

# 15 Generalized Low-Rank Optimization for Ultra-dense Fog-RANs

---

Yuanming Shi, Kai Yang, and Yang Yang

Expectations for new wireless networks have become higher since mobile data has grown exponentially and more diverse user services have emerged. Intensive deployment of network infrastructure can alleviate this pressure from the network architecture level and make it still competitive. For the improvement of total energy efficiency and network capacity, it is vital to deploy lots of radio access nodes that are equipped with computing and storage ability, and this is also beneficial for low-latency services when providing access for many moving equipments. The benefits of network densification can be exploited using the emerging fog radio access network (Fog-RAN) architecture through placing the computation and storage resources in the network edge. However, such great promises come with arduous scientific problems. It is a must to make some innovative work for the design of such complicated networks considering kinds of types resources. In this chapter, we will develop a generalized low-rank optimization model for performance enhancements in ultra-dense Fog-RANs, supported by various motivating design objectives, including mobile edge caching, wireless distributed computing, and topological interference alignment. Special attention is paid on algorithmic approaches to handle nonconvex low-rank optimization problems via Riemannian optimization.

## 15.1 Introduction

### 15.1.1 Fog-RANs

Cutting-edge technological advances like Internet of Things (IoTs), telemedicine, cyberphysical systems, and mobile edge intelligence exert tremendous pressure on the computation, communication, and storage capacities of wireless systems. To achieve ubiquitous connectivity for anybody, anything, and at anytime, intensive deployment is a good way [1]. The improvement of total energy efficiency [2], operation of moving applications with low latency [3], and access of many equipments [4] are also solved by this network architecture with computational and storable nodes. Such ultra-dense networks help to integrate with new network technologies, such as cloud radio access networks (Cloud-RANs) [5], edge caching [6], and mobile edge computing [7]. And these can be finished through combing attractive ways in various fields like software-defined networking [8], network function virtualization, and cloud

and edge computing. To leverage the resources across the end devices, edge nodes, and cloud data center, fog computing [9, 10] provides an end-to-end horizontal architecture to distribute computing, storage, control, and networking functions to the network edge.

By enabling fog computing in ultra-dense wireless networks, Fog-RANs thus provide a powerful platform to integrate the communication, storage, computing, control, and network function at the widespread access nodes [2, 3, 11]. Particularly, ultra-dense deployment of Fog-RANs, which are deployed intensively, can take full advantage of dense networks by placing signal processing units and interference management locally through SDN and edge computing [12]. In addition, the "content-centric" approach is becoming the mainstream of the Internet rather than the "connection-centric" to support massive data transmission [13]. The Fog-RANs can promote the development of the network at the architectural level and provide better quality content transmission through wireless edge caching [6]. At the same time, another problem faced by resource-constrained mobile networks is the increase in computing content in applications. And mobile edge computing offloads the computational process to local storage, which is a very worthwhile approach. Therefore, ultra-dense Fog-RANs are capable of achieving higher data rates and enabling low-latency content delivery, as well as providing ultra-reliable and low-latency communications for many machine-type communications [14].

### 15.1.2 Generalized Low-Rank Models

Ultra-dense Fog-RANs are very complicated to improve optimization, for which it is vital to utilize the accessible information in network side. If you want to effectively perform the encoding and decoding process, it is necessary to get some auxiliary information such as local information and some locally cached content. So, we design a novel low-rank matrix modeling framework, which combines the storage and computing resource to utilize the network-side information. Recently, there has been various research on interference management, which both considers the CSI and basic information, and the main tool for this problem is alignment, coordination, and multipoint transmission. However, the big pay of obtaining global CSI forces many works that mainly aim to reduce strategies such as the mixed and tardy CSI. Topology interference management (TIM) [15] is an effective solution, which only needs to know some network connection information. The reason for this is that the weak wireless channel considers path loss and shadow. But the TIM problem is a more difficult problem and may not have all the results in each scenario and thus become a linear index coding problem [15]. For the improvement of algorithm design in TIM problem, a novel suggestion that suggests modeling the network copulation mode in Fog-RANs as a fragmentary matrix is proposed. Then it is helpful to do the linear precoding and decoding in high quality, after the generalized matrix completion problem is formulated.

The strong modeling framework can also be utilized to deal with some other design difficulties in Fog-RANs. Being capable of caching content at the edge of the network

can effectively optimize throughput and latency, which is due to the cacheable space of densely deployed devices and access points [6]. In general, the placement and delivery of content are two mainly parts of content-centric communications. However, edge caching still has some serious problems unresolved because of the combination of wired and wireless in Fog-RANs. Fortunately, the information that is cached at different devices can be cached by the fragmentary matrix framework, which could also be expanded to distributed computing wireless networks [16]. MapReduce and Spark, as the distributed computing structures whose main thought is that intermediate values in the "map" phase based on the near accessible database, are seen as the side information for the "reduce" phase. So, this could help decrease the communication payment in the "shuffle" phase to get the intermediate values, which are not computed not in the "map" phase. Some other problems, like the modeling design of distributed computing systems, can also be solved by this fragmentary matrix modeling approach. The side information modeling matrix is beneficial to reduce interference over  $r$  channel, which is utilized, resulting in an interference-free channel with  $1/r$  degrees of freedom (DoF) – i.e., the first-order data description. It is not difficult to find that the number of channels is actually the matrix dimension, which also is the inverse of the realizable DoF. To maximize the realizable DoF, we could get this minimum of the matrix rank, generating the general optimization issue with lower rank considering the constraint set encoding the network side information.

### 15.1.3 Low-Rank Optimization Algorithms

#### Convex Optimization Approaches

Various applications about machine learning, recommendation systems, and big data analysis [17–19] benefit from the low-rank model. And some heuristic algorithms that can ensure the optimality have been proposed because there is a constraint of low-rank nonconvexity. As the rank of the matrix corresponds to the number of its nonzero singular values, we seek the convex surrogate for the rank function using the sum of its singular values – i.e., the nuclear norm. The nuclear norm is the convex hull of the collection of atomic unit-norm rank-one matrices and is thus the tightest convex relaxation of the rank function [17]. The resulting nuclear norm minimization can be further reformulated as a semidefinite program, which if convex could be solved in polynomial time. The optimality guarantees for the nuclear norm minimization can be normally established via convex geometry analysis and the theory conic integral geometry [20] for the problems in data science – e.g., low-rank matrix recovery. However, the nuclear norm minimization approach cannot scope to a big problem because of the high computation and memory costs, as we need to store and optimize the entire matrix. Furthermore, many low-rank problems may have no effective convex relaxation. In particular, the nuclear norm minimization approach is inapplicable to generalized low-rank optimization problem for topological interference management in Fog-RANs, as it always returns a full rank solution [21]. It is thus critical to consider alternative approaches, which can scale to large problem sizes and are effective to generalized low-rank models.

### Riemannian Optimization over Matrix Manifolds

Another promising solution for low-rank optimized minimization is on the basis of matrix factorization – e.g., the alternating minimization [18] and Riemannian optimization method [22]. It is finished through recasting the original low-rank minimization problem as solving a result of fixed-rank constraints optimization problems. The Riemannian optimization framework has the ability of utilizing the Riemannian quotient manifold of the fixed-rank matrices in the search space. What’s more, the Riemannian conjugate gradient and trust-region algorithms are globally convergent (i.e., they converge to first-order and second-order KKT points on manifolds [23]) with superlinear [24] and even quadratic convergence rates [23]. The Riemannian algorithm has a higher speed convergence and a smaller error solution, which is different from the alternating minimization and gradient descent algorithms. Basically, the objective function in Riemannian algorithm must have a certain smoothness [25]. As for the optimization problems with a high dimension, the Riemannian algorithms can have a global optimality – e.g., dictionary learning [26], generalized phase retrieval [26], and community detection problems [27]. If there are enough samples, it is definite to get the globally optimal value, and each point could be evaded by Riemannian trust-region algorithms [26, 27]. For general low-rank optimization problems in Fog-RANs, we shall propose well-designed reformulation and smooth approximation approaches in order to harness the benefits of Riemannian optimization techniques.

#### 15.1.4 Outline

In Section 15.2, we demonstrate that typical problems in ultra-dense Fog-RANs can be addressed by settling one or a sequence of generalized low-rank optimization problems. Section 15.3 presents a systematic Riemannian optimization framework to solve the generalized low-rank optimization problem by utilizing the quotient manifold architecture of fixed-rank matrices. Consequences of numerical experiments are illustrated in Section 15.5, then it is the summary and discussion in Section 15.6.

### 15.2 Generalized Low-Rank Models in Ultra-dense Fog-RANs

Here, we provide a generalized low-rank framework to optimize the computation, communication and storage resources in ultra-dense Fog-RANs. The typical examples of the low-rank optimization for topological interference alignment and cache-aided interference channel will be presented. More applications and unique challenges of settling the generalized low-rank optimization problems will also be discussed.

#### 15.2.1 A Generalized Low-Rank Framework

Consider the generalized low-rank optimization problem with the rank function as the objective

$$\mathcal{P}_1 : \underset{M \in \mathbb{C}^{m \times n}}{\text{minimize}} \text{rank}(M) \quad \text{subject to } M \in \mathcal{C}, \quad (15.1)$$

where matrix  $M \in \mathbb{C}^{m \times n}$  is the optimization variable and  $\mathcal{C}$  is the feasible constraint set. The rank-constrained version of the generalized low-rank optimization problem is given as follows:

$$\mathcal{P}_2 : \underset{M \in \mathbb{C}^{m \times n}}{\text{minimize}} f(M) \quad \text{subject to} \quad \text{rank}(M) = r, \quad (15.2)$$

where matrix  $M \in \mathbb{R}^{m \times n}$  is the optimization variable and  $r \leq \min\{m, n\}$  is known as a priori. The readers are encouraged to refer to [17, 18] for the applications of generalized low-rank models in data science. In this section, we will show that typical optimization problems in ultra-dense Fog-RANs could be formulated as a generalized low-rank optimization problem. Even though the structured low-rank formulation enjoys the benefits of modeling flexibility, the rank function is nonconvex, which brings computational challenges. Observing that the minimum rank of problem  $\mathcal{P}_1$  can be found by settling the fixed-rank optimization problem  $\mathcal{P}_2$ , we thus focus on developing a systematic framework for solving the rank-constrained optimization.

### 15.2.2 Low-Rank Optimization Examples in Fog-RANs

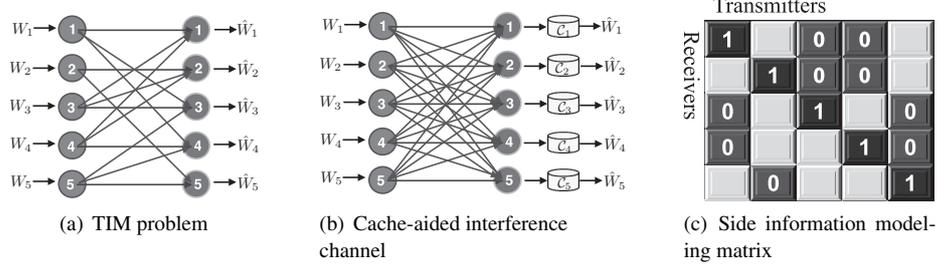
Ultra-dense Fog-RANs are strictly complicated to obtain an optimized result, for which it is not realistic to utilize the available network-side information for system design. To demonstrate the wide-ranging applications of low-rank optimization in Fog-RANs, we mainly focus on the topological interference management problem [21]. The proposed framework can be principled to extend more typical scenarios in Fog-RANs, including mobile edge caching systems [28] and on-device distributed computing systems [29], where the side information can be exploited to efficiently optimize across the communication, computation, and storage resources via the generalized low-rank optimization.

#### System Model

Consider a partially connected  $K$ -user interference channel with single-antenna transmitters and receivers shown in Figure 15.1(a), which is a typical architecture in Fog-RANs. Without loss of generality, we assume the  $k$ -th transmitter and the  $k$ -th receiver are connected. Each message  $W_k$  is available to transmitter  $k$  and shall be delivered to receiver  $k$ . The channel coefficient  $h_{kl} \in \mathbb{C}$  between transmitter  $l$  and user  $k$  is nonzero only for  $(k, l) \in \mathcal{S} \cup \{(i, i) : i = 1, \dots, K\}$ , where  $\mathcal{S}$  is the set of interference links. Block fading channel model is assumed – i.e.,  $h_{kl}$  stays static over  $r$  subsequent channels – during which the input and output relationship is given by

$$\mathbf{y}_k = h_{kk} \mathbf{x}_k + \sum_{(k,i) \in \mathcal{S}} h_{ki} \mathbf{x}_i + \mathbf{z}_k, \quad \forall k = 1, \dots, K. \quad (15.3)$$

$\mathbf{x}_k \in \mathbb{C}^r$  is the transmitted signal by the  $k$ -th transmitter over  $r$  channel uses,  $\mathbf{y}_k \in \mathbb{C}^r$  is the received signal at the  $k$ -th receiver, and  $\mathbf{z}_k \in \mathbb{C}^r$  represents the isotropic additive white Gaussian noise – i.e.,  $\mathbf{z}_k \sim \mathcal{CN}(\mathbf{0}, \sigma_k^2 \mathbf{I}_r)$ . The average power constraint of each transmitter is given by  $\frac{1}{r} \mathbb{E}[\|\mathbf{x}_i\|^2] \leq \rho$ , where  $\rho > 0$  is the maximum transmit power.



**Figure 15.1** (a) A typical example of TIM problem in the partially-connected 5-user interference channel. The index set of connected links is given by  $\mathcal{S} = \{(1, 3), (1, 4), (2, 3), (2, 4), (3, 1), (3, 5), (4, 1), (4, 5), (5, 1)\}$ . (b) A typical example of cache-aided five-user interference channel. The index set of cached messages at each receiver is given by  $\mathcal{C}_1 = \{2, 5\}$ ,  $\mathcal{C}_2 = \{1, 5\}$ ,  $\mathcal{C}_3 = \{2, 4\}$ ,  $\mathcal{C}_4 = \{2, 3\}$  and  $\mathcal{C}_5 = \{1, 3, 4\}$ . (c) Let  $\mathbf{M} = [M_{ij}] = [\mathbf{u}_i^H \mathbf{v}_j] \in \mathbb{C}^{K \times K}$ , where  $\mathbf{u}_i \in \mathbb{C}^r$  and  $\mathbf{v}_j \in \mathbb{C}^r$  are precoding and decoding vectors, and  $r$  denotes the number of channel extensions for transmission. The incomplete matrix  $\mathbf{M}$ , which has known entries indexed by  $\mathcal{S}$ , is the side information modeling matrix for (a) and (b).  $M_{ii} = 1$  represents the preservation of wished signals for each receiver,  $M_{ij} = 0, \forall (i, j) \in \mathcal{S}$  represents the cancellation of interferences, and  $M_{ij} = *, \forall (i, j) \notin \mathcal{S} \cup \{(i, i)\}$  can be arbitrary (unknown) values.

Messages  $W_1, W_2, \dots, W_K$  are assumed to be mutually independent. If there exists an encoding and decoding scheme such that as the code word’s length  $N$  approaches infinity, the probability of erroneous decoding for all messages can be arbitrarily small simultaneously [30], we claim that the rate tuple  $(R_1, R_2, \dots, R_K)$  is achievable. In the partially connected  $K$ -user interference channel, DoF is defined as [15, 31]

$$\text{DoF}_i = \limsup_{\rho \rightarrow \infty} \frac{R_i}{\log(\rho)}, \forall i. \quad (15.4)$$

The DoF area  $\mathcal{D}$  is further defined as the closure of achievable DoF tuples. In particular, the symmetric DoF is given by the maximum value  $\text{DoF}_{\text{sym}}$  such that DoF allocation  $\text{DoF}_i = \text{DoF}_{\text{sym}}, \forall i$  is inside the DoF region – i.e.,

$$\text{DoF}_{\text{sym}} = \limsup_{\rho \rightarrow \infty} \left[ \sup_{(R_{\text{sym}}, \dots, R_{\text{sym}}) \in \mathcal{D}} \frac{R_{\text{sym}}}{\log(\rho)} \right]. \quad (15.5)$$

It is particular interesting to study linear schemes for interference management because of the low complexity and the DoF optimality in a number of scenarios [15, 31, 32]. We thus focus on linear interference management schemes and take DoF as the performance metric, which is realized by exploiting the intrinsic correlation between DoF and matrix rank.

**Example 15.1 Topological interference management.** Linear interference alignment [31] is a powerful tool to mitigate interferences, which has the ability to achieve half the cake for every user in a  $K$ -user interference channel. Let  $\mathbf{v}_j \in \mathbb{C}^r$  and  $\mathbf{u}_i \in \mathbb{C}^r$  be the transmit and receiver beam-forming vector at transmitter  $j$  and receiver  $i$ , respectively.

Let  $s_k$  be the encoded information symbol for transmitter  $k$  with unit power – i.e.,  $\mathbb{E}\{|s_k|^2\} = 1$ . The received signal by receiver  $k$  is thus given as

$$\mathbf{y}_k = h_{kk}\mathbf{v}_k s_k + \sum_{(k,i) \in \mathcal{S}} h_{ki}\mathbf{v}_i s_i + \mathbf{z}_k. \quad (15.6)$$

Linear interference alignment is an asymptotically high SNR regime. To accomplish decoding, we impose the constraints that the desired signal space  $h_{kk}\mathbf{v}_k$  is complementary to the interference space  $\sum_{(k,i) \in \mathcal{S}} h_{ki}\mathbf{v}_i$  at each receiver  $i$ . That is, the interference terms are aligned and then canceled, while the desired signal is preserved by projecting the received signal vector  $\mathbf{y}_i$  onto the space  $\mathbf{u}_i$  – i.e.,

$$h_{kk}\mathbf{u}_k^H \mathbf{v}_k \neq 0, \forall k = 1, \dots, K, \quad (15.7)$$

$$h_{kj}\mathbf{u}_k^H \mathbf{v}_j = 0, (k, j) \in \mathcal{S}. \quad (15.8)$$

If both conditions (15.7) and (15.8) are satisfied, we can obtain parallel interference-free channels over  $r$  channel uses. Thus,  $1/r$  DoF is realized for message  $W_i$ . Note that we can further rewrite conditions (15.7) and (15.8) as the following channel independent conditions:

$$\mathbf{u}_k^H \mathbf{v}_k \neq 0, \forall k = 1, \dots, K, \quad (15.9)$$

$$\mathbf{u}_k^H \mathbf{v}_j = 0, (k, j) \in \mathcal{S}. \quad (15.10)$$

Therefore, the transceivers  $\mathbf{u}_i$ 's and  $\mathbf{v}_j$ 's are designed based only on the network topology information instead of acquiring instantaneous channel state information (CSI). This approach is termed topological interference management (TIM), with which the information symbol  $s_k$  can be estimated from

$$\tilde{s}_k = (\mathbf{u}_k^H \mathbf{v}_k)^{-1} \mathbf{u}_k^H \mathbf{y}_k. \quad (15.11)$$

To assist numerical algorithm design, we specify  $\mathbf{u}_k^H \mathbf{v}_k = 1$  for condition (15.7) without loss of generality. By defining  $\mathbf{M} = [\mathbf{u}_i^H \mathbf{v}_j] = [M_{ij}] \in \mathbb{C}^{K \times K}$ , conditions (15.7) and (15.8) can be further rewritten as

$$\mathcal{P}_\Omega(\mathbf{M}) = \mathbf{I}_K, \quad (15.12)$$

where  $\mathbf{I}_K$  is the  $K \times K$  identity matrix. The orthogonal projection onto  $\Omega$  – i.e.,  $\mathcal{P}_\Omega : \mathbb{C}^{K \times K} \rightarrow \mathbb{C}^{K \times K}$  – preserves  $M_{ij}$  for  $(i, j) \in \Omega$  and is zero for  $(i, j) \notin \Omega$ . Here the set  $\Omega$  is defined as  $\Omega = \{i \times j, (i, j) \in \mathcal{S} \cup \{(k, k)\}\}$ . To avoid trivial solutions, we make the assumption that  $r \leq \min\{m, n\}$ . Since  $\mathbf{M} = \mathbf{U}\mathbf{V}$  with  $\mathbf{U} = [\mathbf{u}_1, \dots, \mathbf{u}_K]^H \in \mathbb{C}^{K \times r}$  and  $\mathbf{V} = [\mathbf{v}_1, \dots, \mathbf{v}_K] \in \mathbb{C}^{r \times K}$ , we have  $\text{rank}(\mathbf{M}) = r$  and achievable DoF as  $1/r$  for each message. Thus, we can find the maximum achievable DoF by solving the following low-rank optimization problem [21]:

$$\mathcal{P}_{\text{TIM}} : \underset{\mathbf{M} \in \mathbb{R}^{K \times K}}{\text{minimize}} \text{rank}(\mathbf{M}) \quad \text{subject to } \mathcal{P}_\Omega(\mathbf{M}) = \mathbf{I}_K. \quad (15.13)$$

The modeling framework is illustrates in Figure 15.1, where Figure 15.1(a) shows a five-user interference channel and Figure 15.1(c) shows the corresponding modeling

matrix. The target of TIM is to achieve the side information modeling matrix, followed by extracting the precoder and decoder [21].

Note that we restrict problem  $\mathcal{P}_{\text{TIM}}$  to the true area without losing any performance despite of achievable DoFs. This is because the problem parameter  $\mathbf{I}_K$  is a real matrix, and if the element  $X_{ij} = \mathbf{u}_i^H \mathbf{v}_j, \forall i \neq j, (i, j) \notin \mathcal{S}$  is restricted to the real field, the corresponding signals will not contribute any interference.

**Example 15.2 Mobile edge caching.** In Fog-RANs, mobile edge caching is realized by pushing contents to mobile edges in advance at off-peak time, whose advantages have been demonstrated in facilitating interference management [33]. Thus, the caching capability can reduce the end-to-end latency within networks. Consider that mobile users are cache enabled, and we shall adopt the topological interference management technique to avoid the overwhelming CSI acquisition overhead in ultra-dense Fog-RANs. Let  $\mathcal{R}_k \subseteq \{1, \dots, K\}$  ( $k \notin \mathcal{R}_k$ ) be the index set of messages cached at mobile user  $k$ , for which they have been delivered during off-peak time. Thus, the received signal at mobile user  $k$  can be rewritten as

$$\mathbf{y}_k = h_{kk} \mathbf{v}_k s_k + \sum_{i \in \mathcal{R}_k, (k, i) \in \mathcal{S}} h_{ki} \mathbf{v}_i s_i + \sum_{j \notin \mathcal{R}_k, (k, j) \in \mathcal{S}} h_{kj} \mathbf{v}_j s_j + \mathbf{z}_k. \quad (15.14)$$

Likewise, interference alignment is achieved by preserving the desired signals while canceling interferences, resulting in the following topological interference alignment condition:

$$\mathbf{u}_k^H \mathbf{v}_k \neq 0, \forall k = 1, \dots, K, \quad (15.15)$$

$$\mathbf{u}_k^H \mathbf{v}_j = 0, (k, j) \in \mathcal{S}, j \notin \mathcal{R}_k. \quad (15.16)$$

Then the desired signal  $s_k$  will be decoded from

$$\tilde{s}_k = (\mathbf{u}_k^H \mathbf{v}_k)^{-1} \mathbf{u}_k^H \left( \mathbf{y}_k - \sum_{j \in \mathcal{R}_k} h_{kj} \mathbf{v}_j s_j \right), \quad (15.17)$$

while achieving  $1/r$  symmetric DoF for each message delivery. Therefore, by denoting  $\mathbf{M} = [M_{ij}] = [\mathbf{u}_i^H \mathbf{v}_j]$ , the transceiver could be achieved by settling the following low-rank optimization problem:

$$\mathcal{P}_{\text{cache}} : \underset{\mathbf{M} \in \mathbb{R}^{K \times K}}{\text{minimize}} \text{rank}(\mathbf{M}) \quad \text{subject to } \mathcal{P}_{\Omega}(\mathbf{M}) = \mathbf{I}_K, \quad (15.18)$$

where  $\Omega = \{i \times j, (i, j) \in \mathcal{S} \cup \{(k, k)\}, j \notin \mathcal{R}_i\}$ . Figure 15.1(b) gives an illustrative example of a cache-enabled five-user interference channel, where the side information is shown in the side information modeling matrix in Figure 15.1(c).

Note that we consider the single data stream case in these examples – i.e., each message  $W_k$  is represented with a single information symbol  $s_k \in \mathbb{C}$ . They can be easily expanded to ordinary multiple data streams cases, where message  $W_k$  has representation vector  $\mathbf{s}_k \in \mathbb{C}^{d_k}$  with  $d_k$  data streams.

### Discussions

A generalized low-rank optimization framework has been widely developed for optimizing across communication resources, computation resources, storage resources, and data analysis in the ecosystem of ultra-dense Fog-RANs [1], including

- interference management for achievable DoF [21, 34] and sum-rate [35] maximization;
- mobile edge caching with cache-enabled receivers [28] and transmitters [36];
- user admission control [37] with topological interference management;
- data shuffling in wireless distributed fog computing systems [29];
- low-latency communication in massive IoT networks [38]; and
- high-dimensional data processing for mobile edge caching in Fog-RANs [39].

These applications demonstrate that the generalized low-rank optimization framework is a powerful tool for modeling system design and optimization problems in ultra-dense Fog-RANs. However, the nonconvexity of the rank function makes it critical to solve the generalized low-rank optimization problem scaling to large network sizes. Furthermore, the nuclear norm relaxation approach is inapplicable to problems  $\mathcal{P}_{\text{TIM}}$  and  $\mathcal{P}_{\text{cache}}$  since  $\text{Trace}(M) \leq \|M\|_*$  with  $\|\cdot\|_*$  is the nuclear norm. For this reason, problem  $\mathcal{P}$  could be seen as a generalized low-rank optimization problem lacking in effective convex relaxation approaches, which motivates the design of nonconvex paradigms in Fog-RANs.

## 15.3 The Power of Nonconvex Paradigms for Ultra-dense Fog-RANs

In this section, we illustrate some fresh trends in nonconvex optimization algorithms. The generalized low-rank optimization problems fall in the category of solving a sequence of optimization problems on the matrix manifold. We then introduce the general Riemannian optimization framework and its implementation details.

### 15.3.1 Low-Rank Optimization via Nonconvex Factorization

Recently, a new line of works focusing on developing efficient nonconvex procedures for low-rank optimization problems has attracted much attention, and some among them have provided an optimality guarantee [19]. Progress has been made on nonconvex approaches, including projected gradient methods, stochastic gradient methods, conditional gradient methods, Riemannian manifold optimization algorithms, for machine learning problems and high-dimensional statistical problems such as phase retrieval, low-rank matrix completion, and blind deconvolution. Particularly, manifold optimization is becoming a powerful and general approach to address nonconvex optimization problems by exploiting manifold structures of problems. Manifold structures such as rank constraint and orthogonality are ubiquitous in machine learning applications, including dimensionality reduction, sensor network localization, low-rank matrix recovery, community detection, and phase synchronization.

General formulation of Riemannian optimization framework [25] is given by

$$\mathcal{P}_{\text{manifold}} : \underset{\mathbf{X} \in \mathcal{M}}{\text{minimize}} f(\mathbf{X}), \quad (15.19)$$

where  $f(\mathbf{X})$  is a smooth ( $C^\infty$ ) objective function on a smooth manifold  $\mathcal{M}$ . Observing that the set of fixed-rank matrices

$$\mathcal{M} := \{\mathbf{X} \in \mathbb{R}^{K \times K} : \text{rank}(\mathbf{X}) = r\} \quad (15.20)$$

is a smooth manifold [22], our generalized low-rank optimization problems  $\mathcal{P}_{\text{TIM}}$  and  $\mathcal{P}_{\text{cache}}$  thus can be addressed by alternatively performing the following fixed-rank optimization

$$\begin{aligned} \underset{\mathbf{X} \in \mathbb{R}^{K \times K}}{\text{minimize}} f(\mathbf{X}) &= \|\mathcal{P}_\Omega(\mathbf{X}) - \mathbf{I}_K\|_F^2 \\ \text{subject to} & \text{rank}(\mathbf{X}) = r, \end{aligned} \quad (15.21)$$

and increasing rank  $r$  [21, 40]. The minimum rank of matrix  $\mathbf{X}$  can thereby be detected as the minimum  $r$  such that the affine constraint  $\mathcal{P}_\Omega(\mathbf{X}) = \mathbf{I}_K$  is satisfied.

Riemannian optimization on fixed-rank manifold is based on reparameterizing a fixed-rank matrix via matrix factorization  $\mathbf{X} = \mathbf{U}\mathbf{V}^\top$ . Note that this low-rank matrix factorization is not unique as  $\mathbf{X}$  remains invariant under the transformation

$$(\mathbf{U}, \mathbf{V}) \mapsto (\mathbf{U}\mathbf{S}^{-1}, \mathbf{V}\mathbf{S}^\top) \quad (15.22)$$

for all matrices  $\mathbf{S} \in \text{GL}(r)$  – i.e., the set of  $r \times r$  full rank matrices. This indeterminacy makes the critical points of the objective function  $f(\mathbf{U}\mathbf{V}^\top)$  *not isolated* on  $\mathbb{R}^{K \times r} \times \mathbb{R}^{K \times r}$ .

We encode the invariance transformation (15.22) in an abstract search space and optimize problems directly over a set of equivalence classes – i.e.,

$$[(\mathbf{U}, \mathbf{V})] := \{(\mathbf{U}\mathbf{S}^{-1}, \mathbf{V}\mathbf{S}^\top) : \mathbf{S} \in \text{GL}(r)\}. \quad (15.23)$$

The set of equivalence classes is termed as the *quotient space* and is denoted by

$$\mathcal{M}_r := \mathcal{M}/\text{GL}(r), \quad (15.24)$$

where the total space  $\mathcal{M}$  is the product space  $\mathbb{R}^{K \times r} \times \mathbb{R}^{K \times r}$ . Consequently, problem (15.21) can be studied on the quotient space – i.e.,

$$\underset{[\mathbf{X}] \in \mathcal{M}_r}{\text{minimize}} f([\mathbf{X}]), \quad (15.25)$$

where  $[\mathbf{X}] = [(\mathbf{U}, \mathbf{V})]$  is defined in (15.23).

### 15.3.2 The Framework of Riemannian Optimization

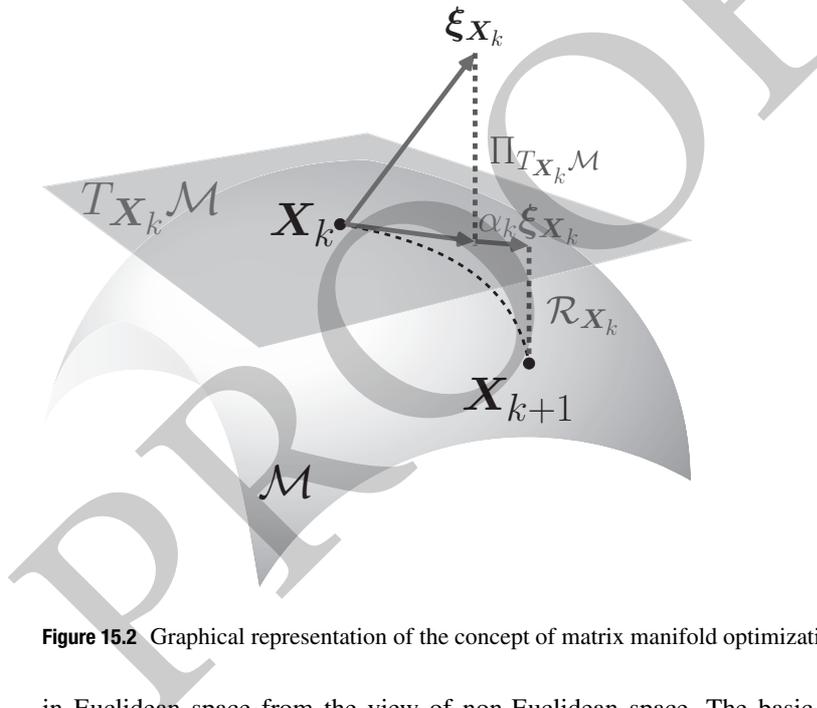
We will first give a brief introduction on Riemannian optimization framwrok at a high-level standpoint. Riemannian optimization generalizes the concepts for standard unconstrained optimization algorithms, including gradient and Hessian, from Euclidean space to Riemannian manifold, thereby creating a new paradigm of studying the constraints

---

**ALGORITHM 15.1** *Riemannian optimization framework for problem  $\mathcal{P}_{\text{manifold}}$*

---

- 1: **Initialize:** initial point  $\mathbf{X}_0$ ,  $k = 0$
  - 2: **while** not converged **do**
  - 3:   Choose a descent direction  $\xi_k$  and a step size  $\alpha_k$  via implementing different algorithms such as conjugate gradient method, trust-region method, etc.
  - 4:   Update  $\mathbf{X}_{k+1} = \mathcal{R}_{\mathbf{X}_k}(\alpha_k \xi_k)$
  - 5:    $k = k + 1$ .
  - 6: **end while**
- Output:**  $\mathbf{X}_k$
- 



**Figure 15.2** Graphical representation of the concept of matrix manifold optimization

in Euclidean space from the view of non-Euclidean space. The basic elements of a Riemannian optimization framework in the computational space  $\mathcal{M}$  consist of choosing a descent direction  $\xi_{\mathbf{X}}$ , determining a step size  $\alpha_k$  in the  $k$ -th iteration, and defining a pullback operation, called *retraction*, to make sure that the next point lies in the manifold. A descent direction  $\xi_{\mathbf{X}}$  is chosen from the *tangent space* of manifold  $\mathcal{M}$ , which is the linearization of the manifold at  $\mathbf{X}_k$  and denoted as  $T_{\mathbf{X}_k} \mathcal{M}$ . With the computed descent direction  $\xi_{\mathbf{X}}$  and step size  $\alpha_k$ , the notion of moving from  $\mathbf{X}_k$  in the direction of  $\alpha_k \xi_{\mathbf{X}}$  is given by the pullback operation  $\mathcal{R}_{\mathbf{X}_k}(\alpha_k \xi_k)$ , where the retraction operator is defined as  $\mathcal{R}_{\mathbf{X}} : \mathbb{R}^{K \times K} \rightarrow \mathcal{M}$ . Based on the above notions, a general matrix manifold optimization algorithm is presented in Algorithm 15.1, whose graphic representation is illustrated in Figure 15.2. We will show more details about Riemannian optimization in Section 15.4.

### 15.3.3 Practical Implementation

We have thus far presented the high-level Riemannian optimization framework. Here, we will discuss the implementation of Riemannian optimization algorithms.

#### **Manopt: A MATLAB Toolbox for Optimization on Manifolds**

Optimization on manifolds, or Riemannian optimization, aims at finding optimizers (at least local) for problem  $\mathcal{P}_{\text{manifold}}$ , where the search space  $\mathcal{M}$  is a smooth manifold endowed with a Riemannian structure. A *Manopt* toolbox [41] is built thanks to the maturity of the smooth Riemannian optimization theory, its widespread applications, and its excellent performance. The toolbox architecture consists of different types of manifolds, solvers for different algorithms, and problem-dependent descriptions. To address an optimization problem on a predefined Riemannian manifold, we can simply choose the manifold from the library, make a description of the cost function and derivatives (possible gradient and Hessian) on this manifold, and pass the parameters on to a solver. Some accompanying tools have also been designed, such as checking its derivatives and approximating its Hessian based on the gradient numerically. A manifold in *Manopt* is obtained by calling a factory. The descriptions of its manifold structure include retraction, projection on tangent spaces, operation for converting a Euclidean gradient and Hessian to Riemannian gradient and Riemannian Hessian, etc.

#### **Pymanopt: A Python Toolbox for Optimization on Manifolds**

The *Pymanopt* toolbox [42] for Riemannian optimization is built with Python. To further improve the usability for the average user, *Pymanopt* supports various Python libraries for automated differentiation. This is based on the fact that computing and implementing gradients and Hessian is laborious and error prone, especially for cost functions with matrix variables. For example, the Riemannian trust-region algorithm requires the Hessian information (which is a high-rank tensor). It is tedious and error prone to implement even for an experienced user. To address these difficulties, *Pymanopt* combines Riemannian optimization and automated differentiation, thereby improving the convenience and efficiency for practitioners. This makes the implementation of the Riemannian trust-region method in *Pymanopt* as easy as using one of the first-order or even derivative-free methods.

## 15.4 Matrix Optimization on Quotient Manifold

We present the details of Riemannian optimization algorithms, which are used to settle the fixed-rank optimization problem (15.25). It is performed on the quotient manifolds and exploits the symmetry structure (i.e., the quotient manifold geometry) in the search space of the fixed-rank constraint and the Hessian of the least-squares structure of the cost function. Specifically, the problem structures will be presented in Section 15.4.1. The matrix representations of all the optimization ingredients are provided in Section 15.4.2, and then algorithm implementation details are provided

in Section 15.4.3. Section 15.4.4 provides the convergence rates and computational complexity.

### 15.4.1 Problem Structures for Fixed-Rank Matrices

We use the symmetry in the fixed-rank constraint for efficient Riemannian optimization algorithms design.

#### Quotient Manifold

Let  $\sim$  be an equivalence relation in the *total* (computational) space  $\mathcal{M}$ . By this equivalence property, the quotient manifold  $\mathcal{M}/\sim$  includes all elements that are *equivalence classes* – i.e.,  $[\mathbf{X}] = \{\mathbf{Y} \in \mathcal{M} : \mathbf{Y} \sim \mathbf{X}\}$ . That is, if  $[\mathbf{X}]$  is an element in the quotient manifold  $\mathcal{M}/\sim$ , then it has matrix representation  $\mathbf{X}$  in  $\mathcal{M}$ . In the context of the presented fixed-rank constraint,  $\mathcal{M}/\sim$  is identified with the fixed-rank manifold  $\mathcal{M}_r$ . The optimization on a quotient manifold relies on defining the “linearization” of the search space, choosing a “search” direction, and determining a way to “move” on the manifold. We will show the details for developing these objects that allow us to develop a first-order Riemannian conjugate gradient algorithm and second-order Riemannian trust-region algorithm in the following part.

The quotient manifold  $\mathcal{M}/\sim$  is an abstract space. So we shall define a matrix representation in the tangent space  $T_{\mathbf{X}}\mathcal{M}$  for each element of the tangent space  $T_{[\mathbf{X}]}(\mathcal{M}/\sim)$  at  $[\mathbf{X}]$ . Equivalently, we should restrict the matrix representation of  $T_{[\mathbf{X}]}(\mathcal{M}/\sim)$  to the directions in the tangent space  $T_{\mathbf{X}}\mathcal{M}$  on  $\mathcal{M}$  at  $\mathbf{X}$  without inducing any displacement along the equivalence class  $[\mathbf{X}]$ . To achieve this goal, we decompose the tangent space  $T_{\mathbf{X}}\mathcal{M}$  into two complementary subspaces – i.e., the *vertical* space and the *horizontal* subspace, where  $\mathcal{V}_{\mathbf{X}} \oplus \mathcal{H}_{\mathbf{X}} = T_{\mathbf{X}}\mathcal{M}$ . The vertical space  $\mathcal{V}_{\mathbf{X}}$  is given by the space tangent to the equivalence class  $[\mathbf{X}]$ . Its complementary subspace – i.e., the horizontal space  $\mathcal{H}_{\mathbf{X}}$  – can provide a modeling representation of the abstract tangent space  $T_{[\mathbf{X}]}(\mathcal{M}/\sim)$  [25, section 3.5.8]. We can always find a unique element in the horizontal space  $\xi_{\mathbf{X}} \in \mathcal{H}_{\mathbf{X}}$  for any tangent vector  $\xi_{[\mathbf{X}]} \in T_{[\mathbf{X}]}(\mathcal{M}/\sim)$  in the abstract space at  $[\mathbf{X}]$ . This unique element  $\xi_{\mathbf{X}}$  is called the *horizontal lift* of  $\xi_{[\mathbf{X}]}$ . The horizontal space is chosen as an *orthogonal complement* of  $\mathcal{V}_{\mathbf{X}}$  in the sense of a Riemannian metric (an inner product), which is a subspace of  $T_{\mathbf{X}}\mathcal{M}$ .

An inner product for any two elements in the tangent space  $g_{\mathbf{X}} : T_{\mathbf{X}}\mathcal{M} \times T_{\mathbf{X}}\mathcal{M} \rightarrow \mathbb{R}$  at  $\mathbf{X} \in \mathcal{M}$  is called the Riemannian metric on the total space. It further defines a Riemannian metric  $g_{[\mathbf{X}]} : T_{[\mathbf{X}]}(\mathcal{M}/\sim) \times T_{[\mathbf{X}]}(\mathcal{M}/\sim) \rightarrow \mathbb{R}$  on the quotient manifold  $\mathcal{M}/\sim$ , which is given by

$$g_{[\mathbf{X}]}(\xi_{[\mathbf{X}]}, \eta_{[\mathbf{X}]}) := g_{\mathbf{X}}(\xi_{\mathbf{X}}, \eta_{\mathbf{X}}). \quad (15.26)$$

Here  $\eta_{[\mathbf{X}]}$  and  $\xi_{[\mathbf{X}]}$  are two vectors in the abstract tangent space  $T_{[\mathbf{X}]}(\mathcal{M}/\sim)$ , and  $\eta_{\mathbf{X}}, \xi_{\mathbf{X}}$  give their horizontal lifts in  $\mathcal{H}_{\mathbf{X}}$  at  $\mathbf{X}$ . Note that the definition of Riemannian metric on quotient manifold requires that the expression  $g_{\mathbf{X}}(\xi_{\mathbf{X}}, \eta_{\mathbf{X}})$  on the total space is irrelevant to the specific representation along the equivalence class  $[\mathbf{X}]$ . That is, if  $\mathbf{Y} \sim \mathbf{X}$ , then the metric in (15.26) obeys the equality  $g_{\mathbf{X}}(\xi_{\mathbf{X}}, \eta_{\mathbf{X}}) = g_{\mathbf{Y}}(\xi_{\mathbf{Y}}, \eta_{\mathbf{Y}})$

for  $\eta_Y$  and  $\xi_Y$  being the horizontal lifts of  $\eta_{[X]}$  and  $\xi_{[X]}$  at  $Y$ . We call that such a Riemannian metric is *invariant* to the equivalence relation  $\sim$ .

### Riemannian Metric

For fixed-rank matrices, a particular Riemannian metric that is invariant on the total space  $\mathcal{M}$  and takes the symmetry (15.22) into account is given by

$$g_X(\xi_X, \eta_X) = \text{Tr}((V^T V)\xi_U^T \eta_U) + \text{Tr}((U^T U)\xi_V^T \eta_V), \quad (15.27)$$

where  $X = (U, V)$  is the factorization model and  $\xi_X, \eta_X \in T_X \mathcal{M}$ . Note that  $(\xi_U, \xi_V) \in \mathbb{R}^{K \times r} \times \mathbb{R}^{K \times r}$  is the matrix characterization of  $\xi_X$  (and similarly  $\eta_X$ ).

Here we will show that (15.27) remains invariant to the transformation (15.22). Assuming that  $Y \in [X]$  is another element in the equivalent class with matrix representation  $(UM^{-1}, VM)$  for a particular nonsingular square matrix  $M$ , the matrix representations of the tangent vectors  $\xi_Y$  and  $\eta_Y$  are given by  $(\xi_{UM^{-1}}, \xi_{VM^T})$  and  $(\eta_{UM^{-1}}, \eta_{VM^T})$ , respectively. In addition, if the *horizontal lifts* of  $\xi_{[X]}$  (similarly for  $\eta_{[X]}$ ) at  $X$  and  $Y$  are respectively given by  $\xi_X$  and  $\xi_Y$ , we have  $\xi_{UM^{-1}} = \xi_U M^{-1}$  and  $\xi_{VM^T} = \xi_V M^T$  [25, example 3.5.4] (similar for  $\eta_Y$ ). It is then obtained that  $g_X(\xi_X, \eta_X) = g_Y(\xi_Y, \eta_Y)$ , which demonstrates the invariance of the metric (15.27) to the transformation (15.22). Therefore, a unique metric is defined on the quotient space  $\mathcal{M}/\sim$ .

## 15.4.2 Matrix Representation for the Quotient Manifolds

Given the metric (15.27) defined on  $\mathcal{M}$ , the development of Riemannian ingredients required for Riemannian optimization algorithms follows [25]. The tangent space  $T_X \mathcal{M}$ , horizontal space  $\mathcal{H}_X$ , and vertical space  $\mathcal{V}_X$  have the following matrix characterizations:

$$T_X \mathcal{M} = \mathbb{R}^{K \times r} \times \mathbb{R}^{K \times r}, \quad (15.28)$$

$$\mathcal{H}_x = \{(\zeta_U, \zeta_V) : U^T \zeta_U V^T V = U^T U \zeta_V^T V\}, \quad (15.29)$$

$$\mathcal{V}_x = \{(-U\Lambda, V\Lambda^T) : \Lambda \in \mathbb{R}^{r \times r}\}, \quad (15.30)$$

where  $\zeta_U, \zeta_V \in \mathbb{R}^{K \times r}$ .

We further need to define a projection from the tangent space onto the horizontal space, which is denoted by the linear operator  $\Pi_X : T_X \mathcal{M} \mapsto \mathcal{H}_X$ . For an element  $\eta_X \in T_X \mathcal{M}$ , the projection onto the horizontal space is given by

$$\Pi_X(\eta_X) = (\eta_U + U\Lambda, \eta_V - V\Lambda^T), \quad (15.31)$$

where  $\Lambda \in \mathbb{R}^{r \times r}$  is obtained by ensuring that the projection of  $\eta_X$  belongs to the horizontal space in (15.29) and given by

$$\begin{aligned} U^T(\eta_U + U\Lambda)V^T V &= U^T U(\eta_V - V\Lambda^T)^T V \\ \Rightarrow \Lambda &= 0.5(\eta_V^T V(V^T V)^{-1} - (U^T U)^{-1}U^T \eta_U). \end{aligned} \quad (15.32)$$

By choosing the metric in (15.27) and the horizontal space as the orthogonal complement of  $\mathcal{V}_{\mathbf{X}}$ , the quotient manifold  $\mathcal{M}/\sim$  becomes a *Riemannian submersion* of  $(\mathcal{M}, g)$  [25, section 3.6.2]. This makes it convenient to develop matrix representations of the Riemannian ingredients on the quotient manifold  $\mathcal{M}/\sim$ , which will be introduced in the following part.

### Riemannian Gradient

The horizontal lift of the Riemannian gradient  $\text{grad}_{[\mathbf{X}]}f$  of  $f$  on  $\mathcal{M}/\sim$  has matrix representation

$$\text{horizontal lift of } \text{grad}_{[\mathbf{X}]}f = \text{grad}_{\mathbf{X}}f = \left( \frac{\partial f}{\partial \mathbf{U}}(\mathbf{V}^T \mathbf{V})^{-1}, \frac{\partial f}{\partial \mathbf{V}}(\mathbf{U}^T \mathbf{U})^{-1} \right), \quad (15.33)$$

where  $\text{grad}_{\mathbf{X}}f$  is the Riemannian gradient of  $f$  in  $\mathcal{M}$ .  $\partial f/\partial \mathbf{U}$  and  $\partial f/\partial \mathbf{V}$  denote the *partial derivatives* of function  $f$  with respect to  $\mathbf{U}$  and  $\mathbf{V}$ , respectively.

### Riemannian Hessian

In order to develop second-order Riemannian optimization algorithms, we need to define the *connection*  $\nabla_{\xi_{\mathbf{X}}}\eta_{\mathbf{X}}$ , which plays the role of the directional derivative of the gradient along a search direction. Since both the Riemannian gradient and the search direction should be elements in the tangent space for Riemannian optimization, Riemannian connection is given as a *covariant derivative* of vector field  $\eta_{\mathbf{X}}$  with respect to another vector field  $\xi_{\mathbf{X}}$ . By applying the *Koszul* formula [25, theorem 5.3.1], the matrix characterization of the Riemannian connection in the total space  $\mathcal{M}$  can be computed as

$$\nabla_{\xi_{\mathbf{X}}}\eta_{\mathbf{X}} = \text{D}\eta_{\mathbf{X}}[\xi_{\mathbf{X}}] + (\mathbf{A}_{\mathbf{U}}, \mathbf{A}_{\mathbf{V}}) \quad (15.34)$$

$$\begin{aligned} \mathbf{A}_{\mathbf{U}} &= \eta_{\mathbf{U}}\text{Sym}(\xi_{\mathbf{V}}^T \mathbf{V})(\mathbf{V}^T \mathbf{V})^{-1} + \xi_{\mathbf{U}}\text{Sym}(\eta_{\mathbf{V}}^T \mathbf{V})(\mathbf{V}^T \mathbf{V})^{-1} \\ &\quad - \mathbf{U}\text{Sym}(\eta_{\mathbf{V}}^T \xi_{\mathbf{V}})(\mathbf{V}^T \mathbf{V})^{-1} \end{aligned} \quad (15.35)$$

$$\begin{aligned} \mathbf{A}_{\mathbf{V}} &= \eta_{\mathbf{V}}\text{Sym}(\xi_{\mathbf{U}}^T \mathbf{U})(\mathbf{U}^T \mathbf{U})^{-1} + \xi_{\mathbf{V}}\text{Sym}(\eta_{\mathbf{U}}^T \mathbf{U})(\mathbf{U}^T \mathbf{U})^{-1} \\ &\quad - \mathbf{V}\text{Sym}(\eta_{\mathbf{U}}^T \xi_{\mathbf{U}})(\mathbf{U}^T \mathbf{U})^{-1}, \end{aligned} \quad (15.36)$$

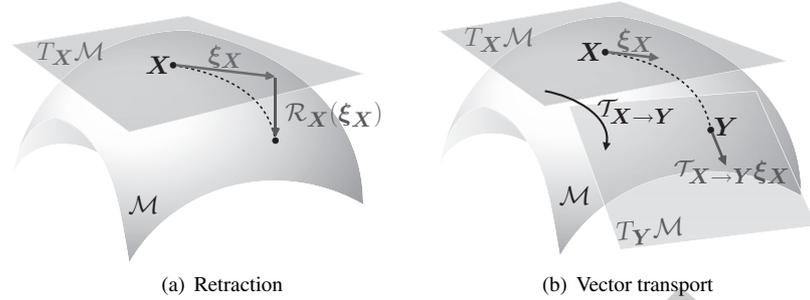
where the symmetric operation  $\text{Sym}(\mathbf{Z})$  extracts the symmetric part of matrix  $\mathbf{Z}$  – i.e.,  $\text{Sym}(\mathbf{Z}) = (\mathbf{Z} + \mathbf{Z}^T)/2$ .  $\text{D}\eta_{\mathbf{X}}[\xi_{\mathbf{X}}]$  is the Euclidean directional derivative, which is given by

$$\text{D}\eta_{\mathbf{X}}[\xi_{\mathbf{X}}] := \lim_{t \rightarrow 0} (\eta_{\mathbf{X}+t\xi_{\mathbf{X}}} - \eta_{\mathbf{X}})/t. \quad (15.37)$$

The Riemannian connection on the quotient manifold  $\mathcal{M}/\sim$ , denoted as  $\nabla_{\xi_{[\mathbf{X}]}}\eta_{[\mathbf{X}]}$ , has a unique matrix representation [25, proposition 5.3.3] given by

$$\text{horizontal lift of } \nabla_{\xi_{[\mathbf{X}]}}\eta_{[\mathbf{X}]} = \Pi_{\mathbf{X}}(\nabla_{\xi_{\mathbf{X}}}\eta_{\mathbf{X}}), \quad (15.38)$$

where  $\xi_{[\mathbf{X}]}$  and  $\eta_{[\mathbf{X}]}$  are two vector fields in the quotient manifold  $\mathcal{M}/\sim$ , and their horizontal lifts in the total space  $\mathcal{M}$  are respectively given by  $\xi_{\mathbf{X}}$  and  $\eta_{\mathbf{X}}$ . The projection operator  $\Pi_{\mathbf{X}}(\cdot)$  follows the definition in (15.31).



**Figure 15.3** Visual representation of retraction and vector transport in Riemannian optimization

Based on the definition of Riemannian connection, we further define the *Riemannian Hessian operator*  $\text{Hess}_{[\mathbf{X}]} f[\xi_{[\mathbf{X}]}]$  as the directional derivative of the Riemannian gradient in the direction  $\xi_{[\mathbf{X}]}$ . The matrix expression for the horizontal lift of the Riemannian Hessian in  $\mathcal{M}/\sim$  can be derived from (15.38) and (15.34) and is given by

$$\text{horizontal lift of } \text{Hess}_{[\mathbf{X}]} f[\xi_{[\mathbf{X}]}] = \Pi_{\mathbf{X}}(\nabla_{\xi_{\mathbf{X}}} \text{grad}_{\mathbf{X}} f), \quad (15.39)$$

where  $\xi_{[\mathbf{X}]} \in T_{[\mathbf{X}]}(\mathcal{M}/\sim)$  is a vector field in the tangent space of the quotient manifold  $\mathcal{M}/\sim$  and  $\xi_{\mathbf{X}} \in \mathcal{H}_{\mathbf{X}}$  is its horizontal lift.

### Retraction

As we discussed, an iterative Riemannian optimization algorithm requires defining the *retraction* operation  $R_{\mathbf{X}} : \mathcal{H}_{\mathbf{X}} \rightarrow \mathcal{M} : \xi_{\mathbf{X}} \mapsto R_{\mathbf{X}}(\xi_{\mathbf{X}})$  [25, definition 4.1.1], which represents “moving in a direction” on the Riemannian manifold. A natural update  $\mathbf{X}_+ = R_{\mathbf{X}}(\xi_{\mathbf{X}})$  on the manifold  $\mathcal{M}$  with a search direction  $\xi_{\mathbf{X}} = (\xi_U, \xi_V) \in \mathcal{H}_{\mathbf{X}}$  is given by

$$R_U(\xi_U) = U + \xi_U, \quad R_V(\xi_V) = V + \xi_V, \quad (15.40)$$

which translates into the update  $[\mathbf{X}_+] = [R_{\mathbf{X}}(\xi_{\mathbf{X}})]$  on  $\mathcal{M}/\sim$ . The retraction operation is illustrated in Figure 15.3(a).

### Vector Transport

A vector transport [25, definition 8.1.1]  $\mathcal{T}_{\mathbf{X} \rightarrow \mathbf{Y}} \xi_{\mathbf{X}}$  on a manifold  $\mathcal{M}$ , is a smooth mapping that transports a vector  $\xi_{\mathbf{X}}$  in the tangent space at  $\mathbf{X}$  to a vector in the tangent space at  $\mathbf{Y}$  under certain conditions. For our problem where the total space is an open subset of the Euclidean space, the matrix representation of the vector transport is given by

$$\text{horizontal lift of } \mathcal{T}_{[\mathbf{X}] \rightarrow [\mathbf{Y}]} \xi_{[\mathbf{X}]} = \Pi_{\mathbf{Y}}(\xi_{\mathbf{X}}). \quad (15.41)$$

The concept of vector transport is illustrated in Figure 15.3(b).

All concrete Riemannian ingredients are summarized in Table 15.1.

**Table 15.1** Riemannian ingredients

	minimize $_{\mathbf{X} \in \mathcal{M}_r} f(\mathbf{X})$ with $\mathbf{X} = \mathbf{U}\mathbf{V}^T$
Matrix representation	$\mathbf{X} = (\mathbf{U}, \mathbf{V})$
Total space $\mathcal{M}$	$\mathbb{R}^{K \times r} \times \mathbb{R}^{K \times r}$
Group action	$(\mathbf{U}\mathbf{M}^{-1}, \mathbf{V}\mathbf{M}^T), \mathbf{M} \in \text{GL}(r)$
Quotient space $\mathcal{M}/\sim$	$\mathbb{R}^{K \times r} \times \mathbb{R}^{K \times r} / \text{GL}(r)$
Vectors in the ambient space	$(\mathbf{Z}_U, \mathbf{Z}_V) \in \mathbb{R}^{K \times r} \times \mathbb{R}^{K \times r}$
Matrix representation of a tangent vector $\xi_{\mathbf{X}} \in T_{\mathbf{X}}\mathcal{M}$	$(\xi_U, \xi_V) \in \mathbb{R}^{K \times r} \times \mathbb{R}^{K \times r}$
Riemannian Metric for $\xi_{\mathbf{X}}, \eta_{\mathbf{X}} \in T_{\mathbf{X}}\mathcal{M}$	$g_{\mathbf{X}}(\xi_{\mathbf{X}}, \eta_{\mathbf{X}}) = \text{Tr}((\mathbf{V}^T \mathbf{V}) \xi_U^T \eta_U) + \text{Tr}((\mathbf{U}^T \mathbf{U}) \xi_V^T \eta_V)$
Vertical space $\mathcal{V}_{\mathbf{X}}$	$\{(-\mathbf{U}\Lambda, \mathbf{V}\Lambda^T) : \Lambda \in \mathbb{R}^{r \times r}\}$ , where $\Lambda$ is given by (15.32)
Horizontal space $\mathcal{H}_{\mathbf{X}}$	$\{(\zeta_U, \zeta_V) \in \mathbb{R}^{K \times r} \times \mathbb{R}^{K \times r} : \mathbf{U}^T \zeta_U \mathbf{V}^T \mathbf{V} = \mathbf{U}^T \mathbf{U} \zeta_V^T \mathbf{V}\}$
Projection of a vector $\eta_{\mathbf{X}} \in T_{\mathbf{X}}\mathcal{M}$ onto the horizontal space $\mathcal{H}_{\mathbf{X}}$	$\Pi_{\mathbf{X}}(\eta_{\mathbf{X}}) = (\eta_U + \mathbf{U}\Lambda, \eta_V - \mathbf{V}\Lambda^T)$ , where $\Lambda$ is given by (15.32)
Retraction of a vector $\xi_{\mathbf{X}}$ in the horizontal space onto the manifold	$R_{\mathbf{X}}(\xi_{\mathbf{X}}) = (\mathbf{U} + \xi_U, \mathbf{V} + \xi_V)$
Matrix representation of the Riemannian gradient $\text{grad}_{\mathbf{X}} f$	$\left(\frac{\partial f}{\partial \mathbf{U}} (\mathbf{V}^T \mathbf{V})^{-1}, \frac{\partial f}{\partial \mathbf{V}} (\mathbf{U}^T \mathbf{U})^{-1}\right)$ , where $\partial f / \partial \mathbf{U}$ and $\partial f / \partial \mathbf{V}$ are the partial derivatives of $f$ with respect to $\mathbf{U}$ and $\mathbf{V}$ , respectively
Matrix representation of the Riemannian Hessian $\text{Hess}_{\mathbf{X}} f[\xi_{\mathbf{X}}]$ along a vector $\xi_{\mathbf{X}}$ in the horizontal space	$\Pi_{\mathbf{X}}(\nabla_{\xi_{\mathbf{X}}} \text{grad}_{\mathbf{X}} f)$ , where $\text{grad}_{\mathbf{X}} f$ has the representation shown above, the Riemannian connection $\nabla_{\xi_{\mathbf{X}}} \eta_{\mathbf{X}}$ is given in (15.34), and the projection operator $\Pi_{\mathbf{X}}$ is defined in (15.31)
Matrix representation of the vector transport of $\xi_{\mathbf{X}}$ at the horizontal tangent space of $\mathbf{X}$ to the tangent space of $\mathbf{Y}$	$\Pi_{\mathbf{Y}}(\xi_{\mathbf{X}})$ , where the projection operator $\Pi_{\mathbf{Y}}$ is defined in (15.31)

### 15.4.3 Riemannian Optimization Algorithms

Based on the matrix representations of the Riemannian ingredients on abstract search space  $\mathcal{M}_r / \sim$ , we shall implement Riemannian optimization algorithms in the computation space  $\mathcal{M}_r$ . To trade off the computational complexity and the convergence rate, we present a first-order Riemannian conjugate gradient algorithm and a second-order Riemannian trust-region algorithm in this subsection.

#### Riemannian Conjugate Gradient Method

The Riemannian gradient descent algorithm takes the search direction as the negative Riemannian gradient and the step size, which is determined by backtracking the line search method following the Armijo rule [25, 4.6.3]. The search direction for the Riemannian conjugate gradient method at iteration  $i$  can be expressed as  $\Xi_i := -\text{grad}_{\mathbf{X}_i} f + \beta_i \mathcal{T}_{\mathbf{X}_{i-1} \rightarrow \mathbf{X}_i}(\Xi_{i-1})$ , where  $\text{grad}_{\mathbf{X}_i} f \in \mathcal{H}_{\mathbf{X}}$  is the Riemannian gradient at point  $\mathbf{X}_i \in \mathcal{M}_r$ . The parameter  $\beta_i$  can be chosen following the generalized version of Hestenes-Stiefel [43], which is given by

$$\beta_i = \frac{g_{\mathbf{X}_i}(\text{grad}_{\mathbf{X}_i} f, \text{grad}_{\mathbf{X}_i} f - \mathcal{T}_{\mathbf{X}_{i-1} \rightarrow \mathbf{X}_i}(\text{grad}_{\mathbf{X}_{i-1}} f))}{g_{\mathbf{Y}_k}(\Xi_{i-1}, \text{grad}_{\mathbf{X}_i} f - \mathcal{T}_{\mathbf{X}_{i-1} \rightarrow \mathbf{X}_i}(\text{grad}_{\mathbf{X}_{i-1}} f))}. \quad (15.42)$$

Consequently, the sequence of the iterates for Riemannian conjugate gradient method is given by

$$\mathbf{X}_{i+1} = \mathcal{R}_{\mathbf{X}_i}(\alpha_i \mathbf{4}_i), \quad (15.43)$$

where  $\alpha_i$  denotes the step size.

### Riemannian Trust-Region Method

The Riemannian trust-region algorithm extends the trust-region method in Euclidean space [44, chapter 4] to a Riemannian quotient manifold with superlinear rate convergence, whose global convergence have been proved in [25, chapter 7]. It is achieved by iteratively solving the *trust-region subproblem* on  $\mathcal{M}/\sim$  at each iteration. In the trust-region subproblem, we shall minimize the *locally quadratic* approximation of the objective function  $f : \mathcal{M} \rightarrow \mathbb{R}$  at  $\mathbf{X} \in \mathcal{M}$ ,

$$\begin{aligned} & \underset{\xi_{\mathbf{X}} \in \mathcal{H}_{\mathbf{X}}}{\text{minimize}} \quad g_{\mathbf{X}}(\xi_{\mathbf{X}}, \text{grad}_{\mathbf{X}} f) + \frac{1}{2} g_{\mathbf{X}}(\xi_{\mathbf{X}}, \text{Hess}_{\mathbf{X}} f[\xi_{\mathbf{X}}]) \\ & \text{subject to} \quad g_{\mathbf{X}}(\xi_{\mathbf{X}}, \xi_{\mathbf{X}}) \leq \Delta^2, \end{aligned} \quad (15.44)$$

where  $\Delta$  denotes the trust-region radius.

The solution  $\xi_{\mathbf{X}}$  to the trust-region sub-problem (15.44) is a direction in the horizontal space that minimizes the quadratic approximation. By checking whether the cost function has sufficient decrease or not, we can accept or reject a potential iterate. The concrete matrix characterizations of projection operator (15.31), retraction (15.40), Riemannian gradient (15.33), and Riemannian Hessian (15.39) allow to adopt an *off-the-shelf* trust-region implementation on Riemannian manifolds with the Manopt [41] or Pymanopt [42] toolbox, which implements [25, algorithm 1] for inexactly solving the trust-region subproblem at each iteration.

#### 15.4.4 Convergence and Computational Complexity

Both the first-order Riemannian conjugate gradient algorithm and the second-order Riemannian trust-region algorithm are *globally convergent* under some mild assumptions on the objective function [23]. Starting from an arbitrary initial point, the Riemannian conjugate gradient algorithm converges to first-order KKT points, while the Riemannian trust-region algorithm converges to second-order KKT points. Theoretically, their *worst-case* global convergence rates – i.e., the required number of iterations given a fixed target accuracy – are also established.

The numerical complexity of the Riemannian optimization algorithm for solving (15.21) depends on the computational cost for (i) computing partial derivatives of the objective function  $f$  and (ii) performing manifold-related operations. The computational cost of these operations is listed below:

1. Computing the partial derivatives of the objective function  $f$  with respect to  $\mathbf{U}$  and  $\mathbf{V}$ :  $O((K + |\mathcal{S}|)r)$
2. Computing the Riemannian gradient according to formula (15.33):  $O(Kr^2 + r^3)$

3. Performing the projection operation (15.31):  $O(Kr^2 + r^3)$
4. Performing the retraction operation  $R_{\mathbf{X}}$  in (15.40):  $O(Kr)$
5. Computing the Riemannian Hessian with the formulas (15.38):  $O(Kr^2 + r^3)$

It is clear that the computational complexity of manifold-related operations are linear in  $K$  and  $|S|$ , and cubic in  $r$ .

## 15.5 Numerical Results

In this section, we conduct numerical experiments by choosing the topological interference management problem  $\mathcal{P}_{\text{TIM}}$  in partially connected  $K$ -user interference channels as a representative example.

We compare the convergence rates of the Riemannian optimization algorithms with other state-of-the-art methods. The Riemannian conjugate gradient algorithm and the Riemannian trust-region algorithm are termed “CGRP” and “TRRP,” respectively, and randomly initialized. These two algorithms are compared to the following state-of-the-art algorithms:

- LRGeom: In this algorithm, we adopt the manifold optimization algorithm for fixed-rank optimization developed in [22] to solve problem  $\mathcal{P}_{\text{TIM}}$ . This is based on embedded manifold instead of quotient manifold and termed “LRGeom.”
- LMaFit: This algorithm adopts the alternating minimization scheme with rank adaptivity to solve problem  $\mathcal{P}_{\text{TIM}}$  [45].

Consider a partially connected 100-user interference channel, and we generate 400 interference links uniformly at random. The convergence rates of different algorithms are shown in Figure 15.4(a) and Figure 15.4(b) for fixed-rank optimization problem  $\mathcal{P}_r$  (15.21) with  $r = 4$  and  $r = 5$ , respectively. We define the metric as the normalized residual – i.e.,  $\epsilon = \|\mathcal{P}_{\Omega}(\mathbf{X}) - \mathbf{I}_K\|_F / \sqrt{K}$ . Numerical results in both figures demonstrate that the Riemannian trust-region algorithm TRRP has the fastest convergence rate and the highest precision solutions in only a few iterations among other three algorithms. The Riemannian conjugate gradient CGRP achieves a faster convergence rate than LRGeom [22], benefiting from exploiting the invariance on quotient manifold. The convergence rate of alternating minimization approach LMaFit [45] is the lowest among all algorithms. We also find that the TRRP algorithm returns a solution with rank four under the stopping criterion  $\epsilon \leq 10^{-6}$  according to the two figures. Although both CGRP and LRGeom yield rank five solutions, the LRGeom algorithm achieves a slower convergence rate. In contrast, we can only obtain a solution with rank larger than five via the LMaFit algorithm under the stopping criterion  $\epsilon \leq 10^{-6}$ .

In this experiment, we show the achievable symmetric DoF of the Riemannian trust-region algorithm. Consider a partially connected 30-user interference channel. We generate the sets of connected interference links uniformly at random with probability of each pair  $(i, j) \in \mathcal{S}$  as  $p$  for  $i \neq j$ . Given  $p$  and rank parameter  $r$ , we simulate and average over 100 realizations of network topology. The phase transition behavior of

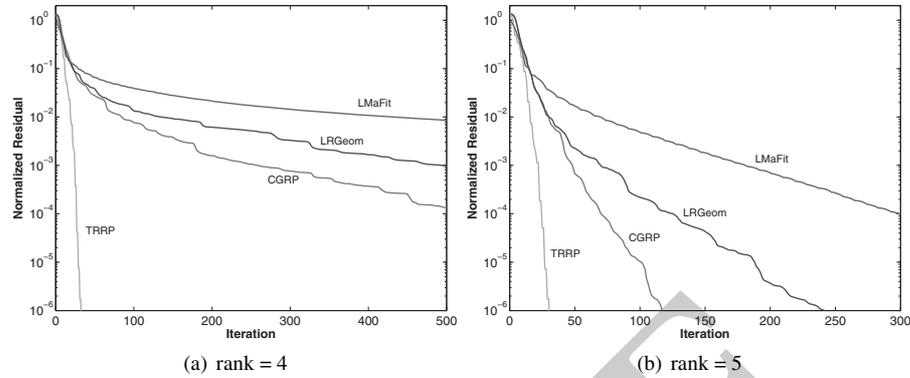


Figure 15.4 Convergence rate with the rank of matrix  $\mathbf{X}$  as four and five, respectively

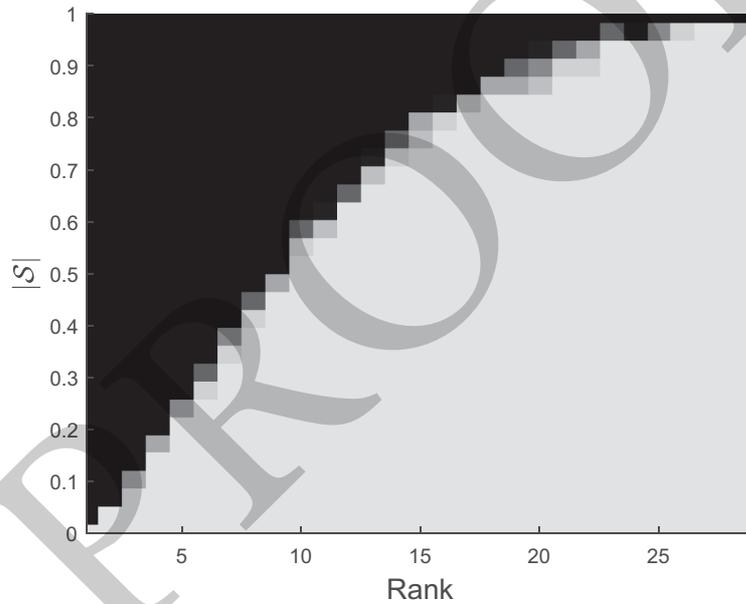


Figure 15.5 Phase transitions for the topological interference management problem for a partially connected  $K$ -user interference channel ( $K = 30$ ). The heat map indicates the empirical probability of success (dark color = 0 percent; light color = 100 percent).

the generalized low-rank optimization in topological interference management is shown in Figure 15.5. It characterizes the relationship between the achievable DoF and the number of connected interference links on average. It is observed that increasing the number of connected interference links leads to a decreasing success probability of recovering the incomplete side information modeling matrix given rank  $r$ . This result thus provides a guideline for the content placement in cache-aided interference channels and network deployment in dense wireless networks.

In summary, the presented Riemannian optimization algorithms have demonstrated their effectiveness by exploiting the quotient geometry of the fixed-rank manifold, as well as utilizing the second-order optimization method. Particularly, there exists a trade-off between the achievable symmetric DoF and the computational complexity using first-order CGRP algorithm (applicable in large-sized networks) and the second-order TRRP algorithm (applicable in small-sized and medium-sized networks).

## 15.6 Summary and Discussion

This chapter presented the generalized low-rank optimization approach for optimizing across communication, storage, and computation resources in ultra-dense Fog-RANs by exploiting side information and network structures. Illustrative application examples presented the incomplete matrix representations of modeling various types of network side information. The frameworks of developing both the first-order Riemannian conjugate gradient algorithm and second-order Riemannian trust-region algorithm on fixed-rank quotient manifold was provided. Their effectiveness for designing ultra-dense Fog-RANs has been demonstrated from the presented methodologies and numerical results.

There remain a number of interesting questions despite the encouraging progress. Heretofore, the main applications of the generalized low-rank optimization techniques are concentrated upon improving the network spectral efficiency and energy efficiency in ultra-dense Fog-RANs. The theoretical analysis for the generalized low-rank optimization models and algorithms is also an interesting topic. Significant progress has been made in the theoretical analysis of convex relaxation approaches [46] and non-convex procedures for low-rank optimization problems. But it is still difficult to extend current theoretical consequences to the generalized low-rank optimization problems  $\mathcal{P}_1$  and  $\mathcal{P}_2$  because of the complex architectures. In the end, there are various interesting research directions for improving the computational scalability of kinds of algorithms – e.g., using randomized algorithms based on sketching.

## References

- [1] Y. Shi, J. Zhang, W. Chen, and K. B. Letaief, "Generalized sparse and low-rank optimization for ultra-dense networks," *IEEE Commun. Mag.*, vol. 56, no. 6, pp. 42–48, June 2018.
- [2] Y. Yang, K. L. Wang, G. W. Zhang, X. Chen, X. Luo, and M. T. Zhou, "MEETS: Maximal energy efficient task scheduling in homogeneous fog networks," *IEEE Internet Things J.*, vol. 5, no. 5, pp. 4076–4087, Oct. 2018.
- [3] Y. Yang, S. Zhao, W. Zhang, Y. Chen, X. Luo, and J. Wang, "DEBTS: Delay energy balanced task scheduling in homogeneous fog networks," *IEEE Internet Things J.*, vol. 5, no. 3, pp. 2094–2106, June 2018.
- [4] X. Liu, Y. Shi, J. Zhang, and K. B. Letaief, "Massive CSI acquisition for dense Cloud-RANs with spatial-temporal dynamics," *IEEE Trans. Wireless Commun.*, vol. 17, no. 4, pp. 2557–2570, Apr. 2018.

- [5] T. Q. Quek, M. Peng, W. Yu, and O. Simeone, *Cloud Radio Access Networks: Principles, Technologies, and Applications*. Cambridge: Cambridge University Press, 2017.
- [6] E. Bastug, M. Bennis, and M. Debbah, “Living on the edge: The role of proactive caching in 5G wireless networks,” *IEEE Commun. Mag.*, vol. 52, no. 8, pp. 82–89, Aug. 2014.
- [7] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, “A survey on mobile edge computing: The communication perspective,” *Commun. Surveys Tuts.*, vol. 19, no. 4, pp. 2322–2358, 2017.
- [8] D. Kreutz, F. M. Ramos, P. E. Verissimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, “Software-defined networking: A comprehensive survey,” *Proc. IEEE*, vol. 103, no. 1, pp. 14–76, Jan. 2015.
- [9] S. Zhao, Y. Yang, Z. Shao, X. Yang, H. Qian, and C.-X. Wang, “Femos: Fog-enabled multitier operations scheduling in dynamic wireless networks,” *IEEE Internet Things J.*, vol. 5, no. 2, pp. 1169–1183, Apr. 2018.
- [10] M. Chiang, S. Ha, I. Chih-Lin, F. Rizzo, and T. Zhang, “Clarifying fog computing and networking: 10 questions and answers,” *IEEE Commun. Mag.*, vol. 55, no. 4, pp. 18–20, Apr. 2017.
- [11] Y. Shi, J. Zhang, K. B. Letaief, B. Bai, and W. Chen, “Large-scale convex optimization for ultra-dense Cloud-RAN,” *IEEE Wireless Commun.*, vol. 22, no. 3, pp. 84–91, June 2015.
- [12] Y. Shi, J. Zhang, and K. B. Letaief, “Group sparse beamforming for green Cloud-RAN,” *IEEE Trans. Wireless Commun.*, vol. 13, no. 5, pp. 2809–2823, May 2014.
- [13] G. Xylomenos, C. N. Ververidis, V. A. Siris, N. Fotiou, C. Tsilopoulos, X. Vasilakos, K. V. Katsaros, and G. C. Polyzos, “A survey of information-centric networking research,” *IEEE Commun. Surveys Tuts.*, vol. 16, no. 2, pp. 1024–1049, 2014.
- [14] Z. Dawy, W. Saad, A. Ghosh, J. G. Andrews, and E. Yaacoub, “Toward massive machine type cellular communications,” *IEEE Wireless Commun.*, vol. 24, no. 1, pp. 120–128, Feb. 2017.
- [15] S. A. Jafar, “Topological interference management through index coding,” *IEEE Trans. Inf. Theory*, vol. 60, no. 1, pp. 529–568, Jan. 2014.
- [16] S. Li, M. A. Maddah-Ali, and A. S. Avestimehr, “Coding for distributed fog computing,” *IEEE Commun. Mag.*, vol. 55, no. 4, pp. 34–40, Apr. 2017.
- [17] M. A. Davenport and J. Romberg, “An overview of low-rank matrix recovery from incomplete observations,” *IEEE J. Sel. Topics Signal Process.*, vol. 10, no. 4, pp. 608–622, June 2016.
- [18] M. Udell, C. Horn, R. Zadeh, S. Boyd *et al.*, “Generalized low rank models,” *Found. Trends Mach. Learn.*, vol. 9, no. 1, pp. 1–118, 2016.
- [19] Y. Chen and Y. Chi, “Harnessing structures in big data via guaranteed low-rank matrix estimation,” *IEEE Signal Process. Mag.*, vol. 35, no. 4, pp. 14–31, July 2018.
- [20] D. Amelunxen, M. Lotz, M. B. McCoy, and J. A. Tropp, “Living on the edge: Phase transitions in convex programs with random data,” *Inf. Inference*, vol. 3, no. 3, pp. 224–294, June 2014.
- [21] Y. Shi, J. Zhang, and K. B. Letaief, “Low-rank matrix completion for topological interference management by Riemannian pursuit,” *IEEE Trans. Wireless Commun.*, vol. 15, no. 7, pp. 4703–4717, July 2016.
- [22] B. Vandereycken, “Low-rank matrix completion by Riemannian optimization,” *SIAM J. Optim.*, vol. 23, no. 2, pp. 1214–1236, June 2013.
- [23] N. Boumal, P.-A. Absil, and C. Cartis, “Global rates of convergence for nonconvex optimization on manifolds,” *IMA J. Numerical Anal.*, vol. 39, no. 1, pp. 1–33, 2018.

- [24] B. Mishra and R. Sepulchre, "Riemannian preconditioning," *SIAM J. Optim.*, vol. 26, no. 1, pp. 635–660, 2016.
- [25] P.-A. Absil, R. Mahony, and R. Sepulchre, *Optimization algorithms on matrix manifolds*. Princeton, NJ: Princeton University Press, 2009.
- [26] J. Sun, Q. Qu, and J. Wright, "When are nonconvex problems not scary?" *NIPS Workshop on Non-convex Optimization for Machine Learning: Theory and Practice*, 2015.
- [27] N. Boumal, V. Voroninski, and A. Bandeira, "The non-convex Burer-Monteiro approach works on smooth semidefinite programs," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2016, pp. 2757–2765.
- [28] K. Yang, Y. Shi, and Z. Ding, "Low-rank matrix completion for mobile edge caching in Fog-RAN via Riemannian optimization," in *Proc. IEEE Global Communications Conference (GLOBECOM)*, Dec. 2016, pp. 1–6.
- [29] —, "Data shuffling in wireless distributed computing via low-rank optimization," *IEEE Trans. Signal Process.*, Apr. 2019.
- [30] T. M. Cover and J. A. Thomas, *Elements of information theory*. John Wiley & Sons, 2012.
- [31] V. R. Cadambe and S. A. Jafar, "Interference alignment and degrees of freedom of the  $K$ -user interference channel," *IEEE Trans. Inf. Theory*, vol. 54, no. 8, pp. 3425–3441, Aug. 2008.
- [32] G. Bresler, D. Cartwright, and D. Tse, "Feasibility of interference alignment for the MIMO interference channel," *IEEE Trans. Inf. Theory*, vol. 60, no. 9, pp. 5573–5586, Sept. 2014.
- [33] N. Naderializadeh, M. A. Maddah-Ali, and A. S. Avestimehr, "Fundamental limits of cache-aided interference management," *IEEE Trans. Inf. Theory*, vol. 63, no. 5, pp. 3092–3107, May 2017.
- [34] K. Yang, Y. Shi, and Z. Ding, "Generalized low-rank optimization for topological cooperation in ultra-dense networks," *IEEE Trans. Wireless Commun.*, Mar. 2019.
- [35] K. Yang, Y. Shi, J. Zhang, Z. Ding, and K. B. Letaief, "A low-rank approach for interference management in dense wireless networks," in *Proc. IEEE Global Conf. Signal Inf. Process. (GlobalSIP)*, 2016, pp. 708–712.
- [36] K. Yang, Y. Shi, and Z. Ding, "Generalized matrix completion for low complexity transceiver processing in cache-aided Fog-RAN via the Burer-Monteiro approach," *Proc. IEEE Global Conf. Signal Inf. Process. (GlobalSIP)*, pp. 863–867, Nov. 2017.
- [37] Y. Shi, B. Mishra, and W. Chen, "Topological interference management with user admission control via Riemannian optimization," *IEEE Trans. Wireless Commun.*, vol. 16, no. 11, pp. 7362–7375, Nov. 2017.
- [38] J. Dong, K. Yang, and Y. Shi, "Blind demixing for low-latency communication," *IEEE Trans. Wireless Commun.*, vol. 18, no. 2, pp. 897–911, Feb. 2019.
- [39] —, "Ranking from crowdsourced pairwise comparisons via smoothed matrix manifold optimization," in *Proc. IEEE Int. Conf. Data Mining Workshops (ICDMW)*, Nov. 2017, pp. 949–956.
- [40] A. Uschmajew and B. Vandereycken, "Line-search methods and rank increase on low-rank matrix varieties," in *Proc. Int. Symp. Nonlinear Theory Appl. (NOLTA)*, 2014, pp. 52–55.
- [41] N. Boumal, B. Mishra, P.-A. Absil, and R. Sepulchre, "Manopt, a MATLAB toolbox for optimization on manifolds," *J. Mach. Learn. Res.*, vol. 15, pp. 1455–1459, Aug. 2014.
- [42] J. Townsend, N. Koep, and S. Weichwald, "Pymanopt: A Python toolbox for optimization on manifolds using automatic differentiation," *J. Mach. Learn. Res.*, vol. 17, no. 137, pp. 1–5, 2016.

- [43] M. R. Hestenes and E. Stiefel, “Methods of conjugate gradients for solving linear systems,” *J. Res. National Bureau Standards*, vol. 49, no. 1, pp. 409–435, 1952.
- [44] S. Wright and J. Nocedal, “Numerical optimization,” *Springer Science*, vol. 35, no. 67–68, p. 7, 1999.
- [45] Z. Wen, W. Yin, and Y. Zhang, “Solving a low-rank factorization model for matrix completion by a nonlinear successive over-relaxation algorithm,” *Math. Programm. Comput.*, vol. 4, no. 4, pp. 333–361, 2012.
- [46] E. J. Candès and B. Recht, “Exact matrix completion via convex optimization,” *Found. Comput. Math.*, vol. 9, no. 6, p. 717, Apr. 2009.

PROOF