

Review

Phase Shift Design in RIS Empowered Wireless Networks: From Optimization to AI-Based Methods

Zongze Li ¹, Shuai Wang ², Qingfeng Lin ³, Yang Li ⁴, Miaowen Wen ⁵, Yik-Chung Wu ^{3,*} and H. Vincent Poor ⁶

¹ Peng Cheng Laboratory, Shenzhen 518038, China; lizz@pcl.ac.cn

² Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences, Shenzhen 518055, China; s.wang@siat.ac.cn

³ Department of Electrical and Electronic Engineering, The University of Hong Kong, Hong Kong; qflin@eee.hku.hk

⁴ Shenzhen Research Institute of Big Data, Shenzhen 518172, China; liyang@sribd.cn

⁵ School of Electronics and Information Engineering, South China University of Technology, Guangzhou 510640, China; eemwwen@scut.edu.cn

⁶ Department of Electrical and Computer Engineering, Princeton University, Princeton, NJ 08544, USA; poor@princeton.edu

* Correspondence: ycwu@eee.hku.hk

Abstract: Reconfigurable intelligent surfaces (RISs) offer the potential to customize the radio propagation environment for wireless networks. To fully exploit the advantages of RISs in wireless systems, the phases of the reflecting elements must be jointly designed with conventional communication resources, such as beamformers, the transmit power, and computation time. However, due to the unique constraints on the phase shifts and the massive numbers of reflecting units and users in large-scale networks, the resulting optimization problems are challenging to solve. This paper provides a review of the current optimization methods and artificial-intelligence-based methods for handling the constraints imposed by RISs and compares them in terms of the solution quality and computational complexity. Future challenges in phase-shift optimization involving RISs are also described, and potential solutions are discussed.

Keywords: artificial intelligence; numerical optimization; resource allocation; reconfigurable intelligent surfaces



Citation: Li, Z.; Wang, S.; Lin, Q.; Li, Y.; Wen, M.; Wu, Y.-C.; Poor, H.V. Phase Shift Design in RIS Empowered Wireless Networks: From Optimization to AI-Based Methods. *Network* **2022**, *2*, 398–418. <https://doi.org/10.3390/network2030025>

Academic Editor: Vasilis Friderikos

Received: 23 May 2022

Accepted: 4 July 2022

Published: 11 July 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

It is well-known that line-of-sight (LoS) propagation is a desirable but rarely occurring scenario for wireless communications. A standard technique to address this issue is to deploy more active nodes, such as base stations (BSs), access points, or relays to improve coverage and compensate for the high propagation loss in a non-LoS environment. However, this approach will incur high energy consumption and deployment/backhaul/maintenance costs. Worse still, this can also cause severe and complicated network interference issues.

Recently, reconfigurable intelligent surfaces (RISs), which are passive devices equipped with large numbers of low cost reflective elements, have emerged as a promising technology to overcome the above challenges. Compared with the conventional active nodes approach, which actively transmits the signals, an RIS shapes the incoming signal by adjusting the phase shifts of the reflecting elements, and could provide virtual LoS links between a BS and mobile users even the direct LoS path is blocked.

A simple scenario is illustrated in Figure 1, where there is one BS and one user, each with a single antenna. The equivalent channel from the user to the BS is the multiplication of the channel from the user to the RIS, the amplitude and phase shifts of the RIS reflective elements, and the channel from the RIS to the BS. Clearly, the RIS provides an unprecedented way of controlling the channel quality.

In particular, with a similar reasoning as with the traditional equal gain combining, it has been shown in this simple scenario that the maximum signal-to-noise ratio (SNR) is achieved when the RIS phase shift matches the combined phase of the user to RIS channel and the RIS to BS channel [1]. Since RIS operation is free of noise amplification and self interference [2], RISs have significant potential to enhance both spectral and energy efficiencies in urban environments [3]. Furthermore, due to the passive nature of RISs, they can be flexibly deployed in building facades, indoor walls, aerial platforms, roadside billboards, vehicle windows, etc.

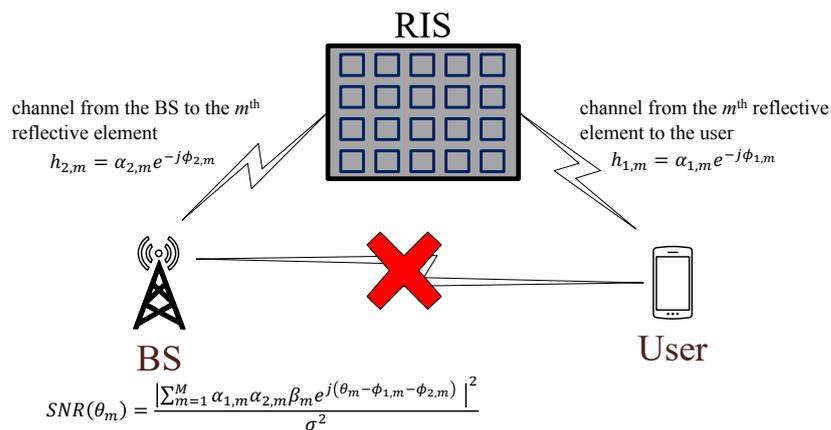


Figure 1. Illustration of RIS function in a simple scenario.

While RISs could be game-changing, the phase shift design in a practical modern communication system would be more complicated than as shown in the simple scenario of Figure 1. In a practical communication system, an RIS would serve more than one user at any particular time. Simply matching the channel of one user may hurt the communication quality of other users. In general, the phase shifts need to be optimized to balance the communication needs of all users, and more often than not, the RIS phase shifts will be optimized together with other communication resources.

To illustrate the importance of optimizing the phase shifts together with other resources, we consider a vehicle-to-everything (V2X) system in Figure 2, which consists of a BS located on the left side of the map, an RIS located at the intersection, and three intelligent vehicles marked in different colors. Each car is equipped with a front camera and LiDAR that capture data from the environment. These sensed data need to be transmitted to the BS for cooperative perception, remote driving, or vehicle platooning.

Due to significant shadowing effects, the received signal power reduces quickly with distance away from the intersection, and high data rate transmission could not be achieved. One can either take a longer duration for transmission, which is not desirable as outdated data is not useful in an intelligent traffic system, or use lossy compression to reduce the amount of data to be sent, which would unfortunately compromise the integrity of information if the compression loss is too much. We illustrate the consequences of the latter option and show how an RIS might help to mitigate them.

In particular, we use the simulation platform of Car Learning to Act (CARLA) and Pytorch in Ubuntu 18.04 with a GeForce GTX 1080GPU for graphic rendering and generation of vivid sensing data [4]. The ground-truth images of a particular frame from the front cameras are shown at the lower-left of Figure 2. We simulate three transmission schemes: (a) direct transmission without an RIS; (b) RIS-aided transmission with random phase shifts but optimized beamformer at the BS; and (c) RIS-aided transmission with optimized phase shifts and beamformer at BS. We use the total sum rate of all three users as the optimization objective, and the resulting signal-to-interference-plus-noise ratios (SINRs) and available data-rates of the three vehicles are shown on the right side of Figure 2.

Due to the aggressive compression for fitting the data into a poor channel, the images received without the help of an RIS are blurry. With an RIS, there is an observable im-

provement even just with random phase shifts. If the phase shifts of the RIS are optimized, the received images match the ground-truth well. This demonstrates the desirability of deploying an RIS and the optimization of phase shifts together with other resources in this V2X communication scenario.

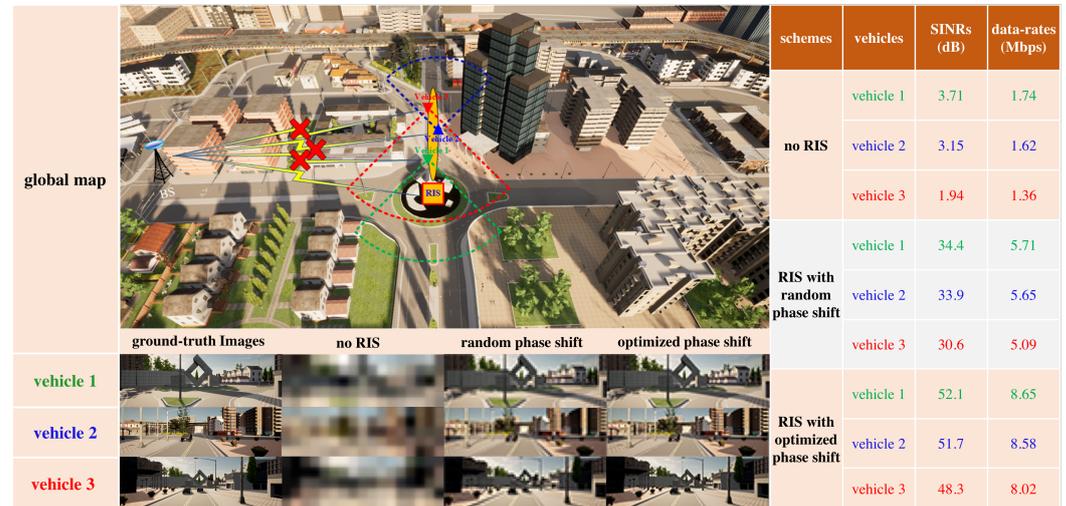


Figure 2. RIS-aided V2X for autonomous driving with camera video stream transmission. The video data is generated by the CARLA simulator [4], and the SINRs are computed using MATLAB.

Due to the promising prospects of RISs in future wireless networks, the amount of related research has exploded in the past few years. Furthermore, a number of overview or survey articles from various perspectives have also been published, and they are summarized in Table 1.

Table 1. Representative survey/overview of papers related to RIS.

Reference	Review Focuses
[2]	Differences and similarities between RIS and relay.
[1]	RIS-aided wireless communications and its future research.
[5]	RIS technology for wireless communication, and its applications.
[6]	State-of-the-art solutions for RIS-empowered wireless networks with an emphasis on applying RIS as multipath controller and energy-efficient transmitter.
[7]	Hardware designs, channel models, and channel estimation techniques for RIS-aided wireless networks.
[8]	The implementations, applications, and open research problems of large intelligent surface.
[9]	The functional and physical architecture of software-controlled metasurface and discuss its network-layer integration.
[10]	The holographic MIMO surface, its hardware architectures as well as main characteristics.
[11]	RIS applications, state-of-the-art research and future research directions.
[12]	RIS channel estimation, passive information transfer, and resource allocation.

Compared with the recent survey articles on RISs, this paper focuses on reviewing RIS phase-shift optimization from signal processing and artificial intelligence (AI) standpoints. In particular, to optimize the nonconvex-constrained phase shifts at an RIS, a number of optimization methods have been proposed in the literature, including semidefinite relaxation (SDR), the penalty method, the majorization-minimization (MM) algorithm [13], the manifold method [14], gradient descent (GD) [15], and convex relaxation (CR) [16]. AI

methods, such as unsupervised learning [17], supervised learning [18], and reinforcement learning [19], have also recently emerged as viable solutions. However, the properties of these diverse algorithms are scattered in the literature, and there is a lack of comparisons among them in the context of RISs. To fill this gap, in this paper, for the first time, we summarize these techniques, reveal their relationships, and compare their properties via simulations.

The common notation that will be used in this article is summarized in Table 2, and the rest of this paper is organized as follows. Section 2 gives the RIS resource allocation examples and general formulation. Section 3 reviews the optimization methods under continuous phase shifts. Section 4 summarizes the learning-based methods. Section 5 discusses the future challenges. Finally, Section 6 concludes the paper.

Table 2. Notation used in this article.

Notation	Description
M	number of reflecting elements in RIS
K	number of users
σ^2	variance of white Gaussian noise
\mathbf{H}	channels from BS to the RIS
$\mathbf{h}_{r,k}$	channels from RIS to user k
$\mathbf{h}_{d,k}$	channels from BS to user k
$\hat{\mathbf{h}}_k$	equivalent channels from BS to user k
\mathbf{e}	vector of RIS coefficients
β_m	amplitude for RIS's m^{th} reflecting element
θ_m	phase shift for RIS's m^{th} reflecting element
$\mathbf{Q} = \mathbf{e}\mathbf{e}^H$	rank-one auxiliary variable of \mathbf{e}

2. RIS Resource Allocation Examples and General Formulation

In wireless resource allocation involving an RIS, there are two types of resources. One comprises the conventional communication resources, such as a beamforming vector, artificial noise, transmit power, and computation time. The other consists of the RIS coefficients. Each type of resource has its own constraint, and there are possibly additional constraints coupling the two types of resources. Below are three application examples and their problem formulations. In each of the examples, it is assumed that there are M reflecting elements, and the RIS coefficients are expressed in a vector $\mathbf{e} := [e_1, \dots, e_M]^H \in \mathcal{F}$, with \mathcal{F} being the feasible set of RIS coefficients, and the specific form of \mathcal{F} will be discussed after the three examples.

- Secure beamforming for multiple-input single-output (MISO) systems [20]: As shown in Figure 3a, the BS communicates with a single-antenna user with the help of an RIS in the presence of a single-antenna eavesdropper. The goal is to maximize the achievable secrecy rate by jointly optimizing the beamformer at the BS and the phase shift coefficients of the RIS under the transmit power constraint at the BS. To be specific, let the channels from the BS to the RIS, from the RIS to user, from the RIS to eavesdropper, and the beamforming vector at the BS be, respectively, denoted by $\mathbf{H} \in \mathbb{C}^{M \times N}$, $\mathbf{h} \in \mathbb{C}^{M \times 1}$, $\mathbf{g} \in \mathbb{C}^{M \times 1}$, and $\mathbf{w} \in \mathbb{C}^{N \times 1}$. Then, the secrecy rate maximization problem is given by

$$\begin{aligned} \max_{\mathbf{w}, \mathbf{e}} \quad & \log_2 \left(\frac{\sigma^2 + |\mathbf{e}^H \text{diag}(\mathbf{h}^H) \mathbf{H} \mathbf{w}|^2}{\sigma^2 + |\mathbf{e}^H \text{diag}(\mathbf{g}^H) \mathbf{H} \mathbf{w}|^2} \right), \\ \text{s.t.} \quad & \|\mathbf{w}\|^2 \leq P_{\max}, \\ & \mathbf{e} \in \mathcal{F}, \end{aligned} \quad (1)$$

where σ^2 is the variance of white Gaussian noise at the user.

- MISO uplink communication networks [21]: There are a number of single-antenna mobile users transmitting signals to a multi-antenna BS with the assistance of an RIS as shown in Figure 3b. The objective is to minimize the total uplink transmit power by jointly optimizing the phase shift coefficients of the RIS \mathbf{e} , the transmission power x_k of the user k under the limited transmission power P_k , and the signal-to-interference-and-noise-ratio (SINR) constraints. Let the channels from the BS to the RIS, from the RIS to user k , and from the BS to user k be, respectively, denoted by $\mathbf{H} \in \mathbb{C}^{M \times N}$, $\mathbf{h}_{r,k} \in \mathbb{C}^{M \times 1}$, $\mathbf{h}_{d,k} \in \mathbb{C}^{N \times 1}$ with $k \in \{1, \dots, K\}$. Accordingly, the weighted power minimization problem is given by

$$\begin{aligned} \min_{\mathbf{x}, \mathbf{e}} \quad & \lambda^T \mathbf{x}, \\ \text{s.t.} \quad & x_k \leq P_k, \quad \forall k, \\ & \mathbf{e} \in \mathcal{F}, \\ & x_k \hat{\mathbf{h}}_k \left(\sigma^2 \mathbf{I}_N + \sum_{i \neq k} x_i \hat{\mathbf{h}}_i^H \hat{\mathbf{h}}_i \right)^{-1} \hat{\mathbf{h}}_k^H \geq r_k, \quad \forall k, \end{aligned} \quad (2)$$

where $\hat{\mathbf{h}}_k = \mathbf{h}_{r,k}^H \text{diag}(\mathbf{e}) \mathbf{H} + \mathbf{h}_{d,k}^H \in \mathbb{C}^{1 \times N}$ is the equivalent channel from user k to the BS, $\lambda = [\lambda_1, \dots, \lambda_K]^T$ represents the weights for mobile users, and r_k is the minimum SINR requested by the user k .

- Computation offloading in Internet of Things (IoT) networks [22]: In the downlink transmission of an RIS-aided cache-enabled radio access network, a multi-antenna BS transmits signals to a number of single-antenna users as shown in Figure 3c. The goal is to minimize the total network cost that consists of both the backhaul capacity and the transmission power by adjusting the caching proportion of the file requested by user k , the precoding vector $\mathbf{p}_k \in \mathbb{C}^{M \times 1}$ at the BS for user k , and the RIS coefficients. In addition, the constraint on the RIS coefficients, we also have a constraint on the size of total cached content to be smaller than the local storage size S_{\max} at the BS. Further letting the target rate of user k be denoted by R_k , the total network cost minimization problem is formulated as

$$\begin{aligned} \min_{\mathbf{x}, \mathbf{e}, \{\mathbf{p}_k\}_{k=1}^K} \quad & \sum_{k=1}^K (1 - x_k) R_k + \eta \sum_{k=1}^K \|\mathbf{p}_k\|^2, \\ \text{s.t.} \quad & \sum_{k=1}^K x_k \leq S_{\max}, \\ & x_k \in [0, 1], \quad \forall k \\ & \mathbf{e} \in \mathcal{F}, \\ & \frac{|\hat{\mathbf{h}}_k \mathbf{p}_k|^2}{\sum_{l \neq k} |\hat{\mathbf{h}}_k \mathbf{p}_l|^2 + \sigma^2} \geq 2^{R_k/B} - 1, \quad \forall k, \end{aligned} \quad (3)$$

where η is a regularization parameter, $2^{R_k/B} - 1$ is the SINR requirement in terms of the content-delivery target rate of user k , B is the bandwidth of the system, and $\hat{\mathbf{h}}_k$ is defined as in the previous example.

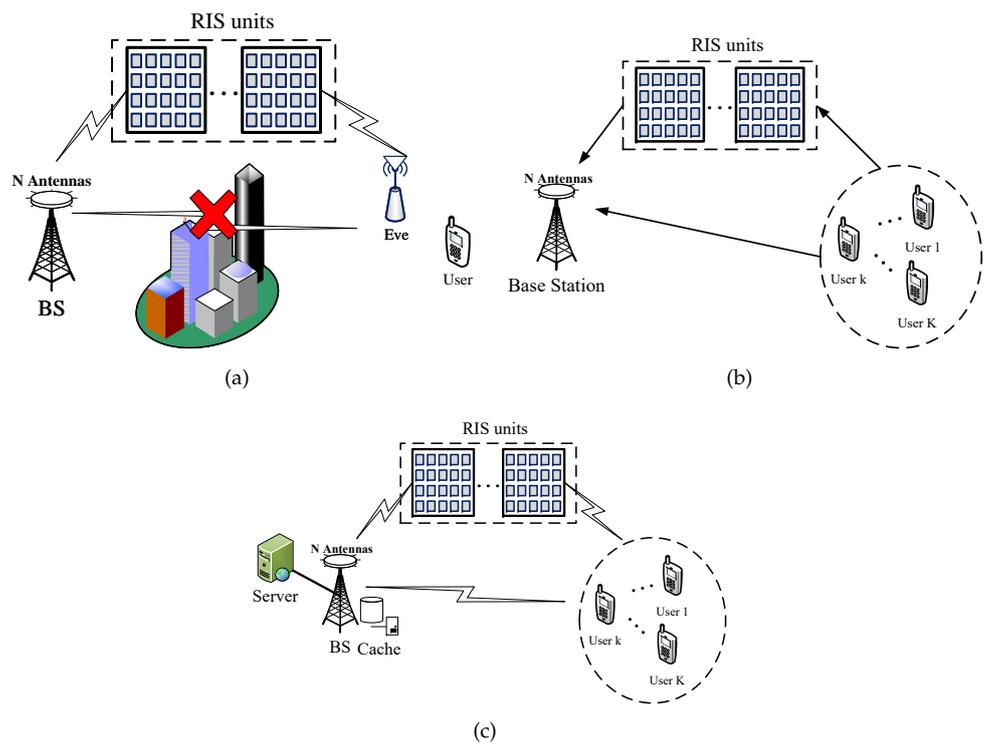


Figure 3. (a) Secure beamforming for MISO systems [20]. (b) MISO uplink communication networks [21]. (c) Computation offloading in IoT networks [22].

In the above three applications and beyond [23–31], we can see that most of the constraints in the resource allocation problems are decoupled in the sense that constraints for the RIS coefficients \mathbf{e} do not involve other resources, and vice versa. For the coupled constraints, e.g., the last constraints in problems (2) and (3), they can be converted into penalty terms in the objective function [32,33] or decoupled by introducing auxiliary variables [34–37]. After these operations, without loss of generality, we consider a general resource allocation problem appearing in the form

$$\min_{\mathbf{x}, \mathbf{e}} f(\mathbf{x}, \mathbf{e}), \quad \text{s.t. } \mathbf{x} \in \mathcal{X}, \mathbf{e} \in \mathcal{F}, \tag{4}$$

where $f(\mathbf{x}, \mathbf{e})$ is a continuous objective function, and \mathbf{x} represents the conventional communication resources with the set \mathcal{X} representing the constraint on \mathbf{x} , such as the maximum transmit power, limited cache size, operation time limitation, etc.

With the decoupled constraints for \mathbf{x} and \mathbf{e} , the optimization problem is tractable under the commonly used block coordinate descent (BCD) framework, which alternatively solves for \mathbf{x} with \mathbf{e} fixed and solves for \mathbf{e} with \mathbf{x} fixed. In particular, when the phase shift coefficients of the RIS \mathbf{e} are given, the resource allocation problem reduces to a standard communication problem without the RIS. On the other hand, when \mathbf{x} is fixed at a certain value, say $\mathbf{x}^{(n)}$, the subproblem for optimizing \mathbf{e} is

$$\min_{\mathbf{e}} f(\mathbf{x}^{(n)}, \mathbf{e}), \quad \text{s.t. } \mathbf{e} \in \mathcal{F}. \tag{5}$$

Before discussing various methods for solving (5), let us review the modeling of the constraint set \mathcal{F} on the RIS coefficients. Depending on whether the phase is modeled as a continuous or discrete variable, the feasible set \mathcal{F} is defined differently:

- **Continuous phase shift:** Each RIS coefficient has infinite phase resolution, i.e., e_m is expressed as $\beta_m e^{i\theta_m}$ with i being the imaginary unit, and θ_m as a real number. For β_m , there are three variations in the literature.
 - C1. β_m is a known constant, which is the ideal phase shift model [23,38,39]. This is the most popular model at the time of writing, and \mathcal{F} is represented by modulus constraints $|e_m|^2 = 1$;
 - C2. β_m is an unknown variable and is independent of θ_m [40,41]. This model leads to a convex set \mathcal{F} , described by $|e_m|^2 \leq c$ for some constant c ;
 - C3. β_m is a function of θ_m . This is a relatively new model and takes the hardware properties into consideration. For example, one of the recent models [42] states that

$$\beta_m(\theta_m) = (1 - \beta_{\min}) \left(\frac{\sin(\theta_m - \phi) + 1}{2} \right)^\alpha + \beta_{\min},$$

where β_{\min} , ϕ and α are known constants related to the specific circuit implementation.

- **Discrete phase shift:** Each RIS coefficient e_m can only take one of the L possible phase shift values.

Among the three continuous phase shift models, C2 is a convex set, and thus its treatment is similar to conventional resource allocation problems. Another way to view C2 is by treating the optimization of β_m and θ_m separately, so that C2 is equivalent to $0 \leq \beta_m \leq \sqrt{c}$ and $|e^{i\theta_m}|^2 = 1$. If we regard the optimization of β_m as part of conventional resources, the remaining constraint $|e^{i\theta_m}|^2 = 1$ reduces to model C1. For C3, although it is non-convex, it can be handled by gradient descent on θ_m (to be detailed in the next section). For C1, even though β_m is known and fixed, due to the modulus requirement, its handling is non-trivial, and there are a number of methods with different solution qualities for tackling this constraint.

On the other hand, for the discrete phase shift case, the corresponding problem (5) is an integer nonlinear program and is NP-hard (i.e., the optimal solution cannot be found in polynomial time). However, the most prevalent way for handling this model is to relax the discrete variables to their continuous counterparts. Then, each of the obtained continuous phase shifts (by any methods for solving continuous phase shift model) is quantized to its nearest discrete value. Since the resolution of discrete phase shifts increases with the number of allowable phases, the quantization loss will be insignificant when the number of allowable phases is large [43].

Since C1 is the most fundamental model, in this paper, we focus on reviewing the optimization methods for model C1, with some of the reviewed methods also applicable to C2 and C3. The AI-based methods will be covered in Section 4 with reinforcement learning also suitable for the discrete phase shift model. Further emerging approaches for handling the discrete phase shift case will be discussed in the section of future challenges.

3. Review on Optimization Methods under Continuous Phase Shift

Currently, the major techniques for optimizing the continuous phase shifts are the SDR method, penalty method, MM method, GD method, manifold method, and CR method. All the reviewed methods are primarily developed for C1 and can be applied to C2 if β_m and θ_m are optimized separately. For C3, this is handled by the GD method due to the complicated dependence of β_m on θ_m . Table 3 provides a quick summary of the reviewed methods in this section.

Table 3. Comparison of optimization methods for continuous phase shift RIS.

Optimization Methods	Property of Solutions	Complexity Order	Applicable Model	Examples
SDR	infeasible/feasible solution	$\mathcal{O}\left(\sqrt{M}(2M^4 + M^3)\right)$	C1 and C2	[23,24]
Penalty	stationary solution	$\mathcal{O}(M^3)$	C1 and C2	[25,26]
MM	locally optimal solution [13]	$\mathcal{O}(M^2)$	C1 and C2	[3,27]
GD	stationary solution [15]	$\mathcal{O}(M)$	C1, C2, and C3	[28,29]
Manifold	stationary solution [14]	$\mathcal{O}(M)$	C1 and C2	[22,30]
CR	feasible solution [15]	$\mathcal{O}(M^3)$ using CVX $\mathcal{O}(M)$ using PG	C1 and C2	[16,31]

3.1. SDR Method

To handle the nonconvex modulus constraints, we can introduce a rank-one auxiliary variable $\mathbf{Q} = \mathbf{e}\mathbf{e}^H$. This translates the optimization variable from \mathbf{e} to \mathbf{Q} , and the objective function changes from $f(\mathbf{x}^{(n)}, \mathbf{e})$ to $f(\mathbf{x}^{(n)}, \mathbf{Q})$. To account for the rank-one property of \mathbf{Q} and the fact that the diagonal elements of \mathbf{Q} are all 1, we need to add constraints $\text{rank}(\mathbf{Q}) = 1$ and $\mathbf{Q}_{m,m} = 1, \forall m$. Then, problem (5) under C1 is equivalent to

$$\begin{aligned} \min_{\mathbf{Q} \succeq \mathbf{0}} \quad & f(\mathbf{x}^{(n)}, \mathbf{Q}), \\ \text{s.t.} \quad & \mathbf{Q}_{m,m} = 1, \quad \forall m, \\ & \text{rank}(\mathbf{Q}) = 1. \end{aligned} \tag{6}$$

Notice that the transformed problem is still intractable due to the rank constraint $\text{rank}(\mathbf{Q}) = 1$. However, the celebrated SDR method (i.e., removing the rank constraint) can be employed to solve this problem if the cost function $f(\mathbf{x}^{(n)}, \mathbf{Q})$ is convex in \mathbf{Q} .

More specifically, with the remaining constraints $\mathbf{Q}_{m,m} = 1$ for $m = 1, \dots, M$ being transformed into semidefinite constraints $\text{Tr}(\mathbf{E}_m \mathbf{Q}) = 1$ for $m = 1, \dots, M$, where $\{\mathbf{E}_m \in \mathbb{C}^{M \times M}\}_{m=1}^M$ is a matrix with a single 1 in the $(m, m)^{th}$ position and zero in all other positions, the variable \mathbf{Q} can be directly updated via the interior point method, which is available in the software package CVX. If f is not convex, we may add another layer of successive convex approximation (SCA) to convexify the objective function in each SCA iteration with the complexity increased by a factor equal to the number of iterations for SCA. However, since the rank-1 constraint is relaxed, the obtained solution may not be a feasible solution to the original problem (6).

In general, a feasibility check is used to verify whether the obtained \mathbf{Q} satisfies the rank constraint. Since the relaxed problem is a convex problem, a closed-form solution for \mathbf{Q} or explicit expression with respect to \mathbf{Q} can be derived in its dual domain. Then, the feasibility check can be done by leveraging the ranks of product inequalities technique [44]. If the rank constraint is not satisfied, a Gaussian randomization procedure can be employed to extract a feasible solution [45]. Since the computational complexity order of SDR is $\mathcal{O}\left(\sqrt{M}(2M^4 + M^3)\right)$, it could be too time-consuming for large-scale RISs.

3.2. Penalty Method

To guarantee a feasible solution while avoiding the feasibility check of the SDR method, a penalty method can be employed. To be specific, the rank constraint $\text{rank}(\mathbf{Q}) = 1$ in (6)

can be equivalently expressed as $\text{Tr}(\sqrt{\mathbf{Q}^*\mathbf{Q}}) - \|\mathbf{Q}\|_2 \leq 0$ [46], where \mathbf{Q}^* is the conjugate of \mathbf{Q} . Then, with the constraint added as a penalized term, this further transforms problem (6) into

$$\begin{aligned} \min_{\mathbf{Q} \succeq 0} \quad & f(\mathbf{x}^{(n)}, \mathbf{Q}) + \frac{1}{\mu} \left(\text{Tr}(\sqrt{\mathbf{Q}^*\mathbf{Q}}) - \|\mathbf{Q}\|_2 \right), \\ \text{s.t.} \quad & \mathbf{Q}_{m,m} = 1, \quad \forall m, \end{aligned} \tag{7}$$

where $\mu \in (0, 1)$ is a penalty factor penalizing the violation of constraint $\text{Tr}(\sqrt{\mathbf{Q}^*\mathbf{Q}}) - \|\mathbf{Q}\|_2 \leq 0$. This transformed objective function now contains a difference-of-convex (DC) term $\text{Tr}(\sqrt{\mathbf{Q}^*\mathbf{Q}}) - \|\mathbf{Q}\|_2$. To convert the DC term to a convex form, SCA can be applied to $-\|\mathbf{Q}\|_2$ (if f is non-convex, the SCA can also be applied to f at the same time).

The resulting problem is convex in \mathbf{Q} if $f(\mathbf{x}^{(n)}, \mathbf{Q})$ is convex. Accordingly, the optimal \mathbf{Q} in each SCA iteration can be obtained by employing the interior-point method. Since the transformed problem is solved under the SCA framework, a stationary solution of \mathbf{Q} can be guaranteed. Furthermore, since problem (5) is equivalent to the transformed problem as μ tends to zero, the obtained solution is also a stationary point to (5). The penalty factor μ is important in controlling how strict the rank constraint is imposed. In practice, it can be a decreasing sequence with respect to the SCA iteration to guarantee a feasible solution of (5) at the end of the iteration. As the interior-point method is adopted in each SCA iteration, the complexity order is at least $\mathcal{O}(M^3)$.

3.3. MM Method

Both the SDR method and the penalty method require a complexity of at least $\mathcal{O}(M^3)$. To reduce the computational complexity, the MM method can be employed to tackle the unit-modulus constraint. The key idea lies in constructing a sequence of surrogate functions that serve as upper bounds of the cost function with respect to the unknown variable \mathbf{e} . Figure 4a visualizes how a linear surrogate function $g(\mathbf{x}^{(n)}, \mathbf{e}|\mathbf{e}^{(r)})$ upper bounds a convex quadratic function $f(\mathbf{x}^{(n)}, \mathbf{e})$ on the unit circle at the r^{th} iteration.

Specifically, given the solution for \mathbf{e} at the r^{th} iteration as $\mathbf{e}^{(r)}$ (the red point in Figure 4), the constructed linear surrogate function needs to satisfy: (a) $g(\mathbf{x}^{(n)}, \mathbf{e}|\mathbf{e}^{(r)}) \geq f(\mathbf{x}^{(n)}, \mathbf{e})$ on the unit circle manifold; (b) $g(\mathbf{x}^{(n)}, \mathbf{e}|\mathbf{e}^{(r)}) = f(\mathbf{x}^{(n)}, \mathbf{e})$ at $\mathbf{e}^{(r)}$; and (c) $\nabla_{\mathbf{e}} f(\mathbf{x}^{(n)}, \mathbf{e}) = \nabla_{\mathbf{e}} g(\mathbf{x}^{(n)}, \mathbf{e}|\mathbf{e}^{(r)})$ at point $\mathbf{e}^{(r)}$. In practice, the second-order Taylor expansion and Jensen’s inequality are commonly used to find $g(\mathbf{x}^{(n)}, \mathbf{e}|\mathbf{e}^{(r)})$ [13].

With the established upper bound $g(\mathbf{x}^{(n)}, \mathbf{e}|\mathbf{e}^{(r)})$, problem (5) under C1 can be iteratively solved with the subproblem at the $(r + 1)^{\text{th}}$ iteration being

$$\min_{\mathbf{e}} \quad g(\mathbf{x}^{(n)}, \mathbf{e}|\mathbf{e}^{(r)}), \quad \text{s.t.} \quad |e_m|^2 = 1, \quad \forall m. \tag{8}$$

Since $g(\mathbf{x}^{(n)}, \mathbf{e}|\mathbf{e}^{(r)})$ is a linear surrogate function, it has a closed-form minimizer $\mathbf{q}_{\mathbf{e}^{(r)}}$. Then, we can project $\mathbf{q}_{\mathbf{e}^{(r)}}$ onto the unit circle manifold to obtain $\mathbf{e}^{(r+1)}$. The next iteration involves finding $\mathbf{q}_{\mathbf{e}^{(r+1)}}$ based on $\mathbf{e}^{(r+1)}$, and the process repeats. Therefore, problem (5) can be iteratively solved, and the final converged point is a local optimal point of problem (5) [13]. The computational complexity of the MM method is dominated by the determination of surrogate functions, which gives a complexity order of $\mathcal{O}(M^2)$.

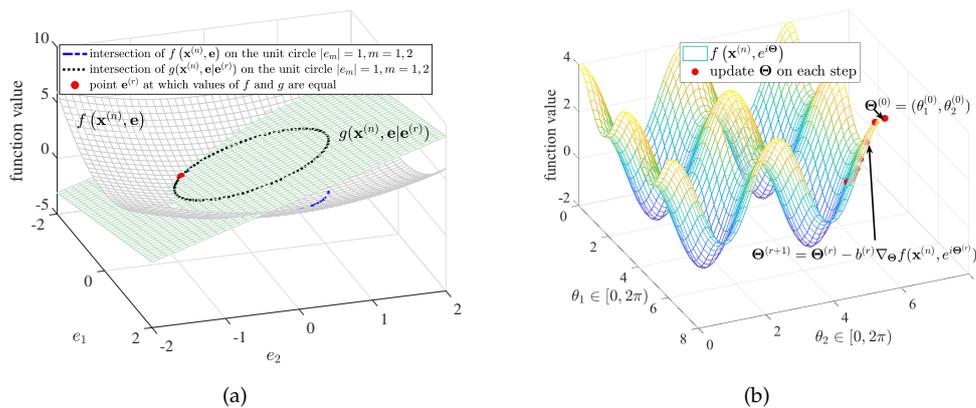


Figure 4. (a) Linear upperbound $g(\mathbf{x}^{(n)}, \mathbf{e}|\mathbf{e}^{(r)})$ for a quadratic function $f(\mathbf{x}^{(n)}, \mathbf{e})$ at a point $\mathbf{e}^{(r)}$ on the unit circle. (b) Graphical representation of the GD method for updating Θ , where $b^{(r)}$ is the update step size and $\nabla_{\Theta} f(\mathbf{x}^{(n)}, e^{i\Theta^{(r)}})$ is the gradient of f at the last iteration solution $\Theta^{(r)}$. Similar figures can be found in optimization literature, e.g., [13,46].

3.4. GD Method

Even with the MM method, the complexity order is quadratic. To further reduce the computational complexity to linear order, GD can be employed to find a stationary point of (5). The key observation is that the ultimate unknown variable in the feasible set \mathcal{F} is in fact $\{\theta_m\}_{m=1}^M$ instead of \mathbf{e} . Therefore, problem (5) can be recast into an unconstrained optimization problem as

$$\min_{\Theta} f(\mathbf{x}^{(n)}, e^{i\Theta}), \text{ s.t. } \Theta = [\theta_1, \dots, \theta_M]^T. \tag{9}$$

By recasting the quadratic function $f(\mathbf{x}^{(n)}, \mathbf{e})$ shown in Figure 4a as $f(\mathbf{x}^{(n)}, e^{i\Theta})$, a graphical demonstration of the GD method is illustrated in Figure 4b. Using a feasible initialization point $\Theta^{(0)}$, $\Theta^{(r+1)}$ can be obtained at the $(r + 1)^{th}$ iteration based on $\Theta^{(r+1)} = \Theta^{(r)} - b^{(r)} \nabla_{\Theta} f(\mathbf{x}^{(n)}, e^{i\Theta^{(r)}})$, where $b^{(r)}$ is the step size. Since only gradient information is involved in each update, GD has a linear complexity order with respect to M , and the final converged point is a stationary solution to (5).

Another point to note is that, by expressing the objective function in terms of Θ , many local minima are introduced compared to the objective function in terms of \mathbf{e} . Therefore, the quality of the converged solution of the GD method highly depends on the initialization. Notice that, since this method directly optimizes with respect to θ_m , it is also applicable to model C3 where β_m is a function of θ_m . The only change in (9) is replacing $e^{i\Theta}$ with $[\beta_1(\theta_1)e^{i\theta_1}, \dots, \beta_M(\theta_M)e^{i\theta_M}]^T$.

3.5. Manifold Method

Recognizing that the constraint set \mathcal{F} forms a complex circle manifold in model C1, another low-complexity method is based on manifold optimization. A representative algorithm in this category is the Riemannian conjugate gradient (CG) method [14], which solves problem (5) on an oblique manifold through alternatively computing the Riemannian gradient, finding the conjugate direction, and performing retraction mapping. A graphical representation of various steps of the Riemannian CG method is illustrated in Figure 5.

More specifically, the Riemannian gradient of $f(\mathbf{x}^{(n)}, \mathbf{e})$ at the l^{th} iteration solution $\mathbf{e}^{(l)}$ is obtained by projecting the Euclidean gradient of f at $\mathbf{e}^{(l)}$ onto the tangent space (blue color step in Figure 5). After obtaining the Riemannian gradient $\text{grad}_{\mathbf{e}^{(l)}} f$, the CG descent direction at point $\mathbf{e}^{(l)}$ can be obtained as $\mathbf{c}^{(l)}$, and $\mathbf{e}^{(l)}$ is updated as $\mathbf{e}^{(l)} + a^{(l)} \mathbf{c}^{(l)}$ on the tangent space, where $a^{(l)}$ is an Armijo backtracking step size (red color step in Figure 5).

Since the updated $\mathbf{e}^{(l)} + a^{(l)}\mathbf{c}^{(l)}$ may not be in the oblique manifold, the final point should be projected onto the oblique manifold by employing a retraction mapping (black color step in Figure 5). This method extends the GD method in the Euclidean space to the Riemannian manifold. Compared to the GD method in the previous subsection, the manifold method does not re-formulate the objective function in terms of Θ and thus avoids the many local minima as shown in Figure 4b. By guaranteeing that the complex circle constraint is satisfied in every iteration, the Riemannian CG method converges to a stationary solution [14]. The computational complexity of the Riemannian CG update is dominated by the gradient step, which only involves element-wise operations. This gives a linear complexity order with respect to M .

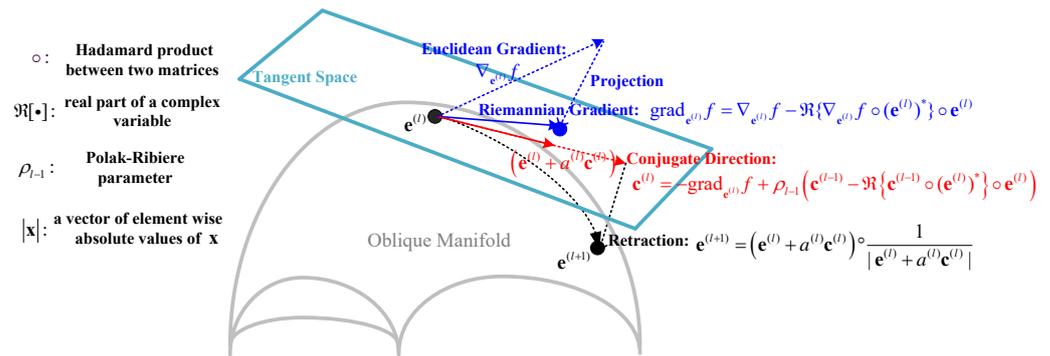


Figure 5. Graphical illustration of the Riemannian CG method at the l^{th} iteration. A similar figure can be found in the optimization literature, e.g., [47].

3.6. CR Method

The idea of the CR method is that, while the constraint set \mathcal{F} in C1 is nonconvex, it can be relaxed to a Euclidean unit ball, which is a convex set. Therefore, problem (5) under C1 can be relaxed into

$$\min_{\mathbf{e}} f(\mathbf{x}^{(n)}, \mathbf{e}), \text{ s.t. } |e_m|^2 \leq 1, \quad \forall m. \tag{10}$$

Since (10) has a convex set, it can be solved via convex tools, such as CVX. Afterward, the solution of the relaxed problem is projected to the nearest point in $|e_m|^2 = 1$ to obtain a feasible solution.

A variant of the above method is replacing the interior point method with the projected-gradient (PG) method, which alternates between gradient steps and projection steps. Although this variant has not been employed in the existing literature involving RISs, it has a linear computational complexity compared to the cubic complexity of the interior point method, and thus is promising for large-scale systems.

Notice that this method is applicable to model C2. For model C2, where \mathcal{F} is already in the form of $|e_m|^2 \leq c$, there is no relaxation involved and the solution is directly obtained from solving (10). Furthermore, unlike other methods applying to C2, there is no need to optimize β_m and θ_m separately since the optimization of β_m is incorporated in (10).

3.7. Summary and Performance Comparison

To summarize, the optimization methods for handling continuous phase shift design in this section can be categorized into relaxation methods (SDR and CR), iterative approximation methods (the penalty-based method and MM), and gradient methods (GD and the manifold method). Their relationships are summarized in Figure 6, and their properties are compared in Table 3.

To compare the performance of different optimization methods, the three application examples mentioned in Section 2 are simulated under phase shift model C1. All simulations are performed on MATLAB R2017a on a Windows X64 desktop with 3.2 GHz CPU and 16 GB RAM. For fair comparisons, all algorithms start from the same initial point (any

feasible point can serve as an initial point), and the stopping criterion for iterative methods is when the relative change of two consecutive objective function values becomes less than 10^{-4} , and the maximum number of iterations for all methods is set to 100. By employing the BCD framework for solving for \mathbf{x} and \mathbf{e} in (4), the three applications can be efficiently solved, and the simulation results are shown in Figure 7a–c, respectively.

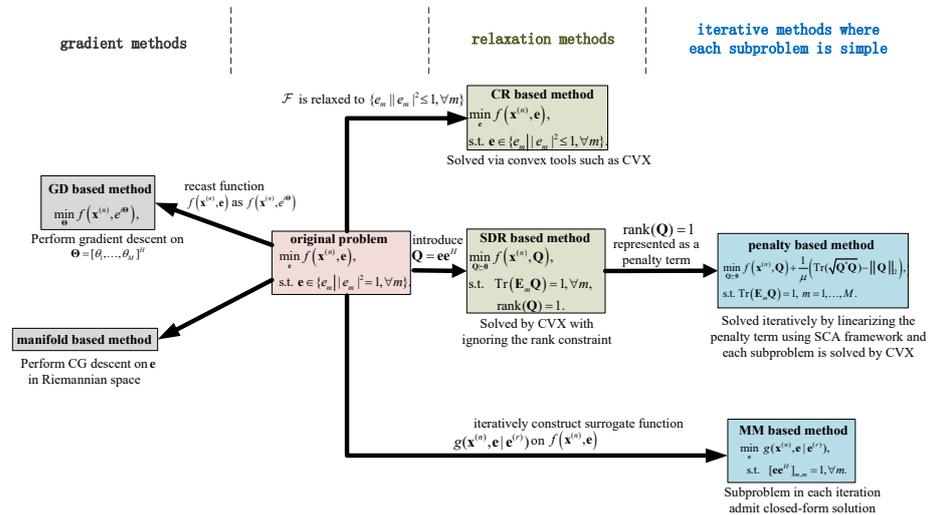


Figure 6. Relationships among different optimization methods.

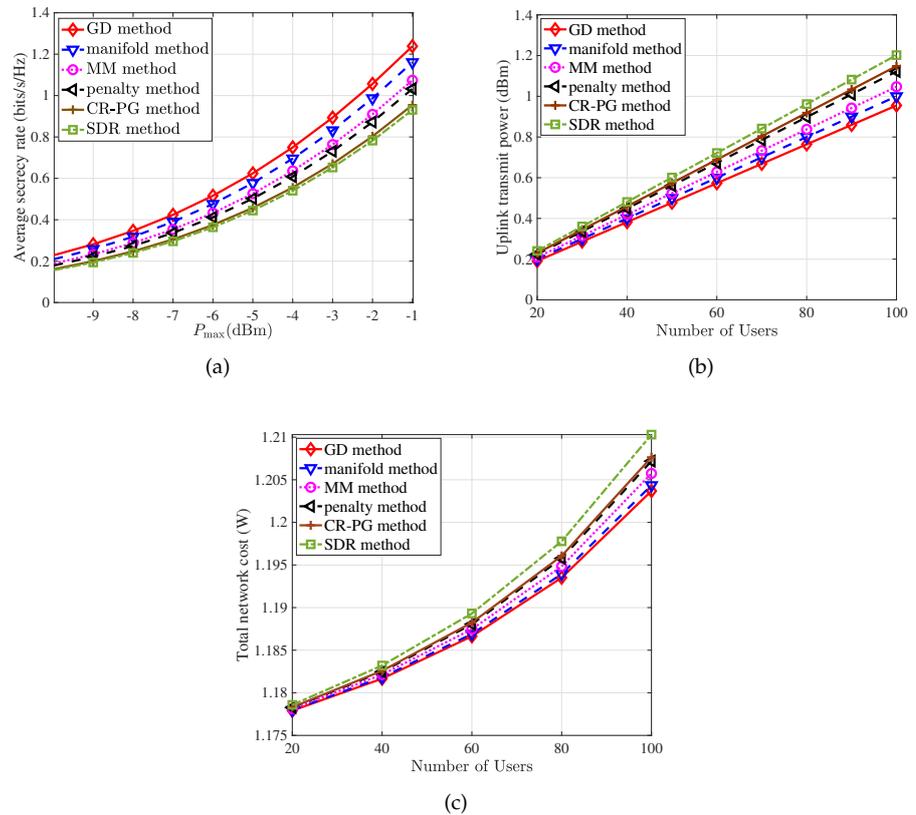


Figure 7. Performance comparisons of six optimization methods with $M = 10$: (a) Secrecy rate versus the maximum transmit power under the number of BS antennas $N = 20$ [20]. (b) Uplink transmit power versus the number of users under the number of BS antennas $N = 20$ and transmission power limitation $P_k = 10.8$ dBm [21]. (c) Total network cost versus the number of users under the number of BS antennas $N = 10$, the target rate $R_k = 10$ MHz, the bandwidth $B = 10$ MHz, the regularization parameter $\eta = 100$ and the local storage size $S_{\max} = 100$ [22].

From these figures, it can be observed that, out of the six algorithms, GD and the manifold method perform consistently well in all three applications, followed by the MM method and the penalty method. On the other hand, the SDR method and CR perform the worst in these three applications. The worse performance of the SDR method and CR is due to the relatively weak guarantee in the solution quality.

On the other hand, the computation times of various methods in the first application are shown in Figure 8a. From this figure, it can be seen that the manifold method, the GD method, and the CR-PG method require the least amounts of computation time among the six algorithms, achieving at least two orders of magnitude reduction compared with the SDR method and the penalty method when $N > 50$. This advantage becomes more prominent as the number of reflecting elements M increases as shown in Figure 8b. The computation times for the other two applications show similar behaviors and thus are not shown here.

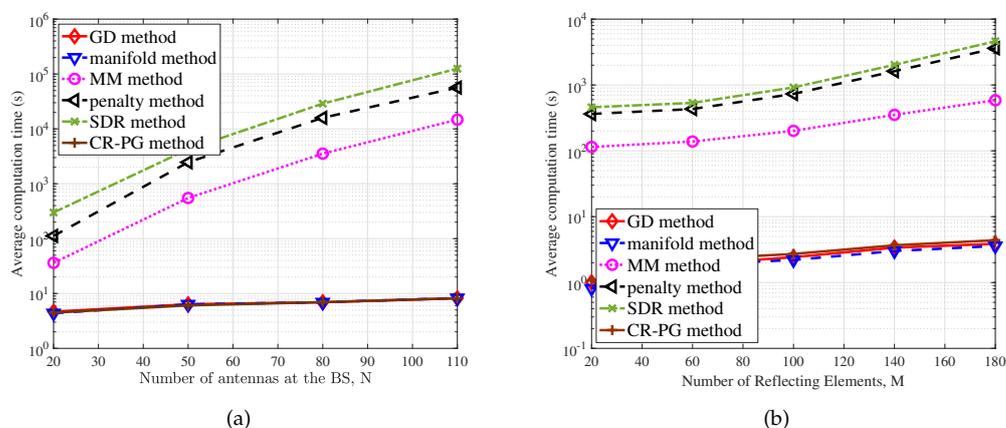


Figure 8. Performance comparisons of six optimization methods: (a) Average computation time versus the number of antennas at the BS when $M = 10$. (b) Average computation time versus the number of reflecting elements when $N = 10$.

4. Learning to Optimize An RIS

In addition to mathematical optimization methods, AI-based methods have recently emerged as a promising direction for solving resource allocation problems. Problem (5) can be regarded as a regression problem (or classification problem for discrete phase shifts), which can be tackled by deep-learning (DL) methods. When DL is employed, a deep neural network (DNN) is adopted to learn the mapping from the channel state information (CSI) to the optimized phase shift coefficients. Once the AI model is trained, the computation of phase shift coefficients is extremely fast, and it can be readily implemented in various operating systems, such as Linux and Android, via model loading. In the following, three learning-based methods are discussed.

4.1. Supervised Learning

In this paradigm, the optimal phase shift \mathbf{e} under a specific channel realization and network setting is obtained by traditional optimization approaches (as detailed in Section 3). This channel realization and the corresponding optimized phase shift are treated as a training sample. If we have many training samples corresponding to different channel realizations, a DNN can be trained to approximate the behavior of a traditional optimization method. The advantage of this approach is that the learning results inherit the solution quality from optimization methods [18,38]. However, it has an additional burden of generating training samples; however, low-complexity methods, such as GD, the manifold method, and CR-PG, help to reduce this burden compared to the SDR and MM methods. Furthermore, supervised learning can be extended to directly solve problem (4) by treating

the channel realization as input and all resources (both \mathbf{x} and \mathbf{e}) as the desired output of the DNN.

4.2. Unsupervised Learning

The connection between unsupervised learning and problem (5) comes from the observation that (5) can be regarded as an unconstrained optimization problem if the variable is viewed in terms of θ_m instead of \mathbf{e} . This view has been adopted in the GD method in Section 3. However, in contrast to the GD method for solving (9) with respect to Θ , unsupervised learning uses a DNN that accepts a channel realization as input and generates the corresponding Θ as output, where the optimization is with respect to the coefficients of the DNN. In unsupervised learning, the objective is to minimize $\mathbb{E}[f(\mathbf{x}, e^{i\Theta})]$, where the expectation is with respect to the distribution of input channel state information.

The training procedure involves first generating a large number of channel realizations and then optimizing Θ and \mathbf{x} under the BCD framework. When optimizing Θ , back propagation is used. On the other hand, when optimizing \mathbf{x} , a conventional optimization technique is used with the expectation tackled via sampling approximation. Different from supervised learning, this approach does not require the labeling of data, which saves a significant amount of time in training data preparation. However, a disadvantage is that the obtained solution does not have any quality guarantee.

4.3. Reinforcement Learning

Another major framework in AI is deep reinforcement learning (DRL). In this framework, the agent (i.e., decision maker) gradually derives its best action through trial-and-error interactions with the environment over time. There are a few basic elements characterizing the DRL process: the state, the action, the reward, and the state action value function.

1. **State:** a set, denoted by S , characterizing the environment. The state $s^{(t)} \in S$ denotes the environment at the time step t .
2. **Action:** a set of allowable actions, denoted by A . Once the agent takes an action $a^{(t)} \in A$ at time instant t (determined by the state action value function), the state of the environment will transit from the current state $s^{(t)}$ to the next state $s^{(t+1)}$.
3. **Reward:** the performance metric of a particular action, denoted by $r^{(t)}$ at time instant t .
4. **State action value function (Q – function):** while the reward represents the immediate return from action a at state s , the state action value function indicates cumulative rewards the agent may get from taking action a in the state s , which is denoted by $Q(s, a)$.

Depending on the types of action spaces, two DRL methods are available: the deep Q-network (DQN) algorithm, which is designed for discrete action spaces, and the deep deterministic policy gradient (DDPG), which is designed for continuous action spaces. Hence, DQN fits the discrete phase shift model, while the DDPG is employed for continuous phase shift variables.

In this subsection, we present a mapping of DQN in the context of resource allocation problems in RIS-empowered wireless networks. In this model, the central controller, which controls the RIS, acts as the agent. At each time slot t , the agent observes a state, $s^{(t)} \in S$, which consists of all channel state information from the wireless system. According to the current state and the Q-function, the agent takes an action, $a^{(t)} = \arg\max_a Q(s^{(t)}, a) \in A$, where A consists of discrete phase shifts that each reflecting element is allowed to take. After performing an action $a^{(t)}$, the agent obtains a reward $r^{(t)}$ determined from the negative objective function of (5) and observes the next state $s^{(t+1)}$ generated by the wireless system. At each time slot, $Q(s^{(t)}, a^{(t)})$ is updated by

$$Q(s^{(t)}, a^{(t)}) = Q(s^{(t)}, a^{(t)}) + \alpha (r^{(t)} + \gamma \arg\max_a Q(s^{(t+1)}, a) - Q(s^{(t)}, a^{(t)})), \quad (11)$$

where α is the learning rate and γ is the discount factor designed for DQN. The aim of the DQN model is to enable the agent to perform actions to maximize the long-term sum reward.

4.4. Summary and Performance Comparison

Different learning-based methods for solving problem (5) are summarized in Figure 9. For supervised learning, since the training samples are generated from conventional optimization methods, the quality of the output is determined by the properties of the solution from the employed optimization method. For the other two methods (unsupervised learning and reinforcement learning), the outputs have no such quality guarantee. To compare the performance of different learning-based methods, the first example mentioned in Section 2 is simulated, with GD optimization selected for generating training samples in supervised learning and also to serve as a performance benchmark.

Figure 10a shows the case of continuous phase shift. It is clear that supervised learning performs close to the GD algorithm. This is not surprising as supervised learning is mimicking the behavior of the optimization method chosen for generating the training data. However, for unsupervised learning, although it does not need training data preparation, it performs unmistakably worse than the supervised learning. Table 4 further shows the training times and inference times of GD, supervised learning, and unsupervised learning. It can be observed that the inference times of deep-learning methods are indeed short compared to the GD method, although their preparation and training times are long.

Table 4. Comparison of learning methods for a phase shift RIS.

Methods	Training Data Preparation Time	Training Time	Inference Time
GD	not applicable	not applicable	21.7 ms
Supervised Learning	4.8 h	10.521 h	87.1 μ s
Unsupervised Learning	not applicable	11.347 h	66.3 μ s
Reinforcement Learning	not applicable	17.862 h	14.3 ms

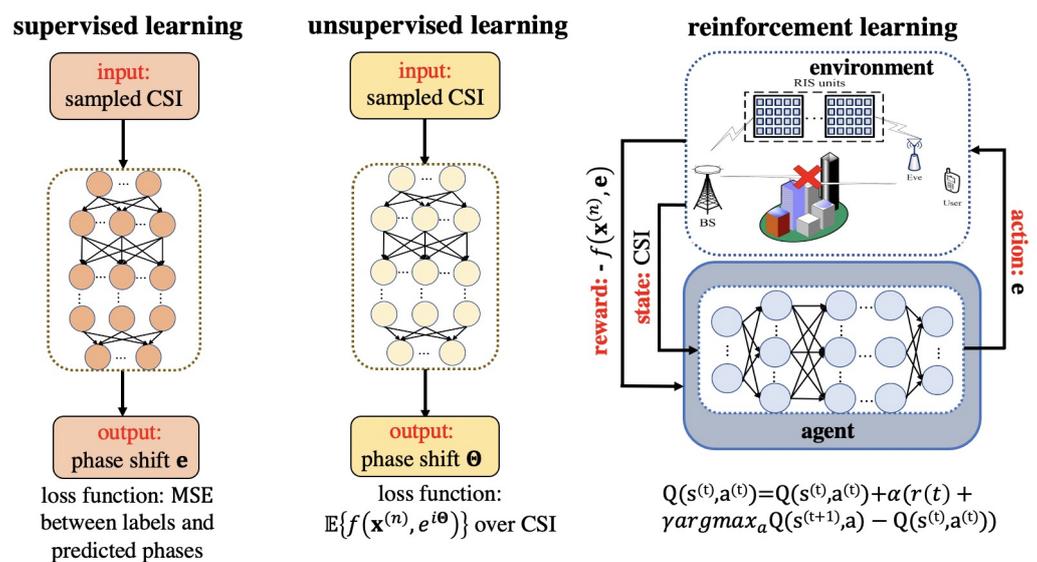


Figure 9. Illustration of different learning methods. The loss functions of supervised learning and unsupervised learning are the mean squared error (MSE) between labels and predicted phases, and the expectation of the objective function of (9) over CSI, respectively.

On the other hand, the performance of deep-learning methods under eight allowable discrete phases is shown in Figure 10b. For the supervised learning and unsupervised learning, we simply apply quantization to the learning results. For DRL, we employ the DQN algorithm, which is trained with a DNN for 2000 epochs and 128 minibatches for each epoch. The GD method with unquantized output is also included in Figure 10b to show the performance limit. It can be seen from Figure 10b that the performance of quantization under supervised and unsupervised learning does not degrade much compared to the unquantized output in Figure 10a. For DQN, its performance lies between supervised learning and unsupervised learning. The training and inference times of DQN are also shown in Table 4.

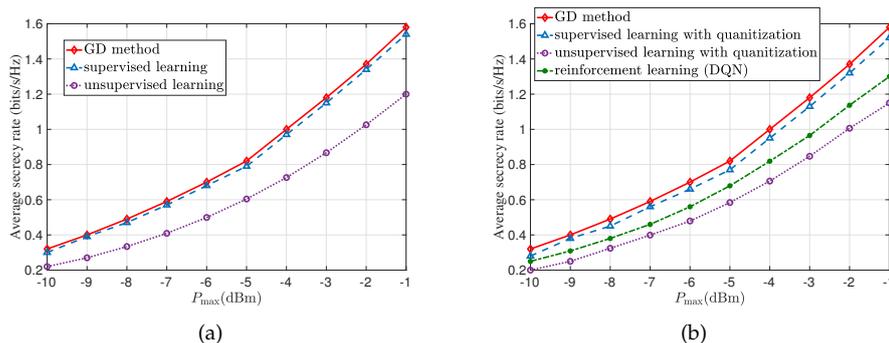


Figure 10. Performance comparison of different learning-based methods in the secrecy rate optimization problem under $N = 10$, $M = 10$. All the simulations are implemented on Colab with TensorFlow 2 and backend GPU. For training, 800,000 independent channels are generated (with the corresponding optimized phase obtained by GD methods for supervised learning). For testing, 200,000 independent channels are used. The adopted neural networks consist of three fully-connected hidden layers, containing 500, 250, and 200 neurons. The rectified linear unit (ReLU) is used as the activation function for the hidden layers and the linear activation function is applied to the output layer. (a) Continuous phase case. (b) Discrete phase case.

5. Future Challenges

While an explosive growth in the number of studies of resource allocation involving RISs has been witnessed in the past few years, there are still challenging problems remaining to be investigated. From communication and wireless network perspectives, when RISs are employed together with other emerging technologies, unique challenges will occur. These include, but are not limited to, full-duplex communications, integrated sensing and communications, Terahertz communications, new forms of multiple access, and mobile edge computing. As these challenges and open issues have been covered in existing RIS review articles [1,2,5–12], this paper focuses on the challenges in signal processing and AI. Below, four such challenges are described, and potential solutions are also discussed.

5.1. Handling Channel Uncertainty

In general, due to the large number of passive reflective elements in RISs, imperfect CSI is inevitable. Considering channel uncertainty, the resource optimization problem would be a stochastic counterpart of the problems discussed earlier. In particular, the CSI random error would make the constraints appear in a probabilistic form, and the objective function takes an extra expectation.

If the distribution of the channel uncertainty is known, this statistical information can be used to transform the probabilistic constraints into deterministic ones and compute the expectation of the objective function explicitly [48–50]. However, due to the cascaded channel created by the RIS, the statistical information of the CSI might be complicated, making the transformation from stochastic problems into deterministic ones suffer performance loss, and/or intractable expectation computation. In those cases, the Monte Carlo simulation-based method could be used to handle the channel uncertainty [51].

On the other hand, learning-based methods can be modified to tackle uncertain CSI, even when the distribution of the channel uncertainty cannot be described in closed-form. In particular, when preparing the training data, we generate both the true CSI and the CSI added with uncertainty. During the training, we input the observed CSI (which contains errors) to the DNN but compute the loss function or reward function using error-free CSI. In this way, the learning system can automatically learn to “denoise” the CSI, while learning the mapping of the RIS phase shifts.

5.2. Handling Discrete Phase Shift

Recently, the discrete phase shift model has begun to emerge under the argument that the reflecting elements only have finite reflection levels due to hardware limitations. The resulting resource allocation problem is even more challenging than its continuous phase shift counterpart since the problem involves both continuous and discrete variables. At the moment of writing, there are two major techniques for solving discrete phase shift problems: quantization or brute-force search, with the majority of works adopting quantization.

For the quantization-based method, we demonstrated in Figure 9b that the performance loss is insignificant if the number of discrete phase shifts is not small. This explains why the quantization-based method is popular among existing works. However, when the number of allowable phases is small (e.g., two or three), the quantization method will lead to inevitable performance degradation. To overcome this issue, the original integer nonlinear program can be iteratively transformed into integer linear programs via linear cuts.

Then, the branch-and-bound algorithm and exhaustive search can be employed to handle the resultant problem with discrete variables [52]. However, these searching methods have an exponential time complexity, which could lead to unacceptable complexity even for modest values of M . Recently, the idea of alternating optimization (AO) has been applied to discrete phase shift search [43], in which multiple phase shifts are optimized one at a time so that the search space in each iteration is small. While this reduces the complexity significantly, only stationary points can be guaranteed.

As can be seen, solving the discrete phase shift design problem is still in an early stage. It remains a challenge to derive a low complexity approach while achieving performance close to that of brute-force search. For the conventional optimization method, the greedy algorithm, despite its heuristic nature, might be suitable here as it has a quadratic complexity order by using a linear search at each step. In addition, by viewing the desired phase angle as a non-zero element in a sparse vector [53], sparse signal processing, such as Lasso approximation [54] and the penalty method [55], can also be applied to handle discrete random variables. On the other hand, although the DQN algorithm of DRL matches the discrete phase problem, it can only provide a feasible solution and has a slow learning rate and unstable learning process. Making DRL more efficient in wireless applications is an important direction.

5.3. Handling the Mobility of RISs and Users

For a large-scale data-centric network, since communication service requirements are highly dynamic and imbalanced among users, it is usually inefficient to deploy RISs at fixed locations. To improve network coverage and serve remote nodes, RISs can be deployed on autonomous systems, such as unmanned aerial vehicles or unmanned ground vehicles for providing flexible channel reconfigurations.

Furthermore, the locations of users may also dramatically change over time in emerging V2X networks. Due to the passive nature of RISs, they cannot send pilot signals to track the movement of the users, especially when the direct links from the BS to users are blocked [56]. With a mobile RIS or users, the system performance not only depends on the RIS's or users' locations but also on the trajectory itself. Consequently, the dimension of design variables is significantly increased. Mathematically, the time-varying phase shift design of a mobile wireless system can be modeled as a high-dimensional dynamic pro-

gramming problem, in which Q-learning, temporal difference learning, and policy iteration algorithms in approximate dynamic programming could provide effective solutions [57].

On the other hand, since the CSI for unvisited places and future time slots are unknown, the prior distribution of channels has to be predetermined via the geometry-based tracing approach. However, as time evolves, the knowledge about the channel distribution should be updated for a better phase shift design. This can be modeled as a partially observable Markov decision process, where the DRL methods can be used to learn the underlying wireless environment while choosing the moving trajectory on the fly. Hence, the state of DRL includes not only the current CSI but also the action from the previous time step. Furthermore, by exploiting the extra partial information (e.g., previous locations and velocities of users or the RIS), the post-decision state algorithm can be used to find an optimized solution in dynamic environments during the training of the DRL model [58].

5.4. Scalability of AI-Based Methods

In AI-based methods, while generic multi-layer perceptrons (MLPs) and convolutional neural networks (CNNs) have been widely used for wireless resource allocation, there are two well-known technical challenges. First, MLPs and CNNs are more difficult to train in large-scale settings than in small-scale counterparts. For example, as demonstrated in the beamforming problem [59], although the performance of CNNs is near-optimal when trained and tested under a two-user setting, there exists an 18% performance gap to the classic algorithm when trained and tested under a 10-user setting. Secondly, MLPs are designed for a pre-defined problem size with fixed input and output dimensions. In the context of an RIS problem, this means that a well-trained MLP for a particular RIS dimension is not applicable to other settings when the numbers of reflecting elements differ.

Recent studies have shown that incorporating a permutation equivariance property into the neural network architecture can reduce the parameter space, avoid a large number of unnecessary permuted training samples, and most importantly make the neural network generalizable to different problem scales [60–63]. In particular, graph neural networks (GNNs) [60,61] and attention-based transformers [62,63] have been shown to possess the permutation equivariance property and have demonstrated superior performance, scalability, and generalization ability in wireless resource allocation problems. For instance, in the beamforming problem, a GNN trained with data generated in a setting of 50 users was shown to achieve near optimal testing performance under a much larger setting of 1000 users [60].

This result simultaneously solves the two challenges mentioned above (difficulty of training in large-scale settings and generalizability to different settings). Interestingly, permutation equivariance also exists in RIS phase shift design problems since exchanging the channels of two reflecting elements should result in a corresponding permutation of the optimized phase shift design. Therefore, it is expected that GNNs and attention-based transformers would be effective neural network architectures for the RIS design problems as well.

6. Conclusions

In this paper, we reviewed and compared current optimization methods for solving resource allocation problems associated with RISs. We note that most of the available methods are tailored to continuous phase shift constraints and that AI-based methods are emerging as serious contenders. With the principles and properties of different algorithms explained and illustrated and future challenges analyzed, we hope that this paper will facilitate the suitable choice of algorithms for design problems involving RISs.

Author Contributions: Conceptualization, Z.L., Y.-C.W. and H.V.P.; methodology, Z.L., S.W., Q.L., Y.L., and M.W.; software, Z.L., S.W., Q.L., Y.L., and M.W.; validation, Z.L., S.W., Q.L., Y.-C.W. and H.V.P.; formal analysis, Z.L., S.W., Q.L., Y.L., M.W., Y.-C.W., and H.V.P.; writing—original draft preparation, Z.L., S.W., Y.L., M.W. and Y.-C.W.; writing—review and editing, Z.L., Q.L., Y.-C.W. and H.V.P.; visualization, Z.L., S.W. and Q.L.; project administration, Y.-C.W. and H.V.P.; funding acquisition, Z.L., S.W., Y.L., M.W., Y.-C.W. and H.V.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by National Natural Science Foundation of China (No. 62101349), Shenzhen Science and Technology Program (No. RCB20200714114956153) and National Science Foundation (No. CCF-1908308).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References

1. Wu, Q.; Zhang, S.; Zheng, B.; You, C.; Zhang, R. Intelligent Reflecting Surface-Aided Wireless Communications: A Tutorial. *IEEE Trans. Commun.* **2021**, *69*, 3313–3351. [\[CrossRef\]](#)
2. Di Renzo, M.; Ntontin, K.; Song, J.; Danufane, F.H.; Qian, X.; Lazarakis, F.; De Rosny, J.; Phan-Huy, D.T.; Simeone, O.; Zhang, R.; et al. Reconfigurable Intelligent Surfaces vs. Relaying: Differences, Similarities, and Performance Comparison. *IEEE Open J. Commun. Soc.* **2020**, *1*, 798–807. [\[CrossRef\]](#)
3. Huang, C.; Zappone, A.; Alexandropoulos, G.C.; Debbah, M.; Yuen, C. Reconfigurable Intelligent Surfaces for Energy Efficiency in Wireless Communication. *IEEE Trans. Wirel. Commun.* **2019**, *18*, 4157–4170. [\[CrossRef\]](#)
4. Dosovitskiy, A.; Ros, G.; Codevilla, F.; Lopez, A.; Koltun, V. CARLA: An open urban driving simulator. In Proceedings of the first Conference on Robot Learning (CoRL 2017), Mountain View, CA, USA, 13–15 November 2017; pp. 1–16.
5. Wu, Q.; Zhang, R. Towards Smart and Reconfigurable Environment: Intelligent Reflecting Surface Aided Wireless Network. *IEEE Commun. Mag.* **2020**, *58*, 106–112. [\[CrossRef\]](#)
6. Basar, E.; Di Renzo, M.; De Rosny, J.; Debbah, M.; Alouini, M.S.; Zhang, R. Wireless Communications Through Reconfigurable Intelligent Surfaces. *IEEE Access* **2019**, *7*, 116753–116773. [\[CrossRef\]](#)
7. Jian, M.; Alexandropoulos, G.C.; Basar, E.; Huang, C.; Liu, R.; Liu, Y.; Yuen, C. Reconfigurable intelligent surfaces for wireless communications: Overview of hardware designs, channel models, and estimation techniques. *Intell. Conver. Netw.* **2022**, *3*, 1–32. [\[CrossRef\]](#)
8. Liang, Y.C.; Long, R.; Zhang, Q.; Chen, J.; Cheng, H.V.; Guo, H. Large Intelligent Surface/Antennas (LISA): Making Reflective Radios Smart. *J. Commun. Inf. Netw.* **2019**, *4*, 40–50. [\[CrossRef\]](#)
9. Liaskos, C.; Nie, S.; Tsiolaridou, A.; Pitsillides, A.; Ioannidis, S.; Akyildiz, I. A New Wireless Communication Paradigm through Software-Controlled Metasurfaces. *IEEE Commun. Mag.* **2018**, *56*, 162–169. [\[CrossRef\]](#)
10. Huang, C.; Hu, S.; Alexandropoulos, G.C.; Zappone, A.; Yuen, C.; Zhang, R.; Renzo, M.D.; Debbah, M. Holographic MIMO Surfaces for 6G Wireless Networks: Opportunities, Challenges, and Trends. *IEEE Wirel. Commun.* **2020**, *27*, 118–125. [\[CrossRef\]](#)
11. Di Renzo, M.; Zappone, A.; Debbah, M.; Alouini, M.S.; Yuen, C.; de Rosny, J.; Tretyakov, S. Smart Radio Environments Empowered by Reconfigurable Intelligent Surfaces: How It Works, State of Research, and The Road Ahead. *IEEE J. Sel. Areas Commun.* **2020**, *38*, 2450–2525. [\[CrossRef\]](#)
12. Yuan, X.; Zhang, Y.J.A.; Shi, Y.; Yan, W.; Liu, H. Reconfigurable-Intelligent-Surface Empowered Wireless Communications: Challenges and Opportunities. *IEEE Wirel. Commun.* **2021**, *28*, 136–143. [\[CrossRef\]](#)
13. Sun, Y.; Babu, P.; Palomar, D.P. Majorization-Minimization Algorithms in Signal Processing, Communications, and Machine Learning. *IEEE Trans. Signal Process.* **2017**, *65*, 794–816. [\[CrossRef\]](#)
14. Absil, P.A.; Mahony, R.; Sepulchre, R. *Optimization Algorithms on Matrix Manifolds*; Princeton University Press: Princeton, NJ, USA, 2009.
15. Ma, Y.; Shen, Y.; Yu, X.; Zhang, J.; Song, S.H.; Letaief, K.B. A Low-Complexity Algorithmic Framework for Large-Scale IRS-Assisted Wireless Systems. *arXiv* **2020**, arXiv:2008.00769.
16. Chen, J.; Liang, Y.C.; Pei, Y.; Guo, H. Intelligent Reflecting Surface: A Programmable Wireless Environment for Physical Layer Security. *IEEE Access* **2019**, *7*, 82599–82612. [\[CrossRef\]](#)
17. Gao, J.; Zhong, C.; Chen, X.; Lin, H.; Zhang, Z. Unsupervised Learning for Passive Beamforming. *IEEE Commun. Lett.* **2020**, *24*, 1052–1056. [\[CrossRef\]](#)
18. Taha, A.; Alrabeiah, M.; Alkhateeb, A. Deep Learning for Large Intelligent Surfaces in Millimeter Wave and Massive MIMO Systems. In Proceedings of the IEEE Global Communication Conference, Waikoloa, HI, USA, 9–13 December 2019; pp. 1–6.

19. Huang, C.; Mo, R.; Yuen, C. Reconfigurable Intelligent Surface Assisted Multiuser MISO Systems Exploiting Deep Reinforcement Learning. *IEEE J. Sel. Areas Commun.* **2020**, *38*, 1839–1850. [[CrossRef](#)]
20. Lu, X.; Yang, W.; Guan, X.; Wu, Q.; Cai, Y. Robust and Secure Beamforming for Intelligent Reflecting Surface Aided mmWave MISO Systems. *IEEE Wirel. Commun. Lett.* **2020**, *9*, 2068–2072. [[CrossRef](#)]
21. Liu, Y.; Zhao, J.; Li, M.; Wu, Q. Intelligent Reflecting Surface Aided MISO Uplink Communication Network: Feasibility and Power Minimization for Perfect and Imperfect CSI. *IEEE Trans. Commun.* **2020**, *69*, 1975–1989. [[CrossRef](#)]
22. Chen, Y.; Wen, M.; Basar, E.; Wu, Y.C.; Wang, L.; Liu, W. Exploiting Reconfigurable Intelligent Surfaces in Edge Caching: Joint Hybrid Beamforming and Content Placement Optimization. *IEEE Trans. Wirel. Commun.* **2021**, *20*, 7799–7812. [[CrossRef](#)]
23. Wu, Q.; Zhang, R. Intelligent Reflecting Surface Enhanced Wireless Network via Joint Active and Passive Beamforming. *IEEE Trans. Wirel. Commun.* **2019**, *18*, 5394–5409. [[CrossRef](#)]
24. You, C.; Zheng, B.; Zhang, R. Channel Estimation and Passive Beamforming for Intelligent Reflecting Surface: Discrete Phase Shift and Progressive Refinement. *IEEE J. Sel. Areas Commun.* **2020**, *38*, 2604–2620. [[CrossRef](#)]
25. Yu, X.; Xu, D.; Sun, Y.; Ng, D.W.K.; Schober, R. Robust and Secure Wireless Communications via Intelligent Reflecting Surfaces. *IEEE J. Sel. Areas Commun.* **2020**, *38*, 2637–2652. [[CrossRef](#)]
26. Zhao, M.M.; Liu, A.; Zhang, R. Outage-Constrained Robust Beamforming for Intelligent Reflecting Surface Aided Wireless Communication. *IEEE Trans. Signal Process.* **2021**, *69*, 1301–1316. [[CrossRef](#)]
27. Wang, H.M.; Bai, J.; Dong, L. Intelligent Reflecting Surfaces Assisted Secure Transmission Without Eavesdropper's CSI. *IEEE Signal Process. Lett.* **2020**, *27*, 1300–1304. [[CrossRef](#)]
28. Nadeem, Q.U.A.; Kammoun, A.; Chaaban, A.; Debbah, M.; Alouini, M.S. Asymptotic Max-Min SINR Analysis of Reconfigurable Intelligent Surface Assisted MISO Systems. *IEEE Trans. Wirel. Commun.* **2020**, *19*, 7748–7764. [[CrossRef](#)]
29. Liu, Y.; Liu, E.; Wang, R.; Geng, Y. Beamforming Designs and Performance Evaluations for Intelligent Reflecting Surface Enhanced Wireless Communication System with Hardware Impairments. *arXiv* **2020**, arXiv:2006.00664.
30. Feng, K.; Li, X.; Han, Y.; Jin, S.; Chen, Y. Physical Layer Security Enhancement Exploiting Intelligent Reflecting Surface. *IEEE Commun. Lett.* **2021**, *25*, 734–738. [[CrossRef](#)]
31. Guo, H.; Liang, Y.C.; Chen, J.; Larsson, E.G. Weighted Sum-Rate Optimization for Intelligent Reflecting Surface Enhanced Wireless Networks. *arXiv* **2019**, arXiv:1905.07920.
32. Wu, Q.; Zhang, R. Joint Active and Passive Beamforming Optimization for Intelligent Reflecting Surface Assisted SWIPT Under QoS Constraints. *IEEE J. Sel. Areas Commun.* **2020**, *38*, 1735–1748. [[CrossRef](#)]
33. Li, Y.; Xia, M.; Wu, Y.C. First-Order Algorithm for Content-Centric Sparse Multicast Beamforming in Large-Scale C-RAN. *IEEE Trans. Wirel. Commun.* **2018**, *17*, 5959–5974. [[CrossRef](#)]
34. Li, Z.; Wang, S.; Mu, P.; Wu, Y.C. Probabilistic Constrained Secure Transmissions: Variable-Rate Design and Performance Analysis. *IEEE Trans. Wirel. Commun.* **2020**, *19*, 2543–2557. [[CrossRef](#)]
35. Wang, S.; Hong, Y.; Wang, R.; Hao, Q.; Wu, Y.C.; Ng, D.W.K. Edge Federated Learning Via Unit-Modulus Over-The-Air Computation. *IEEE Trans. Commun.* **2022**, *70*, 3141–3156. [[CrossRef](#)]
36. Li, Y.; Xia, M.; Wu, Y.C. Energy-Efficient Precoding for Non-Orthogonal Multicast and Unicast Transmission via First-Order Algorithm. *IEEE Trans. Wirel. Commun.* **2019**, *18*, 4590–4604. [[CrossRef](#)]
37. Li, Y.; Xia, M.; Wu, Y.C. Caching at Base Stations With Multi-Cluster Multicast Wireless Backhaul via Accelerated First-Order Algorithms. *IEEE Trans. Wirel. Commun.* **2020**, *19*, 2920–2933. [[CrossRef](#)]
38. Hu, X.; Masouros, C.; Wong, K.K. Reconfigurable Intelligent Surface Aided Mobile Edge Computing: From Optimization-Based to Location-Only Learning-Based Solutions. *IEEE Trans. Commun.* **2021**, *69*, 3709–3725. [[CrossRef](#)]
39. Hua, S.; Zhou, Y.; Yang, K.; Shi, Y.; Wang, K. Reconfigurable Intelligent Surface for Green Edge Inference. *IEEE Trans. Green Commun. Netw.* **2021**, *5*, 964–979. [[CrossRef](#)]
40. Yang, H.; Chen, X.; Yang, F.; Xu, S.; Cao, X.; Li, M.; Gao, J. Design of Resistor-Loaded Reflectarray Elements for Both Amplitude and Phase Control. *IEEE Antennas Wirel. Propag. Lett.* **2017**, *16*, 1159–1162. [[CrossRef](#)]
41. Zhao, M.M.; Wu, Q.; Zhao, M.J.; Zhang, R. Exploiting Amplitude Control in Intelligent Reflecting Surface Aided Wireless Communication With Imperfect CSI. *IEEE Trans. Commun.* **2021**, *69*, 4216–4231. [[CrossRef](#)]
42. Abeywickrama, S.; Zhang, R.; Wu, Q.; Yuen, C. Intelligent Reflecting Surface: Practical Phase Shift Model and Beamforming Optimization. *IEEE Trans. Commun.* **2020**, *68*, 5849–5863. [[CrossRef](#)]
43. Wu, Q.; Zhang, R. Beamforming Optimization for Wireless Network Aided by Intelligent Reflecting Surface With Discrete Phase Shifts. *IEEE Trans. Commun.* **2020**, *68*, 1838–1851. [[CrossRef](#)]
44. Li, N.; Li, M.; Liu, Y.; Yuan, C.; Tao, X. Intelligent Reflecting Surface Assisted NOMA With Heterogeneous Internal Secrecy Requirements. *IEEE Wirel. Commun. Lett.* **2021**, *10*, 1103–1107. [[CrossRef](#)]
45. Luo, Z.; Ma, W.; So, A.M.; Ye, Y.; Zhang, S. Semidefinite Relaxation of Quadratic Optimization Problems. *IEEE Signal Process. Mag.* **2010**, *27*, 20–34. [[CrossRef](#)]
46. Nocedal, J.; Wright, S. *Numerical Optimization*; Springer: New York, NY, USA, 2006.
47. Shi, Y.; Zhang, J.; Chen, W.; Letaief, K.B. Generalized Sparse and Low-Rank Optimization for Ultra-Dense Networks. *IEEE Commun. Mag.* **2018**, *56*, 42–48. [[CrossRef](#)]
48. Li, Z.; Xia, M.; Wen, M.; Wu, Y.C. Massive Access in Secure NOMA under Imperfect CSI: Security Guaranteed Sum-Rate Maximization with First-Order Algorithm. *IEEE J. Sel. Areas Commun.* **2021**, *39*, 998–1014. [[CrossRef](#)]

49. He, X.; Wu, Y.C. Set Squeezing Procedure for Quadratically Perturbed Chance-Constrained Programming. *IEEE Trans. Signal Process.* **2021**, *69*, 682–694. [[CrossRef](#)]
50. Li, Z.; Wang, S.; Wen, M.; Wu, Y.C. Secure Multicast Energy-Efficiency Maximization with Massive RISs and Uncertain CSI: First-Order Algorithms and Convergence Analysis. *IEEE Trans. Wirel. Commun.* **2022**, *2022*, 1. [[CrossRef](#)]
51. Luedtke, J.; Ahmed, S. A sample approximation approach for optimization with probabilistic constraints. *SIAM J. Optim.* **2008**, *19*, 674–699. [[CrossRef](#)]
52. Di, B.; Zhang, H.; Song, L.; Li, Y.; Han, Z.; Poor, H.V. Hybrid Beamforming for Reconfigurable Intelligent Surface based Multi-User Communications: Achievable Rates With Limited Discrete Phase Shifts. *IEEE J. Sel. Areas Commun.* **2020**, *38*, 1809–1822. [[CrossRef](#)]
53. Li, Y.; Xia, M.; Wu, Y.C. Activity Detection for Massive Connectivity Under Frequency Offsets via First-Order Algorithms. *IEEE Trans. Wirel. Commun.* **2019**, *18*, 1988–2002. [[CrossRef](#)]
54. Tibshirani, R. Regression shrinkage and selection via the Lasso: A retrospective Series B Statistical methodology. *J. R. Stat. Soc.* **2011**, *73*, 273–282. [[CrossRef](#)]
55. Shao, M.; Li, Q.; Ma, W.K.; Thus, A.M.C. A Framework for One-Bit and Constant-Envelope Precoding Over Multiuser Massive MISO Channels. *IEEE Trans. Signal Process.* **2019**, *67*, 5309–5324. [[CrossRef](#)]
56. Pan, C.; Ren, H.; Wang, K.; Kolb, J.F.; El Kashlan, M.; Chen, M.; Di Renzo, M.; Hao, Y.; Wang, J.; Swindlehurst, A.L.; et al. Reconfigurable Intelligent Surfaces for 6G Systems: Principles, Applications, and Research Directions. *IEEE Commun. Mag.* **2021**, *59*, 14–20. [[CrossRef](#)]
57. Powell, W.B. *Approximate Dynamic Programming: Solving the Curses of Dimensionality*; John Wiley & Sons: Hoboken, NJ, USA, 2007.
58. Yang, H.; Xiong, Z.; Zhao, J.; Niyato, D.; Xiao, L.; Wu, Q. Deep Reinforcement Learning-Based Intelligent Reflecting Surface for Secure Wireless Communications. *IEEE Trans. Wirel. Commun.* **2021**, *20*, 375–388. [[CrossRef](#)]
59. Ma, Y.; Shen, Y.; Yu, X.; Zhang, J.; Song, S.; Letaief, K.B. Neural Calibration for Scalable Beamforming in FDD Massive MIMO with Implicit Channel Estimation. In Proceedings of the IEEE Global Communication Conference, Waikoloa, HI, USA, 9–13 December 2019; pp. 1–6.
60. Shen, Y.; Shi, Y.; Zhang, J.; Letaief, K.B. Graph Neural Networks for Scalable Radio Resource Management: Architecture Design and Theoretical Analysis. *IEEE J. Sel. Areas Commun.* **2021**, *39*, 101–115. [[CrossRef](#)]
61. Guo, J.; Yang, C. Learning Power Allocation for Multi-Cell-Multi-User Systems With Heterogeneous Graph Neural Networks. *IEEE Trans. Wirel. Commun.* **2022**, *21*, 884–897. [[CrossRef](#)]
62. Li, Y.; Chen, Z.; Wang, Y.; Yang, C.; Wu, Y.C. Heterogeneous Transformer: A Scale Adaptable Neural Network Architecture for Device Activity Detection. *arXiv* **2021**, arXiv:2112.10086.
63. Li, Y.; Chen, Z.; Liu, G.; Wu, Y.C.; Wong, K.K. Learning to Construct Nested Polar Codes: An Attention-Based Set-to-Element Model. *IEEE Commun. Lett.* **2021**, *25*, 3898–3902. [[CrossRef](#)]