems, it becomes infeasible to build an explainable and accurate mathematical model to predict the performance of large-scale systems. In the same context, optimal exhaustive simulations are not a practical option since they are resource-intensive. To this end, the DRL, in particular, has a huge potential to overcome the network optimization deficit by learning the optimal configuration with minimal assumptions and knowledge about the environment. This is of particular importance when

analytical insights are difficult to obtain for non-linear and non-convex optimization problems.

Although the AO field is extensively investigated based on well-defined mathematical models, the heterogeneity of the use cases envisaged for future wireless communication systems advocates for more efficient and intelligent algorithms. Therefore, in this book, we focused on in-depth explanations of how RL can be effectively applied to optimize RIS-assisted wireless communication systems. Chapter 1 a comprehensive overview of the RL working principle, followed by detailed explanations of powerful RL algorithms for both discrete and continuous action spaces, including the Q-learning, the DQL, and the DDPG. In Chap. 2, RIS-assisted wireless communication systems are thoroughly presented. Specifically, the chapter starts by discussing the working principle of RIS. Then, it presents several use cases of RIS in wireless communication systems, including cognitive radio networks, UAVs, and SWIPT. Chapters 3 and 4 focus on continuous and discrete sample optimization problems, respectively, in which the DRL can help to achieve optimal/near-optimal results. The presented examples deployed the DRL to non-convex optimization problems which cannot be solved analytically or require excessive mathematical relaxations. In all the problems, the DRL proved its efficiency in solving various complex objectives in different system settings, including HD, FD operating modes, single RIS, multi-RIS deployment schemes, single user, and multi-user models. Finally, Chap. 5 discusses open research challenges facing the application of the DRL to RIS-assisted networks. It further provides future research directions that need to be investigated to realize the full capabilities of the DRL.

References

1. Li Y, Hu X, Zhuang Y, Gao Z, Zhang P, El-Sheimy N (2020) Deep reinforcement learning (DRL): another perspective for unsupervised wireless localization. IEEE Internet Things J 7(7):6279–6287
2. Xu N, Huo C, Zhang X, Cao Y, Pan C (2022) Hyperparameter configuration learning for ship detection from synthetic aperture radar images. IEEE Geosci Remote Sens Lett 19:1–5
3. Zhang X, Jin S, Wang C, Zhu X, Tomizuka M (2022) Learning insertion primitives with discrete-continuous hybrid action space for robotic assembly tasks. In: 2022 International Conference on Robotics and Automation (ICRA), July 2022, pp 9881–9887
4. Matheron G, Perrin N, Sigaud O (2020) Understanding failures of deterministic actor-critic with continuous Action spaces and sparse rewards. In: Farkaš I, Masulli P, Wermter S (eds) Artificial neural networks and nachine nearning - ICANN 2020, vol 12397. Springer, Berlin. https://doi.org/10.1007/978-3-030-61616-8_25
5. He G, Cui S, Dai Y, Jiang T (2022) Learning task-oriented channel allocation for multi-agent communication. IEEE Trans Veh Technol 71(11):12016–12029
6. Park I, Moh T-S (2021) Multi-agent deep reinforcement learning for walker systems. In: 2021 20th IEEE International Conference on Machine Learning and Applications (ICMLA), December 2021, pp 490–495
7. Torrey L, Shavlik J (2010) Transfer learning. In: Handbook of research on machine learning applications and trends: algorithms, methods, and techniques. IGI Global, Hershey, pp 242–264