Massive MIMO 中通信高效的分布式预编码设计

李勉[1,2,3]，李洋[2,3,4]，张纵辉[1,2]，史清江[2,5]

（1. 香港中文大学（深圳）理工学院，广东 深圳 518172；2. 深圳市大数据研究院，广东 深圳 518172；
3. 鹏城国家实验室，广东 深圳 518055；4. 琶洲实验室（黄埔），广东 广州 510555；
5. 同济大学软件学院，上海 200092）

摘 要：针对多 BBU 基带处理架构，提出一种通信高效的分布式预编码方案，旨在降低 BBU 间前传交互和计算复杂度。首先，提出基于 R-WMMSE 算法的分布式框架，利用最优解的子空间特性无损压缩交互数据，降低数据交互量。然后设计了 2 种基于矩阵乘法的可学习压缩模块，通过优化的计算结构和矩阵参数减少参数和计算量，并保持函数表达能力。最后，以可达速率为优化目标，将可学习模块和分布式预编码算法框架联合优化得到最终模型。所提方案可以在更低的数据交互和计算复杂度要求下，实现预编码性能的保障。

关键词：分布式预编码；数据压缩；深度学习；联合优化

中图分类号：TN929.53

文献标志码：A

DOI: 10.11959/j.issn.1000−436x.2023147

Communication-efficient distributed precoding design for Massive MIMO

LI Mian[1,2,3], LI Yang[2,3,4], ZHANG Zonghui[1,2], SHI Qingjiang[2,5]

1. School of Science and Engineering, The Chinese University of Hong Kong (Shenzhen), Shenzhen 518172, China
2. Shenzhen Research Institute of Big Data, Shenzhen 518172, China
3. Pengcheng Laboratory, Shenzhen 518055, China
4. Pazhou Laboratory (Huangpu), Guangzhou 510555, China
5. School of Software Engineering, Tongji University, Shanghai 200092, China

Abstract: A communication-efficient distributed precoding scheme was proposed for multi-baseband processing unit (BBU) baseband processing architecture, aiming to reduce fronthaul data exchange and computational complexity between BBUs. Firstly, a distributed framework based on R-WMMSE algorithm was proposed, which utilized the subspace property of the optimal solution to compress the interactive data losslessly, thereby reducing data exchange. Furthermore, two learnable compression modules based on matrix multiplication were designed, using optimized computing structures and matrix parameters to reduce the parameters and computations while maintaining function expressiveness. Finally, the learnable modules and the distributed precoding framework were jointly optimized with achievable rate as the optimization objective to obtain the final model. The proposed scheme can achieve guaranteed precoding performance under lower requirements on data interaction and computational complexity

Keywords: distributed precoding, data compression, deep learning, joint optimization

---

收稿日期：2022−12−05；**修回日期：**2023−04−21

通信作者：史清江，shiqj@tongji.edu.cn

基金项目：国家重点研发计划基金资助项目（No.2022YFA1003900）；国家自然科学基金资助项目（No.62071409, No.62231019, No.62101349）；深圳市科技基金资助项目（No.RCJC20210609104448114）；鹏城实验室重点基金资助项目（No.PCL2023AS1-2）

Foundation Items: The National Key Research and Development Program of China (No.2022YFA1003900), The National Natural Science Foundation of China (No.62071409, No.62231019, No.62101349), Shenzhen Science and Technology Program (No.RCJC20210609104448114), Major Key Project of Pengcheng Laboratory (No.PCL2023AS1-2)

0 引言

大规模多输入多输出（Massive MIMO）是5G及未来无线通信系统中的核心技术[1-2]，其核心思想是给基站配置几十乃至数百根天线，同时为几十个用户提供高质量的通信服务。大量天线的加持极大地提高了基带处理的空间分辨率，从而有效提升了通信系统的频谱效率[3]。此外，Massive MIMO可以利用终端移动的随机性、信道衰落的不相关性、不同用户间信道的近似正交性降低用户间干扰和误码率，实现多用户空分复用。基于以上特点，近年来，Massive MIMO在LTE演进、5G和6G领域被广泛讨论[4]。

Massive MIMO也给无线系统的实现带来了巨大的挑战。一方面，天线数量的增加大幅提高了基带处理任务的复杂度，这对芯片的处理性能提出了极高的要求；另一方面，Massive MIMO系统需要支持大量天线，因此需要在芯片设计中综合考虑天线数量、布局、尺寸等复杂因素。这两方面因素导致单基带处理单元（BBU）芯片系统在成本和技术难度上缺乏优势，因此无线设施供应商都转向了多BBU芯片基站系统的方案。

多BBU系统支持灵活可扩展的部署，根据基站天线数量要求调整芯片数量。将基带处理任务分配到多块芯片上进行，降低了对芯片处理性能的要求，是一种可行且经济的设计。主流的基于多BBU系统的天线阵列可以把天线数量做到192甚至更多，但是在进一步增加天线数量时会遇到数据交互，也就是前传流量带宽的瓶颈。具体而言，当多个BBU芯片联合进行基带处理时，芯片间的数据交互量随着天线数量的增加而增长，最终变得难以承载。例如，考虑一个配备256根天线、12 bit模数转换器（ADC, analog to digital converter）的基站，当带宽为80 MHz时，基站BBU的前传速率需求将达到1 Tbit/s，而这样的高数据速率已经超出了现有数据互联标准的承受能力[5-7]。

分布式基带处理系统的BBU间过高的前传流量是阻碍更大规模天线阵列发展的重要因素，是工业界在攻克512天线乃至1 024天线Massive MIMO系统的过程中必须解决的问题。除了研究更高数据交换速度的总线互联接口，另一个值得重点研究的问题是如何从算法层面降低多BBU系统的前传流量。工业界的多BBU系统通常基于“中心节点–分布式节点”的系统架构，其特点是分布式节点处理局部天线数据，中心节点融合处理全局天线数据，达到和集中式算法等效的结果，通用的优化前传流量的手段主要还是直接的数据压缩，如离散傅里叶变换（DFT）去噪、量化压缩[8]等。

如何在保证性能的前提下优化分布式预编码算法的性能是本文考虑的核心问题。学术界关于分布式预编码算法已经有一部分工作。最早的相关工作来自文献[9-10]。文献[9-10]首次提出了下行的分布式基带处理架构，并在该架构上设计了基于交替方向乘子法（ADMM, alternating direction method of multiplier）的迫零（ZF, zero forcing）[11]预编码算法。后来学术界又提出了基于坐标下降（CD, coordinate descent）[5]、维纳滤波（WF, Wiener filter）[12]、消息传递（MP, message passing）的近似ZF和最大比传输（MRT, maximal ratio transmission）的方法[13]。以上工作假定节点之间的连接速率十分受限，因此和工业界的应用仍存在一定割裂的现象，并且由于MRC和ZF预编码的性能不佳，应用潜力不大。在线性预编码算法领域，WMMSE（weighted minimum mean squared error）[14]在至今十多年来一直被视为性能上界的标准。尽管其计算复杂度很高，但是随着移动互联网对预编码算法性能要求的不断提升，WMMSE也逐渐被部署到现网中。目前，学术界还没有关于WMMSE的分布式预编码算法的工作，而前述分布式预编码工作以ZF预编码作为近似性能的上界，同场景下参考价值较低。因此在评估本文算法的性能时，将以集中式ZF、集中式WMMSE算法作为对比算法。

本文提出了一种通信高效的分布式预编码方案，其核心思想为分布式算法框架与可学习数据压缩模块的有机结合。该方案的基础是一种基于WMMSE预编码的分布式变体，被称为分布式R-WMMSE[15]算法。通过向该算法框架中引入可学习模块并进行联合优化，保证了预编码的性能并实现了前传交互的优化。所提方案对可学习压缩模块采用极简的设计，实现了预编码性能和前传交互之间的良好折中。仿真表明，相对于经典的WMMSE算法，本文所提算法在保证预编码性能的前提下，大大降低了前传流量带宽。

1 系统模型

1.1 预编码问题

本节首先介绍Massive MIMO中预编码问题的

数学建模。考虑一个 M 根天线的基站向 K 个用户发送信号，其中，用户 k 的数据流数为 D_k，总流数为 $D=\sum_k D_k$；天线数为 N_k，总天线数为 $N=\sum_k N_k$。用户 k 和基站间的信道矩阵为 $\boldsymbol{H}_k \in \mathbb{C}^{N_k \times M}$，用户 k 的接收信号 $\boldsymbol{y}_k \in \mathbb{C}^{N_k \times 1}$ 可以表示为

$$\boldsymbol{y}_k = \underbrace{\boldsymbol{H}_k \boldsymbol{P}_k \boldsymbol{s}_k}_{\text{用户}k\text{的目标信号}} + \underbrace{\sum_{j=1, j\neq k}^{K} \boldsymbol{H}_k \boldsymbol{P}_j \boldsymbol{s}_j}_{\text{多用户信号干扰}} + \boldsymbol{n}_k \tag{1}$$

其中，$\boldsymbol{s}_k \in \mathbb{C}^{D_k \times 1}$ 表示用户 k 的数据，满足 $\mathbb{E}\left(\boldsymbol{s}_k \boldsymbol{s}_k^{\mathrm{H}}\right)=\boldsymbol{I}$；$\boldsymbol{P}_k \in \mathbb{C}^{M \times D_k}$ 表示用户 k 的预编码矩阵；$\boldsymbol{n}_k \in \mathbb{C}^{N_k \times 1}$ 表示用户 k 处的加性白高斯噪声（AWGN），满足 $\boldsymbol{n}_k \sim \mathrm{CN}\left(0, \sigma_k^2 \boldsymbol{I}\right)$，$\sigma_k^2$ 表示用户 k 天线处的噪声功率。

基站端根据下行信道信息求解不同用户的预编码矩阵。数学上，以最大化加权和速率（WSRM, weighted sum rate maximization）为目标，该问题可以表示为

$$\begin{aligned} \max_{\{\boldsymbol{P}_k\}} \quad & \sum_{k=1}^{K} \alpha_k \log\det\left(\mathrm{I}+\mathrm{SINR}_k\left(\{\boldsymbol{P}_k\}_{k=1}^{K}\right)\right) \\ \text{s.t.} \quad & \sum_{k=1}^{K} \mathrm{Tr}\left(\boldsymbol{P}_k \boldsymbol{P}_k^{\mathrm{H}}\right) \leqslant P_{\max} \end{aligned} \tag{2}$$

其中，$\alpha_k \geqslant 0$ 表示用户 k 的权重，$\boldsymbol{P}_{\max}$ 表示基站的最大发射总功率。事实上，式(2)中目标函数是频谱效率的加权之和，其与带宽的积才是加权和速率。带宽在该优化问题中是常量，因此将频谱效率和可达速率作为目标函数是等效的，故本文也沿用相关工作[14,16]对该问题的称呼。

用户 k 的信干噪比（SINR, signal-to-interference-and-noise ratio）为

$$\begin{aligned} \mathrm{SINR}_k\left(\{\boldsymbol{P}_k\}_{k=1}^{K}\right) = & \boldsymbol{P}_k^{\mathrm{H}} \boldsymbol{H}_k^{\mathrm{H}} \cdot \\ & \left(\sigma_k^2 \boldsymbol{I} + \sum_{m\neq k} \boldsymbol{H}_k \boldsymbol{P}_m \boldsymbol{P}_m^{\mathrm{H}} \boldsymbol{H}_k^{\mathrm{H}}\right)^{-1} \boldsymbol{H}_k \boldsymbol{P}_k \end{aligned} \tag{3}$$

Massive MIMO 的一个重要优势是当基站天线数 M 大于用户天线数 N 时，随着 M 的增加，线性预编码的频谱效率可以逐渐接近理想的频谱效率[17]。反之，当 $M \approx N$ 时，信道线性自相关程度会增加，导致频谱效率降低。

在实际应用中，正常情况下基站工作于 $M>N$ 的状态。为了实现单用户频谱效率和能耗之间的良好折中，通常采用用户调度和天线关断等手段，以维持比值 $\frac{M}{N}>1$ 在一个适当的范围内。本文的讨论也仅考虑 $M>N$ 的情形。

1.2　分布式预编码

多 BBU 系统采用星形拓扑架构执行分布式预编码。具体而言，系统将基站天线分成不同的簇，每簇天线对应一个局部的 BBU，使每个 BBU 只负责局部信号的处理。同时，一个中央 BBU 节点处理对应的全局数据。这种多 BBU 系统能够适应更加灵活的天线数量和分布式的部署，相对于单 BBU 系统，它能够降低对处理芯片性能的要求。

将基站天线划分为 C 簇，每个天线簇中的天线数为 $M_c=\frac{M}{C}$，对应地，信道矩阵和预编码矩阵可以分别划分为

$$\boldsymbol{H}=\begin{bmatrix} \boldsymbol{H}_1^1 & \boldsymbol{H}_1^2 & \cdots & \boldsymbol{H}_1^C \\ \boldsymbol{H}_2^1 & \boldsymbol{H}_2^2 & \cdots & \boldsymbol{H}_2^C \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{H}_K^1 & \boldsymbol{H}_K^2 & \cdots & \boldsymbol{H}_K^C \end{bmatrix}$$

$$\boldsymbol{P}=\begin{bmatrix} \boldsymbol{P}_1^1 & \boldsymbol{P}_2^1 & \cdots & \boldsymbol{P}_K^1 \\ \boldsymbol{P}_1^2 & \boldsymbol{P}_2^2 & \cdots & \boldsymbol{P}_K^2 \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{P}_1^C & \boldsymbol{P}_2^C & \cdots & \boldsymbol{P}_K^C \end{bmatrix} \tag{4}$$

其中，$\boldsymbol{H}_k^c \in \mathbb{C}^{N_k \times M_c}$ 表示第 c 簇天线和第 k 个用户间的信道矩阵，$\boldsymbol{P}_k^c \in \mathbb{C}^{M_c \times D_k}$ 表示相应的预编码矩阵。第 c 簇天线对应的 BBU 存储了 $\boldsymbol{H} \in \mathbb{C}^{N \times M}$ 的第 c 个列块 $\boldsymbol{H}^c \triangleq [\boldsymbol{H}_1^c, \boldsymbol{H}_2^c, \cdots, \boldsymbol{H}_K^c]$。通过 BBU 的局部计算和联合交互处理，第 c 簇天线对应的 BBU 最终计算得到预编码 $\boldsymbol{P} \in \mathbb{C}^{M \times D}\left(D=\sum_{k=1}^{K} D_k\right)$ 的第 c 行分块 $\boldsymbol{P}^c \triangleq [\boldsymbol{P}_1^c, \boldsymbol{P}_2^c, \cdots, \boldsymbol{P}_K^c]$，用于对应簇天线数据的预编码。

本文考虑如图 1 所示的分布式基带处理星形架构，其由一个中心节点和 C 个局部节点（对应 C 簇天线的 BBU）组成。这种架构广为采用，其原因是它能够很好地适应天线分簇所产生的处理流程。天线分簇自然会产生“局部数据”和对应的局部节点；高性能算法需要综合全局数据进行运算，这对应于中心节点的数据处理；而数据汇总和分发的过程则需要中心节点和局部节点之间的数据通路。

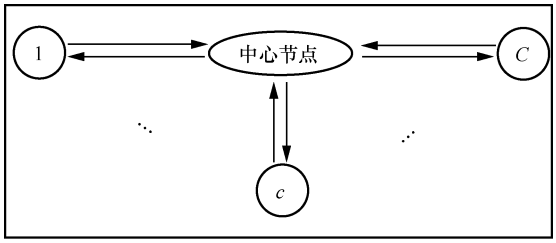

图 1 分布式基带处理星形架构

分布式预编码的前传数据交互是一个往返的过程。局部节点首先对局部信道矩阵 $\boldsymbol{H}^c$ 进行预处理和压缩，然后将压缩结果汇总到中心节点进行进一步运算；中心节点在运算完毕后，将运算结果压缩并传回各个局部节点，然后由各个局部节点计算得到其各自的预编码矩阵。

2 通信高效的分布式预编码设计

本节主要介绍所提方案的技术细节。首先简要介绍了 WMMSE 预编码算法，接着介绍了该算法的一种变体，即 R-WMMSE 分布式预编码，并将其作为本文方案所使用的优化算法框架。在学习方法部分，分别详述了可学习数据压缩模块的设计思路与分析，以及模块与算法框架的整合和联合优化的细节。分布式预编码算法框架与可学习的数据压缩模块共同构成了一个完整的分布式预编码方案。

2.1 WMMSE 预编码算法简介

WMMSE[14]是一种高性能 MIMO 线性预编码算法。其核心在于将原始的最大化加权和速率问题式(2)等价转化为

$$\begin{aligned}\min_{\{\boldsymbol{W}_k,\boldsymbol{U}_k,\boldsymbol{P}_k\}}\quad & \sum_{k=1}^{K}\alpha_k\left(\mathrm{Tr}(\boldsymbol{W}_k\boldsymbol{E}_k)-\log\det(\boldsymbol{W}_k)\right)\\ \text{s.t.}\quad & \sum_{k=1}^{K}\mathrm{Tr}\left(\boldsymbol{P}_k\boldsymbol{P}_k^{\mathrm{H}}\right)\leqslant P_{\max}\end{aligned}\tag{5}$$

其中，$\boldsymbol{W}_k$ 为新引入的辅助变量，$\boldsymbol{E}_k$ 为用户端均方误差矩阵，定义为

$$\begin{aligned}\boldsymbol{E}_k \triangleq{}& (\boldsymbol{I}-\boldsymbol{U}_k^{\mathrm{H}}\boldsymbol{H}_k\boldsymbol{P}_k)(\boldsymbol{I}-\boldsymbol{U}_k^{\mathrm{H}}\boldsymbol{H}_k\boldsymbol{P}_k)^{\mathrm{H}}+\\ &\sum_{m\neq k}\boldsymbol{U}_k^{\mathrm{H}}\boldsymbol{H}_k\boldsymbol{P}_m\boldsymbol{P}_m^{\mathrm{H}}\boldsymbol{H}_k^{\mathrm{H}}\boldsymbol{U}_k+\sigma_k^2\boldsymbol{U}_k^{\mathrm{H}}\boldsymbol{U}_k\end{aligned}\tag{6}$$

其中，$\boldsymbol{U}_k$ 为用户端接收合并矩阵。

通过对问题式(5)采用块坐标下降（BCD, block coordinate descent）法，可以得到经典的 WMMSE 算法。每次迭代依次更新 $\boldsymbol{U}_k$、$\boldsymbol{W}_k$、$\boldsymbol{P}_k$

$$\boldsymbol{U}_k=\left(\sigma_k^2\boldsymbol{I}+\sum_{m=1}^{K}\boldsymbol{H}_k\boldsymbol{P}_m\boldsymbol{P}_m^{\mathrm{H}}\boldsymbol{H}_k^{\mathrm{H}}\right)^{-1}\boldsymbol{H}_k\boldsymbol{P}_k\tag{7}$$

$$\boldsymbol{W}_k=\left(\boldsymbol{I}-\boldsymbol{U}_k^{\mathrm{H}}\boldsymbol{H}_k\boldsymbol{P}_k\right)^{-1}\tag{8}$$

$$\boldsymbol{P}_k=\left(\mu_k\boldsymbol{I}+\sum_{m=1}^{K}\alpha_m\boldsymbol{H}_m^{\mathrm{H}}\boldsymbol{U}_m\boldsymbol{W}_m\boldsymbol{U}_m^{\mathrm{H}}\boldsymbol{H}_m\right)^{-1}\alpha_k\boldsymbol{H}_k^{\mathrm{H}}\boldsymbol{U}_k\boldsymbol{W}_k\tag{9}$$

对 $\boldsymbol{P}_k$ 的子问题求解涉及能量约束，因此需要优化对偶变量 μ_k。预编码矩阵的能量是关于 μ_k 的单调函数，所以在优化 μ_k 时需要使用二分法[14]。WMMSE 预编码算法如算法 1 所示。

算法 1 WMMSE 预编码算法

输入 $\boldsymbol{H}_k^c,\zeta,\alpha_k,\sigma_k^2,P_{\max}$，$c=1,2,\cdots,C$，$k=1,2,\cdots,K$

输出 $\boldsymbol{P}_k^c$，$c=1,2,\cdots,C$，$k=1,2,\cdots,K$

1) 局部节点 c 向中心节点传输 $\boldsymbol{H}_1^c,\boldsymbol{H}_2^c,\cdots,\boldsymbol{H}_K^c$，$c=1,2,\cdots,C$

2) 中心节点初始化 $\boldsymbol{P}_k$ 使 $\sum_{k=1}^{K}\mathrm{Tr}\left(\boldsymbol{P}_k\boldsymbol{P}_k^{\mathrm{H}}\right)\leqslant P_{\max}$

3) 中心节点重复

4) $\quad\boldsymbol{W}_k'\leftarrow\boldsymbol{W}_k$，$k=1,2,\cdots,K$

5) $\quad$按照式(7)～式(9)依次更新 $\boldsymbol{U}_k$、$\boldsymbol{W}_k$、$\boldsymbol{P}_k$，$k=1,2,\cdots,K$

6) 直到 $\left|\sum_{k=1}^{K}\alpha_k(\log\det(\boldsymbol{W}_k)-\log\det(\boldsymbol{W}_k'))\right|\leqslant\zeta$

7) 中心节点将 $\boldsymbol{P}_1^c,\boldsymbol{P}_2^c,\cdots,\boldsymbol{P}_K^c$ 传输至局部节点 c，$c=1,2,\cdots,C$

WMMSE 预编码的数据交互量分析如下。在算法 1 的第 1 行，每个局部节点 c 需要向中心节点传输 $\boldsymbol{H}_1^c,\boldsymbol{H}_2^c,\cdots,\boldsymbol{H}_K^c$，共传输 MN 个复数；在第 7 行，中心节点向局部节点 c 分发预编码矩阵 $\boldsymbol{P}_1^c,\boldsymbol{P}_2^c,\cdots,\boldsymbol{P}_K^c$，共传输 MD 个复数。所以，总的数据交互量为 $M(N+D)$个复数。当基站天线数量 M 巨大时，产生极大的前传流量，这样的设计直接阻碍了更大的 Massive MIMO 天线阵的发展。

2.2 R-WMMSE 分布式预编码算法

本文的分布式预编码方案使用一种 WMMSE 算法的分布式变体（称为 R-WMMSE）作为算法框架，可提供较好的可解释性。利用优化问题中最优解的子空间特性，R-WMMSE 分布式预编码将 BBU 间的交互数据压缩到相应的低维子空间，从而有效地降低了数据交互量。需要强调的是，在预编码性能上，R-WMMSE 预编码和 WMMSE 预编码具备相同的性能。

在对 R-WMMSE 分布式预编码算法进行推导前，先介绍引理 1。

引理 1　对于最大化和速率问题式(2)，任意一个最优解 $\{\boldsymbol{P}_k\}_{k=1}^K$ 一定满足能量约束的等号条件，即 $\sum_k \mathrm{Tr}((\boldsymbol{P}_k)^{\mathrm{H}}\boldsymbol{P}_k)=P_{\max}$。

证明　使用反证法证明引理 1。

假定 $\{\boldsymbol{P}_k\}_{k=1}^K, k=1,2,\cdots,K$ 是一个最优解，并且不满足能量约束中的等号，即

$$\sum_k \mathrm{Tr}\left(\boldsymbol{P}_k\boldsymbol{P}_k^{\mathrm{H}}\right)<P_{\max} \tag{10}$$

记 $\beta=\sqrt{\dfrac{P_{\max}}{\sum_k \mathrm{Tr}\left(\boldsymbol{P}_k\boldsymbol{P}_k^{\mathrm{H}}\right)}}>1$，可构造一个可行解 $\tilde{\boldsymbol{P}}_k=\beta\boldsymbol{P}_k,\ k=1,2,\cdots,K$，满足能量约束中的等号条件

$$\sum_k \mathrm{Tr}\left(\widetilde{\boldsymbol{P}}_k\widetilde{\boldsymbol{P}}_k^{\mathrm{H}}\right)=\beta^2\sum_k \mathrm{Tr}\left(\boldsymbol{P}_k\boldsymbol{P}_k^{\mathrm{H}}\right)=P_{\max} \tag{11}$$

下面证明新构造的可行解具有更优的性能（目标函数值）。这样的结论基于式(12)的正定性

$$\begin{aligned}&\mathrm{SINR}_k\left(\{\tilde{\boldsymbol{P}}_k\}_{k=1}^K\right)-\mathrm{SINR}_k\left(\{\boldsymbol{P}_k\}_{k=1}^K\right)=\\&\boldsymbol{P}_k^{\mathrm{H}}\boldsymbol{H}_k^{\mathrm{H}}\left(\frac{\sigma_k^2}{\beta^2}\boldsymbol{I}+\sum_{m\neq k}\boldsymbol{H}_k\boldsymbol{P}_m\boldsymbol{P}_m^{\mathrm{H}}\boldsymbol{H}_k^{\mathrm{H}}\right)^{-1}\boldsymbol{H}_k\boldsymbol{P}_k-\\&\boldsymbol{P}_k^{\mathrm{H}}\boldsymbol{H}_k^{\mathrm{H}}\left(\sigma_k^2\boldsymbol{I}+\sum_{m\neq k}\boldsymbol{H}_k\boldsymbol{P}_m\boldsymbol{P}_m^{\mathrm{H}}\boldsymbol{H}_k^{\mathrm{H}}\right)^{-1}\boldsymbol{H}_k\boldsymbol{P}_k=\\&\boldsymbol{P}_k^{\mathrm{H}}\boldsymbol{H}_k^{\mathrm{H}}\left(\left(\frac{\sigma_k^2}{\beta^2}\boldsymbol{I}+\sum_{m\neq k}\boldsymbol{H}_k\boldsymbol{P}_m\boldsymbol{P}_m^{\mathrm{H}}\boldsymbol{H}_k^{\mathrm{H}}\right)^{-1}-\right.\\&\left.\left(\sigma_k^2\boldsymbol{I}+\sum_{m\neq k}\boldsymbol{H}_k\boldsymbol{P}_m\boldsymbol{P}_m^{\mathrm{H}}\boldsymbol{H}_k^{\mathrm{H}}\right)^{-1}\right)\boldsymbol{H}_k\boldsymbol{P}_k\succ 0,k=1,2,\cdots,K\end{aligned} \tag{12}$$

记 $\lambda_i(\boldsymbol{X})$ 为矩阵 $\boldsymbol{X}$ 的第 i 大特征值，那么有

$$\begin{aligned}&\lambda_i\left(\boldsymbol{I}+\mathrm{SINR}_k\left(\{\tilde{\boldsymbol{P}}_k\}_{k=1}^K\right)\right)>\\&\lambda_i\left(\boldsymbol{I}+\mathrm{SINR}_k\left(\{\boldsymbol{P}_k\}_{k=1}^K\right)\right),k=1,2,\cdots,K\end{aligned} \tag{13}$$

从而有

$$\begin{aligned}&\sum_{k=1}^K\alpha_k\log\det\left(\boldsymbol{I}+\mathrm{SINR}_k\left(\{\tilde{\boldsymbol{P}}_k\}_{k=1}^K\right)\right)>\\&\sum_{k=1}^K\alpha_k\log\det\left(\boldsymbol{I}+\mathrm{SINR}_k\left(\{\boldsymbol{P}_k\}_{k=1}^K\right)\right)\end{aligned} \tag{14}$$

也就是说，假定一个最优解 $\{\boldsymbol{P}_k\}_{k=1}^K$ 不满足能量约束中的等号，那么就能够找到更优的解 $\{\tilde{\boldsymbol{P}}_k\}_{k=1}^K$，这与 $\{\boldsymbol{P}_k\}_{k=1}^K$ 的最优性矛盾。以上推导证明了问题式(2)的最优解一定满足能量约束中的等号。证毕。

基于引理 1，可以证明定理 1[15]。

定理 1　对于最大化和速率问题式(2)，任意一个最优的预编码矩阵可以表示为 $\boldsymbol{P}_k=\boldsymbol{H}^{\mathrm{H}}\boldsymbol{X}_k$，$\boldsymbol{X}_k\in\mathbb{C}^{N\times D_k}, k=1,2,\cdots,K$。也就是说，$\{\boldsymbol{P}_k\}_{k=1}^K$ 是最优解的必要条件是它在 $\boldsymbol{H}^{\mathrm{H}}$ 的列空间 $\mathrm{R}(\boldsymbol{H}^{\mathrm{H}})$ 中。

证明　使用反证法证明定理 1。

假定问题式(2)存在最优解 $\{\boldsymbol{P}_k\}_{k=1}^K$ 并且此解不在 $\boldsymbol{H}^{\mathrm{H}}$ 的列空间 $\mathrm{R}(\boldsymbol{H}^{\mathrm{H}})$ 中。注意到，$\boldsymbol{H}^{\mathrm{H}}$ 的列空间 $\mathrm{R}(\boldsymbol{H}^{\mathrm{H}})$ 和 $\boldsymbol{H}$ 的零空间 $\mathrm{N}(\boldsymbol{H})$ 是一对正交空间，本文可以对 $\{\boldsymbol{P}_k\}_{k=1}^K$ 的列进行投影，形成分别位于这一对正交子空间的两部分：$\boldsymbol{A}_k=\prod_{\mathrm{R}(\boldsymbol{H}^{\mathrm{H}})}\boldsymbol{P}_k$ 和 $\boldsymbol{B}_k=\prod_{\mathrm{N}(\boldsymbol{H})}\boldsymbol{P}_k=\boldsymbol{P}_k-\boldsymbol{A}_k$。

将 $\boldsymbol{P}_k=\boldsymbol{A}_k+\boldsymbol{B}_k, k=1,2,\cdots,K$ 代入问题式(2)的目标函数中，有

$$\begin{aligned}&\sum_{k=1}^K\alpha_k\log\det\left(\boldsymbol{I}+\mathrm{SINR}_k\left((\boldsymbol{P}_k)_{k=1}^K\right)\right)=\\&\sum_{k=1}^K\alpha_k\log\det\left(\boldsymbol{I}+\boldsymbol{H}_k\boldsymbol{P}_k\boldsymbol{P}_k^{\mathrm{H}}\boldsymbol{H}_k^{\mathrm{H}}\cdot\right.\\&\left.\left(\sum_{m\neq k}\boldsymbol{H}_k\boldsymbol{P}_m\boldsymbol{P}_m^{\mathrm{H}}\boldsymbol{H}_k^{\mathrm{H}}+\sigma_k^2\boldsymbol{I}\right)^{-1}\right)=\\&\sum_{k=1}^K\alpha_k\log\det\left(\boldsymbol{I}+\boldsymbol{H}_k\boldsymbol{A}_k\boldsymbol{A}_k^{\mathrm{H}}\boldsymbol{H}_k^{\mathrm{H}}\cdot\right.\\&\left.\left(\sum_{m\neq k}\boldsymbol{H}_k\boldsymbol{A}_m\boldsymbol{A}_m^{\mathrm{H}}\boldsymbol{H}_k^{\mathrm{H}}+\sigma_k^2\boldsymbol{I}\right)^{-1}\right)\end{aligned} \tag{15}$$

其中，最后一个等号成立是因为 $\boldsymbol{H}^{\mathrm{H}}\boldsymbol{B}_k=0$。注意到，预编码矩阵中仅有 $\mathrm{R}(\boldsymbol{H}^{\mathrm{H}})$ 中的成分 $\{\boldsymbol{A}_k\}_{k=1}^K$ 对目标函数值有贡献。这个等式关系说明了 $\{\boldsymbol{A}_k\}_{k=1}^K$ 也是一个最优解。

因为 $\boldsymbol{P}_k$ 不在 $\mathrm{R}(\boldsymbol{H}^{\mathrm{H}})$ 中，所以 $\boldsymbol{P}_k$ 在 $\mathrm{N}(\boldsymbol{H})$ 的投影 $\boldsymbol{B}_k$ 一定不为零，也就有 $\mathrm{Tr}(\boldsymbol{B}_k(\boldsymbol{B}_k)^{\mathrm{H}})>0, k=1,2,\cdots,K$。据此得到能量约束中的不等号被 $\boldsymbol{A}_k$ 满足，也就是

$$\begin{aligned}&P_{\max}\geqslant\sum_k\mathrm{Tr}\left(\boldsymbol{P}_k(\boldsymbol{P}_k)^{\mathrm{H}}\right)=\\&\sum_k\mathrm{Tr}\left((\boldsymbol{A}_k)^{\mathrm{H}}\boldsymbol{A}_k\right)+\sum_k\mathrm{Tr}\left((\boldsymbol{B}_k)^{\mathrm{H}}\boldsymbol{B}_k\right)>\\&\sum_k\mathrm{Tr}\left((\boldsymbol{A}_k)^{\mathrm{H}}\boldsymbol{A}_k\right)\end{aligned} \tag{16}$$

以上推理说明对于每一个不在子空间 $\mathrm{R}(\boldsymbol{H}^{\mathrm{H}})$ 的最优解 $\{\boldsymbol{P}_k\}_{k=1}^K$，可以通过投影构造另一个最优解 $\{\boldsymbol{A}_k\}_{k=1}^K$，并且这个最优解不满足能量约束中的等号，与引理 1 的结论矛盾。综上，本文证明了问题式(2)的任意一个最优解都在 $\boldsymbol{H}^{\mathrm{H}}$ 的列空间中，也就是问题式(2)的所有最优解都可以表示为 $\boldsymbol{P}_k=\boldsymbol{H}^{\mathrm{H}}\boldsymbol{X}_k,\boldsymbol{X}_k\in\mathbb{C}^{N\times D_k},k=1,2,\cdots,K$。证毕。

根据定理 1，可以把问题式(5)的优化空间限制在 $\mathrm{R}(\boldsymbol{H}^{\mathrm{H}})$ 内，从而将求解变量 $\boldsymbol{P}_k\in\mathbb{C}^{M\times D_k}$ 替换为更低维度的 $\boldsymbol{X}_k\in\mathbb{C}^{N\times D_k}$，并且将能量约束归并到目标函数中。记 $\boldsymbol{M}_k\triangleq\boldsymbol{U}_k\boldsymbol{W}_k\boldsymbol{U}_k^{\mathrm{H}}\in\mathbb{C}^{N_k\times N_k}$，$\overline{\boldsymbol{H}}\triangleq\boldsymbol{H}\boldsymbol{H}^{\mathrm{H}}\in\mathbb{C}^{N\times N}$，$\overline{\boldsymbol{H}}_k\triangleq\boldsymbol{H}_k\boldsymbol{H}^{\mathrm{H}}\in\mathbb{C}^{N_k\times N}$，那么类似于式(7)～式(9)，问题式(5)关于 $\boldsymbol{X}_k$ 的解可通过循环更新 $\boldsymbol{U}_k$、$\boldsymbol{W}_k$、$\boldsymbol{X}_k$ 得到

$$\boldsymbol{U}_k=\left(\sum_{m=1}^K\frac{\sigma_k^2}{P_{\max}}\mathrm{Tr}\left(\overline{\boldsymbol{H}}\boldsymbol{X}_m\boldsymbol{X}_m^{\mathrm{H}}\right)\boldsymbol{I}+\sum_{m=1}^K\overline{\boldsymbol{H}}_k\boldsymbol{X}_m\boldsymbol{X}_m^{\mathrm{H}}\overline{\boldsymbol{H}}_k^{\mathrm{H}}\right)\overline{\boldsymbol{H}}_k\boldsymbol{X}_k \tag{17}$$

$$\boldsymbol{W}_k=\left(\boldsymbol{I}-\boldsymbol{U}_k^{\mathrm{H}}\overline{\boldsymbol{H}}_k\boldsymbol{X}_k\right)^{-1} \tag{18}$$

$$\boldsymbol{X}_k=\left(\sum_{m=1}^K\frac{\sigma_m^2}{P_{\max}}\alpha_m\,\mathrm{Tr}\left(\boldsymbol{M}_m\right)\overline{\boldsymbol{H}}+\sum_{m=1}^K\alpha_m\overline{\boldsymbol{H}}_m^{\mathrm{H}}\boldsymbol{M}_m\overline{\boldsymbol{H}}_m\right)\alpha_k\overline{\boldsymbol{H}}_k^{\mathrm{H}}\boldsymbol{U}_k\boldsymbol{W}_k \tag{19}$$

R-WMMSE 分布式预编码算法执行流程如算法 2 所示。

算法 2 R-WMMSE 分布式预编码算法

输入 $\boldsymbol{H}_k^c,\zeta,\alpha_k,\sigma_k^2,P_{\max},c=1,2,\cdots,C,\ k=1,2,\cdots,K$

输出 $\boldsymbol{P}_k^c,\ c=1,2,\cdots,C,\ k=1,2,\cdots,K$

1) 局部节点 c 计算并传输 $\boldsymbol{H}^c(\boldsymbol{H}^c)^{\mathrm{H}}$ 至中心节点，$c=1,2,\cdots,C$

2) 中心节点计算 $\overline{\boldsymbol{H}}=\sum_{c=1}^C\boldsymbol{H}^c(\boldsymbol{H}^c)^{\mathrm{H}}$

3) 中心节点初始化 $\{\boldsymbol{X}_k\}$ 使 $\sum_{k=1}^K\mathrm{Tr}\left(\overline{\boldsymbol{H}}\boldsymbol{X}_k\boldsymbol{X}_k^{\mathrm{H}}\right)\leqslant P_{\max}$

4) 重复

5) $\boldsymbol{W}_k'\leftarrow\boldsymbol{W}_k,\ k=1,2,\cdots,K$

6) 按照式(17)～式(19)依次更新 $\boldsymbol{U}_k$、$\boldsymbol{W}_k$、$\boldsymbol{X}_k$

7) 直到 $\left|\sum_{k=1}^K\alpha_k(\log\det(\boldsymbol{W}_k)-\log\det(\boldsymbol{W}_k'))\right|\leqslant\zeta$

8) 中心节点将 $\boldsymbol{X}_1^c,\boldsymbol{X}_2^c,\cdots,\boldsymbol{X}_K^c$ 传输到局部节点 c，$c=1,2,\cdots,C$

9) 局部节点 c 计算 $\boldsymbol{P}_k^c=(\boldsymbol{H}^c)^{\mathrm{H}}\boldsymbol{X}_k,c=1,2,\cdots,C,k=1,2,\cdots,K$

R-WMMSE 预编码的数据交互分析如下。在算法 2 的第 1 行，每个局部节点 c 向中心节点传输 $\boldsymbol{H}^c(\boldsymbol{H}^c)^{\mathrm{H}}\in\mathbb{C}^{N\times N}$，共需要传输 $\frac{1}{2}CN^2$ 个复数；在第 8 行，中心节点向局部节点 c 分发 $\boldsymbol{X}_1^c,\boldsymbol{X}_2^c,\cdots,\boldsymbol{X}_K^c$，共需要传输 CND 个复数。所以，总的数据交互量为 $\frac{1}{2}CN^2+CND$ 个复数。

评估算法在实际系统中的性能表现时，需要综合考虑全频带、用户调度、算法时间分配等因素，因此本文只能给出简易的估算。下面给出一个示例，当考虑 $M=128$、$N=D=16$、$C=4$ 时，WMMSE 预编码的数据交互量为 4 096 个复数，而 R-WMMSE 的数据交互量仅为 1 536 个复数。当全频带为 80 MHz 时，按照 30 kHz 一个子载波进行切分，复数量化位数为 12 bit（6 bit 实部和 6 bit 虚部），算法执行时限定时间分配为 0.3 ms，那么 WMMSE 预编码执行过程的数据交互为 488.28 Gbit/s，R-WMMSE 预编码则为 183.11 Gbit/s。如果该基站系统最高支持 500 Gbit/s 前传带宽，那么使用 WMMSE 预编码时，系统只能驱动上面介绍的 128 天线，而使用 R-WMMSE 预编码时则能够驱动 256 天线（$M=256,C=8$）。

以上分析表明，在常规的基站规模配置下，相较于 WMMSE 算法，R-WMMSE 分布式预编码大幅优化了前传交互量。同时，示例直观展示了优化数据交互量如何帮助系统支持更大规模的天线阵列。

2.3 可学习的数据压缩模块设计

为了进一步降低算法 2 中（第 1 行和第 8 行）的数据交互量，本节给出可学习的数据压缩模块设计。所介绍的模块设计不依赖于特定预编码算法，而是能与本文提到的各种方法（如 ZF 预编码、WMMSE 预编码、R-WMMSE 预编码等）结合。本文以 R-WMMSE 分布式预编码为例展示方案的可行性。

在所提出的可学习数据压缩模块设计中，每一个

压缩模块由一个压缩函数和一个解压函数共同组成。在发送节点，针对待传输的矩阵数据 $\boldsymbol{A}\in\mathbb{C}^{m\times n}$，设计一个压缩函数 $f_{\theta_1}:\mathbb{C}^{m\times n}\to\mathbb{C}^{p\times q}$，满足 $pq<mn$，其中，$\boldsymbol{\theta}_1$ 是待学习的参数。通过执行函数 $\boldsymbol{B}=f_{\theta_1}(\boldsymbol{A})$，可将高维度的矩阵 $\boldsymbol{A}$ 压缩为较低维度的矩阵 $\boldsymbol{B}$ 后传输，因此只需要传输 $pq<mn$ 个复数。在接收节点，本文设计一个带有可学习参数 $\boldsymbol{\theta}_2$ 的解压函数 $g_{\theta_2}:\mathbb{C}^{p\times q}\to\mathbb{C}^{m\times n}$，通过执行 $\tilde{\boldsymbol{A}}=g_{\theta_2}(\boldsymbol{B})$ 进行数据的恢复。上述压缩函数和解压函数共同组成可学习的数据压缩模块 $F_{\{\theta_1,\theta_2\}}=g_{\theta_2}\circ f_{\theta_1}$，其中。表示函数复合运算。

下面分别介绍3种不同的可学习的数据压缩模块。

1)　单边压缩（SSC, single sided compression）模块

考虑一种简单的矩阵单边压缩，即

$$f_{\theta_1}(\boldsymbol{A})=\boldsymbol{P}_1\boldsymbol{A},\quad g_{\theta_2}(\boldsymbol{B})=\boldsymbol{P}_2\boldsymbol{B}+\boldsymbol{S},$$
$$\boldsymbol{P}_1\in\mathbb{C}^{p\times m},\boldsymbol{P}_2\in\mathbb{C}^{m\times p},\boldsymbol{S}\in\mathbb{C}^{m\times n}\tag{20}$$

其中，$\boldsymbol{\theta}_1$ 即 $\boldsymbol{P}_1$，$\boldsymbol{\theta}_2$ 包含 $\boldsymbol{P}_2$ 和 $\boldsymbol{S}$ 两部分，总参数量为 $mn+2mp$。由 f_{θ_1} 的表达式可以看到，SSC 压缩方式要求 $q=n,p<m$。

2)　双边压缩（DSC, double sided compression）模块

另一种压缩模块执行对矩阵的双边压缩，即

$$f_{\theta_1}(\boldsymbol{A})=\boldsymbol{P}_1\boldsymbol{A}\boldsymbol{Q}_1,\quad g_{\theta_2}(\boldsymbol{B})=\boldsymbol{P}_2\boldsymbol{B}\boldsymbol{Q}_2+\boldsymbol{S},$$
$$\boldsymbol{P}_1\in\mathbb{C}^{p\times m},\boldsymbol{P}_2\in\mathbb{C}^{m\times p},\boldsymbol{Q}_1\in\mathbb{C}^{n\times q},$$
$$\boldsymbol{Q}_2\in\mathbb{C}^{q\times n},\boldsymbol{S}\in\mathbb{C}^{m\times n}\tag{21}$$

其中，$\boldsymbol{\theta}_1$ 包含 $\boldsymbol{P}_1$和$\boldsymbol{Q}_1$ 两部分，$\boldsymbol{\theta}_2$ 包含 $\boldsymbol{P}_2$、$\boldsymbol{Q}_2$ 和 $\boldsymbol{S}$ 三部分，总参数量为 $mn+2mp+2nq$。

3)　全连接（FC, fully connected）模块

参考神经网络的全连接设计，可以直接得到如下的全连接数据压缩模块设计

$$f_{\theta_1}(\boldsymbol{A})=\boldsymbol{P}_3\mathrm{vec}(\boldsymbol{A})$$
$$g_{\theta_2}(\boldsymbol{b})=\mathrm{reshape}(\boldsymbol{P}_4\boldsymbol{b},mn)+\boldsymbol{S}$$
$$\boldsymbol{P}_3\in\mathbb{C}^{pq\times mn},\boldsymbol{P}_4\in\mathbb{C}^{mn\times pq},\boldsymbol{S}\in\mathbb{C}^{m\times n}\tag{22}$$

其中，reshape 函数和 vec 函数正好是一对互逆的映射，reshape 的第二个参数表示输出矩阵的维度，$\boldsymbol{\theta}_1$ 即 $\boldsymbol{P}_3$，$\boldsymbol{\theta}_2$ 包含 $\boldsymbol{P}_4$ 和 $\boldsymbol{S}$ 两部分，总参数量为 $mn+2mnpq$。

下面分析以上 3 种模块的输出元素关于输入元素的依赖关系。所提出的 2 种模块中 SSC 的输入输出关系根据式(20)可以表示为 $F_{\mathrm{SSC}}(\boldsymbol{A})=\boldsymbol{P}_2\boldsymbol{P}_1\boldsymbol{A}+\boldsymbol{S}$。记 $\tilde{\boldsymbol{P}}=\boldsymbol{P}_2\boldsymbol{P}_1$，可以得到如下的逐元素输入输出关系

$$[F_{\mathrm{SSC}}(\boldsymbol{A})]_{k,l}=[\boldsymbol{S}]_{k,l}+\sum_{i=1}^{m}[\tilde{\boldsymbol{P}}]_{k,i}[\boldsymbol{A}]_{i,l}\tag{23}$$

对于 DSC 模块，记 $\tilde{\boldsymbol{P}}=\boldsymbol{P}_2\boldsymbol{P}_1,\tilde{\boldsymbol{Q}}=\boldsymbol{Q}_1\boldsymbol{Q}_2$，那么有

$$[F_{\mathrm{DSC}}(\boldsymbol{A})]_{k,l}=[\boldsymbol{S}]_{k,l}+\sum_{i=1}^{m}\sum_{j=1}^{n}[\tilde{\boldsymbol{P}}]_{k,i}[\boldsymbol{A}]_{i,j}[\tilde{\boldsymbol{Q}}]_{j,l}\tag{24}$$

对于 FC 模块，记 $\tilde{\boldsymbol{P}}=\boldsymbol{P}_4\boldsymbol{P}_3$，类似地，可以得到

$$[F_{\mathrm{FC}}(\boldsymbol{A})]_{k,l}=[\boldsymbol{S}]_{k,l}+\sum_{i=1}^{m}\sum_{j=1}^{n}[\tilde{\boldsymbol{P}}]_{k+(l-1)m,i+(j-1)m}[\boldsymbol{A}]_{i,j}\tag{25}$$

对比式(23)～式(25)，有以下发现。

① SSC 模块的第 k 行第 l 列输出元素为 $\boldsymbol{A}$ 中第 l 列元素的线性组合再加上一个常数。

② DSC 模块的第 k 行第 j 列输出元素为 $\boldsymbol{A}$ 中所有元素的线性组合再加上一个常数，因此具备比 SSC 更强的输入输出关系表达能力。

③ FC 模块的第 k 行第 j 列输出元素为 $\boldsymbol{A}$ 中所有元素的线性组合再加上一个常数，且线性组合权重不共享，和 DSC 具有同水平的输入输出关系表达能力。

值得注意的是，压缩解压层次更多的单边矩阵压缩、双边矩阵压缩模块可以化简为 SSC 和 DSC 模块。例如，包含多个压缩解压矩阵的双边压缩模块

$$F(\boldsymbol{A})=\boldsymbol{P}_4'\boldsymbol{P}_3'\boldsymbol{P}_2'\boldsymbol{P}_1'\boldsymbol{A}\boldsymbol{Q}_1'\boldsymbol{Q}_2'\boldsymbol{Q}_3'\boldsymbol{Q}_4'+\boldsymbol{S}\tag{26}$$

可以化简为前文中介绍的 F_{DSC}（令 $\boldsymbol{P}_1=\boldsymbol{P}_2'\boldsymbol{P}_1'$，$\boldsymbol{P}_2=\boldsymbol{P}_4'\boldsymbol{P}_3'$，$\boldsymbol{Q}_1=\boldsymbol{Q}_1'\boldsymbol{Q}_2'$，$\boldsymbol{Q}_2=\boldsymbol{Q}_3'\boldsymbol{Q}_4'$）。因此，此类更复杂的压缩模块并不具备更强的输入输出关系表达能力，反而会引入更多的参数量和计算复杂度。所以，前文提到的 SSC、DSC 都是同形态（单边矩阵压缩和双边矩阵压缩）模块设计中的最简结构。

综合比较上述 3 种可学习压缩模块的参数量和表达能力，当 m,n,p,q 的数量级相同时，有以下结论成立。

① 复杂度方面：FC 相比 SSC 或 DSC 模块的参数量高 2 阶，对应地引入了高 2 阶的计算复杂度。

② 表达能力方面：FC 和 DSC 模块的表达能力水平相同，且都高于 SSC 模块。

本文认为，所提出的 SSC 和 DSC 模块相比 FC 模块在复杂度和性能方面都分别实现了更好的均衡，后文将用实验佐证该观点。此外，值得注意的

是，以上模块设计所引入的计算复杂度和参数存储开销的量级都不大。其中，计算复杂度和原矩阵所做的矩阵乘法相当，而参数存储开销同样和原矩阵的维度相当。

2.4 分布式算法和可学习压缩模块的联合优化

本节介绍可学习数据压缩模块和分布式算法框架进行联合优化的模型训练方法，并阐述可学习模块提升模型性能的机理。

最直接的模型优化方式是有监督学习，其直接优化 SSC、DSC 的输入输出间的差距，如优化输入输出的均方误差（MSE, mean square error）

$$\min_{\{\boldsymbol{\theta}_1,\boldsymbol{\theta}_2\}} \mathbb{E}\left\|\boldsymbol{A}-F_{\{\boldsymbol{\theta}_1,\boldsymbol{\theta}_2\}}(\boldsymbol{A})\right\|_2^2 \tag{27}$$

其中，期望 $\mathbb{E}$ 是通过对大量随机生成的样本 $\boldsymbol{A}$ 取平均近似得到的。采用梯度下降（GD, gradient descent）法优化式(27)得到可学习压缩模块的参数后，即可将其植入 R-WMMSE 分布式算法中。尽管基于式(27)的独立优化简单且直接，但是其最终得到的模型预编码性能会有较大的损失。其根本原因在于，训练后的带压缩预编码仅逼近未压缩预编码，并没有考虑到对和速率的优化。例如，本文基于 2 轮迭代的 R-WMMSE 的带压缩预编码，其性能上限是 2 轮迭代的 R-WMMSE 预编码，此时其性能与 R-WMMSE 预编码的收敛性能还有较大差距。

为了避免上述的性能损失，本文提出使用无监督学习的方案。直接以下行加权和速率为目标函数（见原问题式(2)），对可学习压缩模块和分布式预编码采用端到端的联合优化。如算法 3 所示，算法执行主要分为 3 个阶段。第一阶段为信道数据的预处理及汇总（第 1～2 行）；第二阶段为预编码的中心迭代计算（第 3～7 行）；第三阶段为预编码矩阵的分发和局部计算（第 8～9 行）。为了优化可学习压缩模块中的参数值，本文对算法 3 采用基于反向传播的梯度下降法。具体而言，首先产生一个训练集 $\Omega=\{\boldsymbol{H}^{(1)},\boldsymbol{H}^{(2)},\cdots,\boldsymbol{H}^{(S)}\}$，其中，$S$ 表示训练集的样本数。对于每个样本，执行算法 3 输出 $\boldsymbol{P}(\boldsymbol{H}^{(i)})$，其中，$i$ 表示第 i 个样本，然后以和速率为目标函数通过反向传播计算其关于压缩模块参数的梯度，从而采用 GD 法更新参数值。

算法 3 通信高效的分布式预编码算法

输入 $\boldsymbol{H}_k^c, T, \alpha_k, \sigma_k^2, P_{\max}$，$c=1,2,\cdots,C$，$k=1,2,\cdots,K$

输出 $\boldsymbol{P}_k^c$，$c=1,2,\cdots,C$，$k=1,2,\cdots,K$

1) 局部节点 c 使用压缩模块 F_H^c 压缩得到 $f_H^c(\boldsymbol{H}^c(\boldsymbol{H}^c)^{\mathrm{H}})$，$c=1,2,\cdots,C$，并传输到中心节点

2) 中心节点解压得到 $F_H^c(\boldsymbol{H}^c(\boldsymbol{H}^c)^{\mathrm{H}})$，$c=1,2,\cdots,C$，并计算压缩后的矩阵 $\overline{\boldsymbol{\mathcal{H}}}\leftarrow\sum_{c=1}^{C}F_H^c(\boldsymbol{H}^c(\boldsymbol{H}^c)^{\mathrm{H}})$

3) 中心节点初始化 $\boldsymbol{X}_k$ 使 $\sum_{k=1}^{K}\mathrm{Tr}\left(\overline{\boldsymbol{H}}\boldsymbol{X}_k\boldsymbol{X}_k^{\mathrm{H}}\right)\leqslant P_{\max}, k=1,2,\cdots,K$

4) 重复 T 次

5) 按照式(28)更新 $\boldsymbol{U}_k$

$$\boldsymbol{U}_k\leftarrow\left(\sum_{m=1}^{K}\frac{\sigma_k^2}{P_{\max}}\mathrm{Tr}\left(\overline{\boldsymbol{\mathcal{H}}}\boldsymbol{X}_m\boldsymbol{X}_m^{\mathrm{H}}\right)\boldsymbol{I}+\sum_{m=1}^{K}\overline{\boldsymbol{\mathcal{H}}}_k\boldsymbol{X}_m\boldsymbol{X}_m^{\mathrm{H}}\overline{\boldsymbol{\mathcal{H}}}_k^{\mathrm{H}}\right)^{-1}\overline{\boldsymbol{\mathcal{H}}}_k\boldsymbol{X}_k \tag{28}$$

6) $\boldsymbol{W}_k\leftarrow(\boldsymbol{I}-\boldsymbol{U}_k^H\overline{\boldsymbol{\mathcal{H}}}_k\boldsymbol{X}_k)^{-1}$

7) 按照式(29)更新 $\boldsymbol{X}_k$

$$\boldsymbol{X}_k\leftarrow\left(\sum_{m=1}^{K}\frac{\sigma_m^2}{P_{\max}}\alpha_m\mathrm{Tr}\left(\boldsymbol{M}_m\right)\overline{\boldsymbol{\mathcal{H}}}+\sum_{m=1}^{K}\alpha_m\overline{\boldsymbol{\mathcal{H}}}_m^{\mathrm{H}}\boldsymbol{M}_m\overline{\mathcal{H}}_m\right)^{-1}\alpha_k\overline{\boldsymbol{\mathcal{H}}}_k^{\mathrm{H}}\boldsymbol{U}_k\boldsymbol{W}_k \tag{29}$$

8) 中心节点使用压缩模块 F_X，将 $f_X(\boldsymbol{X})$ 传输到局部节点 c，$c=1,2,\cdots,C$ /* 其中 $\boldsymbol{X}\triangleq[\boldsymbol{X}_1\ \ \boldsymbol{X}_2\ \ \cdots\ \ \boldsymbol{X}_K]$*/

9) 局部节点 c 解压数据得到 $F_X(\boldsymbol{X})$，计算 $\boldsymbol{P}^c\leftarrow\boldsymbol{H}^cF_X(\boldsymbol{X})$ /*其中 $\boldsymbol{P}^c=[\boldsymbol{P}_1^c\ \boldsymbol{P}_2^c\cdots\boldsymbol{P}_K^c]$ */

值得注意的是，当固定迭代次数时，在特定压缩维度下，本文提出的基于无监督联合优化的算法 3 的性能可以超越同迭代次数（如 2 轮，此时优化迭代算法未收敛）的无压缩损失的 R-WMMSE 算法 2。这是因为无监督优化的算法 3 的训练目标为达到最优解，而固定迭代次数的算法 2 在相应迭代次数下尚未收敛，性能比全局最优解更差。因此算法 3 通过训练有机会得到比算法 2 性能更好的解。

为了直观理解，可以考虑一种特殊情况，即压缩模块不执行维度压缩（输入、输出和压缩维度都相等）。通过恰当的初始化，可将学习模块变成一个恒

等映射，从而在相同迭代次数下，算法 3 模型的初始性能和算法 2 相等。训练开始时，算法 3 模型性能并非最优，可学习压缩模块的参数梯度不为 0。因此，通过 GD 法更新参数，可学习模块的映射输出逐渐改变，从而在恒等映射的基础上产生一个有助于提升目标函数值的偏置（例如，使解更接近最优解）。利用多个迭代中的可学习压缩模块，算法 3 模型可以累积多次性能提升，比同迭代次数的算法 2 性能更佳。

3　实验结果与分析

本节通过仿真实验，展示所提出的通信高效的分布式预编码算法 3 相比于传统算法在预编码性能和前传通信效率方面的优势，证明本文方案对于降低前传流量、支持更大天线阵列的意义。

仿真设置如下，基站天线数 $M=64$，分为 $C=8$ 簇，用户数 $K=8$，每个用户的天线数 $N_k=4$，数据流数 $D_k=2$，则总天线数 $N=32$，总流数 $D=16$。采用 QuaDRiGa（quasi deterministic radio channel generator）信道生成套件（版本 v2.2.0）[18]按照 3GPP-mmw 标准建模[19]生成信道数据。训练集包含 12 000 个信道矩阵，测试集包含 1 200 个信道矩阵。仿真信道参数设定如表 1 所示。

在算法 3 的训练中，样本的 SNR 在−10～25 dB 均匀随机产生。训练和预测中，算法 3 的迭代次数固定为 $T=2$。将算法 3 与现有方法 WMMSE 预编码进行对比，其中，WMMSE 和 R-WMMSE 的迭代次数都为 6 次，与完全收敛的性能之间还存在一定差距，这部分性能区间用于展示算法 3 对性能的优化。

图 2(a)和图 2(b)分别展示了将 $\boldsymbol{X}\in\mathbb{C}^{32\times16}$ 的维度压缩为 16×16 和 12×16 时在 DSC、SSC、FC 这 3 种数据压缩模块下算法 3 的性能。图 2(a)将 $\boldsymbol{X}$ 压缩到了其秩的维度，而图 2(b)则将 $\boldsymbol{X}$ 压缩到了比其秩更小的维度。实验中 WMMSE 与 R-WMMSE 的性能几乎一致，代表了使用“无损压缩”的现有方法的性能。

从图 2(a)可以看到，当 $\boldsymbol{X}$ 被压缩到其秩的维度时，本文提出的算法 3 在 DSC、SSC 压缩模块下的性能都优于 R-WMMSE 算法。3 种模块的模型训练目标都是利用自身特定的映射结构，尝试将输入矩阵映射为一个性能更强的解。其性能提升机制和 2.4 节末尾所考虑的特殊情况类似，但并不完全相同。在这种实验条件下，可学习模块的输出在提升目标函数值时，还需要对抗维度压缩的损失。不同的模块表达能力导致了不同的性能。

表 1　　**仿真信道参数设定**

参数名称	参数取值	代码取值	含义
中央频率	qd_simulation_parameters. center_frequency	4.9×10^9	频谱中心频率为 4.9 GHz
3GPP 基线	qd_simulation_parameters.use_3GPP_baseline	1	使用 3GPP 规定的信道特性，不使用额外特性
随机相位	qd_simulation_parameters.use_random_initial_phase	1	使用随机相位
自相关函数	qd_simulation_parameters.autocorrelation_function	‘Comb300’	使用梳状自相关函数，包含 300 个正弦信号，用于产生与空间位置相关的随机信号
阵列类型	tx_array. qd_arrayant[1]	‘3GPP-mmw’	3GPP 毫米波阵列
通道垂直振子数	tx_array.qd_arrayant[2]	4	每通道 4 垂直振子
通道水平振子数	tx_array.qd_arrayant[3]	1	每通道 1 水平振子
极化类型	tx_array.qd_arrayant[5]	6	±45°极化
天线下倾角	tx_array.qd_arrayant[6]	15	下倾角 15°
振子间距	tx_array.qd_arrayant[7]	0.5	振子间距半波长
每列通道数	tx_array.qd_arrayant[8]	4	每列 4 通道
每行通道数	tx_array.qd_arrayant[9]	4	每行 4 通道
通道垂直间距	tx_array.qd_arrayant[10]	0.5	垂直间距半波长
通道水平间距	tx_array.qd_arrayant[11]	0.5	水平间距半波长
接收天线类型	rx_array.qd_arrayant[1]	‘xpol4’	4 极化天线
基站抬高	—	25	基站抬高 25 m
用户随机分布区域	—	—	天线阵列指向方向，以基站为圆心、半径为 120 m、角度为 120°的扇形区域
用户随机分布类型	—	—	均匀分布

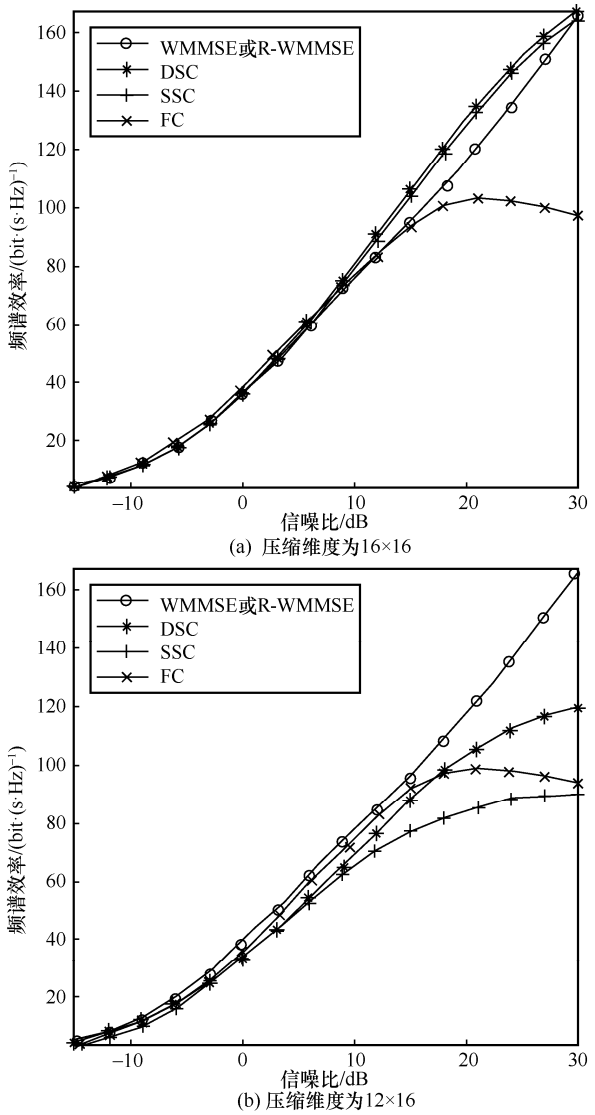

图 2　压缩维度为 16×16 和 12×16 时在 3 种数据压缩模块下算法 3 的性能

① FC 的参数量和复杂度都较高，性能方面反而表现较差。原因在于其参数量过多，结构过于复杂，导致泛化性较差。典型表现如图 2 所示，当测试信噪比接近 25 dB 边界时，使用信噪比−10～25 dB 数据训练出来的 FC 模块性能显著下降。

② DSC 因其较强的输入输出关系表达能力和适中的参数量，具备最佳的性能。与 FC 模块相比，DSC 模块充分利用了输入矩阵的行列信息，左乘提取输入的行间特征（左乘矩阵的每一行可以视作一个特征提取向量），改变矩阵列空间，右乘则正好相反。

③ SSC 相比 DSC 具有更简单的结构，只能提取行间或列间关系，变换单边子空间，但是由于结构更简单，因此更不容易产生过拟合。在较低的复杂度下，仍然可实现良好的泛化性能。

从图2(b)可以看出，当压缩后的维度低于其秩时，3 种模块的性能相比图 2(a)都有所下降，且全部比 R-WMMSE 预编码更低。各模块的性能下降幅度不同，由于 DSC 和 SSC 的运算过程始终保持矩阵结构，过小的压缩维度将导致运算过程降低矩阵的秩，产生信息丢失，削弱这 2 种矩阵模块的表达能力。相比之下，FC 模块则不受矩阵秩的影响。因此，和图 2(a)相比，DSC 和 SSC 的性能损失较大，而 FC 的损失较小。然而，需要强调的一点是，预编码算法应用的核心指标是可达速率，如果可达速率不达标，那么继续降低交互量便没有意义。图 2(b)中的结果表明维度压缩的损失较大，无法通过可学习模块完全补偿，因此需要采用更大的压缩维度。

图 2 的结果表明压缩维度（前传交互流量）和性能之间存在折中。在保证性能的前提下，DSC 和 SSC 可以实现更好的预编码性能和压缩维度的折中。此外，在适当的压缩维度下，DSC、SSC 相比 FC 展现出来的性能优势体现了 2 种矩阵结构的模块设计的优势。

将 $\boldsymbol{X}$ 的维度压缩至 16×16，并固定训练和测试的 SNR 为 20 dB，各算法的性能对比如图 3 所示。对比各算法关于不同输入样本的性能范围，发现 DSC 和 SSC 的频谱速率在不同样本上的差异都在 10 bit/(s·Hz)左右，而 FC 和 R-WMMSE 的差异都达到了 15 bit/(s·Hz)。图 2 和图 3 的实验结果都表明，分布式算法框架和可学习压缩模块联合优化的模型，既从经典算法的计算结构中获得了“鲁棒的性能保证”，又依靠可学习压缩模块获得了“降交互和提性能”的潜力。

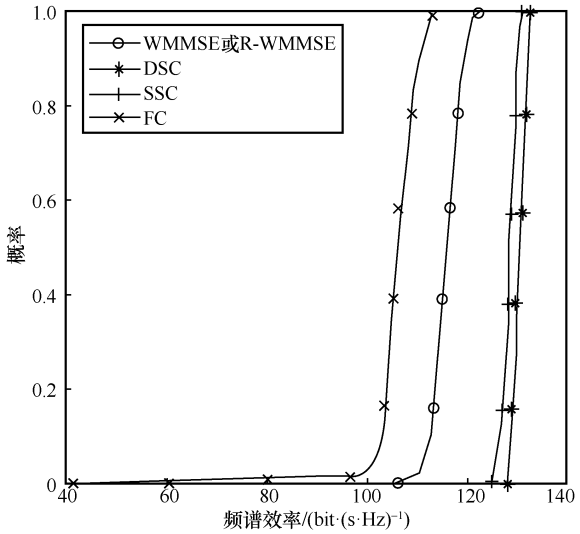

图 3　固定 SNR = 20 dB 时各算法的性能对比

接下来，对比各算法的数据交互量和计算复杂度。表 2 展示了各算法的前传流量大小。从表 2 可

以看到，本文提出的算法 3 在不同的压缩模块下，数据交互量都比 R-WMMSE 小。例如，当压缩维度为 16×16 时，本文提出的算法 3 的数据交互量比 R-WMMSE 降低了多达 25.0%。

表 2　　各算法的前传流量大小

算法	数据交互量/个
R-WMMSE	$\frac{1}{2}CN^2+CDN=8\ 192$
DSC 16×16 SSC 16×16 FC 256×1	$\frac{1}{2}CN^2+\frac{1}{2}CDN=6\ 144(-25.0\%)$
DSC 12×16 SSC 12×16 FC 192×1	$\frac{1}{2}CN^2+\frac{3}{8}CDN=5\ 632(-31.2\%)$

表 3 统计了各算法的复数乘法次数。从表 3 可以看到，本文提出的采用 DSC 和 SSC 的分布式预编码算法在计算复杂度方面相比 R-WMMSE 有相当大的优势，可以极大地降低基带处理的时延。例如，当压缩维度为 16×16 时，采用 DSC 的分布式预编码算法比 R-WMMSE 的计算复杂度降低了 52.9%。

表 3　　各算法的复数乘法次数

方法	复杂度/次
R-WMMSE	400 320
FC 256×1	426 112 (+6.4%)
FC 192×1	360 576 (−9.9%)
DSC 16×16	188 544 (−52.9%)
SSC 16×16	180 352 (−55.0%)
DSC 12×16	182 400 (−54.4%)
SSC 12×16	176 256 (−56.0%)

最后，本文提供了一个参考策略，指导如何在应用中选择合适的模块。这包括选择合适的压缩维度和从 SSC、DSC 中选出一种模块。模块的选择要满足系统的核心需求，例如，在本文所考虑的应用中，核心需求是性能和数据交互，前者保证系统的实用性，后者对应于模块的基本功能。压缩维度是影响这 2 个指标的首要条件。如果系统对性能有严格要求，设计者可以测试 SSC 和 DSC 在不同压缩维度下的性能，找到符合性能需求的压缩维度。然后选择模块。如果在计算复杂度和模型存储（模型参数量）方面没有特别要求，选择 DSC 即可；否则，可以根据计算复杂度和模型存储的具体表现进一步选择。总之，模块选择是一个帕累托最优点的选择问题，需要通过实验，根据系统对不同指标的要求程度做出权衡。

此外，一种经验性的选用策略是，在压缩维度方面尽量保证压缩后矩阵的秩不比原矩阵秩更低，模块选择方面在对计算复杂度和存储没有严苛要求的情况下选用 DSC 模块即可，否则需要基于不同帕累托最优点的实验结果，根据性能指标的重要性进行权衡。

4　结束语

随着未来通信系统中基站天线数的持续增长，BBU 间进行信号处理的前传流量也将极大增加。为了降低前传数据交互，支持更大的天线阵列，本文提出了一种针对 Massive MIMO 系统的通信高效的分布式预编码方案。该方案以 R-WMMSE 分布式预编码作为算法框架，结合高效极简的可学习数据压缩模块设计，通过对两者进行联合优化，可以实现预编码性能和前传通信效率两方面的提升。仿真结果表明，相比于经典的 WMMSE 预编码算法，本文的分布式预编码方案具有更好的性能和更低的数据交互要求。

参考文献：

[1] CHATAUT R, AKL R. Massive MIMO systems for 5G and beyond networks-overview, recent trends, challenges, and future research direction[J]. Sensors (Basel, Switzerland), 2020, 20(10): 2753.

[2] ZTE Corporation. 5G massive MIMO network application[R]. 2020.

[3] BJÖRNSON E, HOYDIS J, SANGUINETTI L. Massive MIMO networks: spectral, energy, and hardware efficiency[J]. Foundations and Trends in Signal Processing, 2017, 11(3/4): 154-655.

[4] Samsung Electronics Co., Ltd. Massive MIMO for new radio[R]. 2020.

[5] LI K P, CASTAÑEDA O, JEON C, et al. Decentralized coordinate-descent data detection and precoding for massive MU-MIMO[C]//Proceedings of 2019 IEEE International Symposium on Circuits and Systems (ISCAS). Piscataway: IEEE Press, 2019: 1-5.

[6] RODRÍGUEZ S J, RUSEK F, EDFORS O, et al. Decentralized massive MIMO processing exploring daisy-chain architecture and recursive algorithms[J]. IEEE Transactions on Signal Processing, 2020, 68: 687-700.

[7] GUSTAVSSON U, FRENGER P, FAGER C, et al. Implementation challenges and opportunities in beyond-5G and 6G communication[J]. IEEE Journal of Microwaves, 2021, 1(1): 86-100.

[8] LEE W, SIMEONE O, KANG J, et al. Multivariate fronthaul quantization for downlink C-RAN[J]. IEEE Transactions on Signal Processing, 2016, 64(19): 5025-5037.

[9] LI K P, SKARAN R, CHEN Y J, et al. Decentralized beamforming for massive MU-MIMO on a GPU cluster[C]//Proceedings of 2016 IEEE Global Conference on Signal and Information Processing (GlobalSIP). Piscataway: IEEE Press, 2017: 590-594.

[10] LI K P, SHARAN R R, CHEN Y J, et al. Decentralized baseband processing for massive MU-MIMO systems[J]. IEEE Journal on Emerging and Selected Topics in Circuits and Systems, 2017, 7(4): 491-507.

[11] SPENCER Q H, SWINDLEHURST A L, HAARDT M. Zero-forcing methods for downlink spatial multiplexing in multiuser MIMO channels[J]. IEEE Transactions on Signal Processing, 2004, 52(2): 461-471.

[12] CASTAÑEDA O, JACOBSSON S, DURISI G, et al. Finite-alphabet Wiener filter precoding for mmWave massive MU-MIMO systems[C]//Proceedings of 2019 53rd Asilomar Conference on Signals, Systems, and Computers. Piscataway: IEEE Press, 2020: 178-183.

[13] SARAJLIĆ M, RUSEK F, RODRÍGUEZ S J, et al. Fully decentralized approximate zero-forcing precoding for massive MIMO systems[J]. IEEE Wireless Communications Letters, 2019, 8(3): 773-776.

[14] SHI Q J, RAZAVIYAYN M, LUO Z Q, et al. An iteratively weighted MMSE approach to distributed sum-utility maximization for a MIMO interfering broadcast channel[J]. IEEE Transactions on Signal Processing, 2011, 59(9): 4331-4340.

[15] ZHAO X T, LU S Y, SHI Q J, et al. Rethinking WMMSE: can its complexity scale linearly with the number of BS antennas?[J]. IEEE Transactions on Signal Processing, 2023, 71: 433-446.

[16] CHRISTENSEN S S, AGARWAL R, CARVALHO E D, et al. Weighted sum-rate maximization using weighted MMSE for MIMO-BC beamforming design[J]. IEEE Transactions on Wireless Communications, 2008, 7(12): 4792-4799.

[17] BJÖRNSON E, LARSSON E G, MARZETTA T L. Massive MIMO: ten myths and one critical question[J]. IEEE Communications Magazine, 2016, 54(2): 114-123.

[18] JAECKEL S, RASCHKOWSKI L, BÖRNER K, et al. QuaDRiGa: a 3-D multi-cell channel model with time evolution for enabling virtual field trials[J]. IEEE Transactions on Antennas and Propagation, 2014, 62(6): 3242-3256.

[19] JAECKEL S, RASCHKOWSKI L, WU S B, et al. An explicit ground reflection model for mm-wave channels[C]//Proceedings of 2017 IEEE Wireless Communications and Networking Conference Workshops (WCNCW). Piscataway: IEEE Press, 2017: 1-5.

[作者简介]

李勉（1999− ），男，江西萍乡人，香港中文大学（深圳）博士生，主要研究方向为信号处理和机器学习优化方法。

李洋（1989− ），男，吉林长春人，博士，深圳市大数据研究院副研究员，主要研究方向为无线资源管理、AI 辅助优化、大规模优化等。

张纵辉（1981− ），男，台湾桃园人，博士，香港中文大学（深圳）副教授，主要研究方向为面向无线通信、机器学习的关键信号处理和优化方法等。

史清江（1980− ），男，浙江绍兴人，博士，同济大学教授，主要研究方向为网络系统优化设计、网络/信号大数据、分布式机器学习等。