Model-driven Distributed WMMSE for Downlink Massive MIMO Systems

Ningxin Zhou, *Student Member, IEEE,* Zheng Wang, *Senior Member, IEEE,*
Qingjiang Shi, *Member, IEEE,* Wei Xu, *Fellow, IEEE* and Yongming Huang, *Fellow, IEEE*

***Abstract*—Distributed signal processing based on the decentralized architecture plays an important role in the next generation of wireless communications. In this paper, we propose a model-driven distributed WMMSE algorithm (MDD-WMMSE) for downlink massive MIMO systems. The proposed MDD-WMMSE is model-driven, which is designed based on a distributed version of the traditional WMMSE algorithm. Specifically, each distributed unit (DU) in it has a local private network and calculates in parallel, while the related distributed training is achieved by simply exchanging the local information. Simulation results show that the proposed MDD-WMMSE achieves competitive performance compared to the centralized WMMSE algorithm but with much lower complexity.**

***Index Terms*—Distributed precoding, massive MIMO, deep unfolding, distributed learning.**

I. Introduction

The celebrated weighted minimum mean squared error (WMMSE) algorithm [1], [2] has been widely applied in the downlink massive multiple-input multiple-output (MIMO) systems due to its rapid convergence to the stationary point of the maximum weighted sum rate problem. Moreover, with respect to WMMSE, a lot of deep learning methods have been proposed. Among them, some black box networks such as deep neural network (DNN) [3] and convolutional neural network (CNN) [4] have been applied to approach the performance of WMMSE with lower complexity. Meanwhile, another model-driven learning method named as deep unfolding is also applied to build more explainable networks by unfolding the iterations of some reformulated WMMSE algorithms [5], [6], which is also be extended to the rate-splitting multiple access (RSMA) scheme [7].

However, as the system dimension increases, the traditional centralized architecture becomes unaffordable due to the high demands for communication bandwidth and computing power of the central unit (CU) [8]. To solve it, some distributed precoder networks based on decentralized architecture have been proposed. In particular, two distributed network based on horizontal federated learning (FL) and vertical FL are designed in [9] for precoder design in cell-free MIMO systems. Nevertheless, a performance gap between these distributed networks and WMMSE still exists. In [10], a distributed-learning-based uplink hybrid beamformer is proposed, where the network is constructed by establishing communications among users, access points, and the CU.

Ningxin Zhou, Zheng Wang, Wei Xu, and Yongming Huang are with the School of Information Science and Engineering, Southeast University, Nanjing 210096, China (email: wznuaa@gmail.com); Qingjiang Shi is with the School of Software Engineering, Tongji University, Shanghai 201804, China.

In this paper, we propose a distributed model-driven precoder named MDD-WMMSE for downlink massive MIMO systems. Specifically, the proposed MDD-WMMSE is built on a distributed version of WMMSE, where each DU trains its network in a parallel distributed manner by exchanging local information. Simulation results show that the proposed MDD-WMMSE has comparable performance and lower complexity compared to the traditional WMMSE algorithm.

II. System Model and Problem Formulation

Consider a massive MIMO system with M antennas base station (BS) serving K users, where each user has N receiving antennas. Let $\mathbf{s}_k \in \mathbb{C}^d$ denote the signal vector transmitting to user $k \in \mathcal{K} = \{1, ..., K\}$, where d represents the size of the transmission data streams. Meanwhile, it is also assumed that the signal vectors transmitted to different users are independent with zero mean and $\mathbb{E}\left[\mathbf{s}_k \mathbf{s}_k^H\right] = \mathbf{I}$. Define $\mathbf{H}_k \in \mathbb{C}^{N\times M}$ and $\mathbf{V}_k \in \mathbb{C}^{M\times d}$ as the channel matrix in Rayleigh fading channel model and precoding matrix between BS and user k, respectively. Then, the receive signal $\mathbf{y}_k$ at k-th user is

$$\mathbf{y}_k = \mathbf{H}_k \mathbf{V}_k \mathbf{s}_k + \sum_{m=1, m\neq k}^{K} \mathbf{H}_k \mathbf{V}_m \mathbf{s}_m + \mathbf{n}_k, \tag{1}$$

where $\mathbf{n}_k \in \mathbb{C}^{N\times 1}$ is the additive white Gaussian noise (AWGN) with $\mathcal{CN}(\mathbf{0}, \sigma_k^2 \mathbf{I})$. Then, the signal-to-interference-plus-noise ratio (SINR) of the k-th user is given by

$$\text{SINR}_k = \mathbf{H}_k \mathbf{V}_k \mathbf{V}_k^H \mathbf{H}_k^H \left(\sum_{m\neq k}^{\mathcal{K}} \mathbf{H}_k \mathbf{V}_m \mathbf{V}_m^H \mathbf{H}_k^H + \sigma_k^2 \mathbf{I} \right)^{-1}, \tag{2}$$

which leads to the downlink achievable rate of user k as

$$R_k = \log\det(\mathbf{I} + \text{SINR}_k). \tag{3}$$

Clearly, based on R_k, precoding aims to maximize the weighted sum rate $R = \sum_{k=1}^{K} \alpha_k R_k$ under the power constraint, i.e.,

$$\begin{aligned} \max_{\{\mathbf{V}_k\}_{k\in\mathcal{K}}} \; & R = \sum_{k=1}^{K} \alpha_k R_k \\ \text{s.t.} \quad & \text{Tr}\left(\sum_{k=1}^{K} \mathbf{V}_k \mathbf{V}_k^H \right) \leq P, \end{aligned} \tag{4}$$

where α_k is the weight of user k and P denotes the total transmit power budget at the transmitter. Since such a problem

is non-convex and NP-hard, it is usually converted to the following equivalent formation [1], [2]

$$\min_{\{\mathbf{U}_k,\mathbf{W}_k,\mathbf{V}_k\}_{k\in\mathcal{K}}} \sum_{k=1}^{K} \alpha_k \Big(\mathrm{Tr}\left(\mathbf{W}_k\mathbf{E}_k\right) - \log\det\left(\mathbf{W}_k\right) \Big) \tag{5a}$$

$$\text{s.t.} \quad \mathrm{Tr}\left(\sum_{k=1}^{K} \mathbf{V}_k\mathbf{V}_k^H\right) \le P, \tag{5b}$$

with the mean squared error (MSE) matrix

$$\begin{aligned}\mathbf{E}_k = &\left(\mathbf{I} - \mathbf{U}_k^H\mathbf{H}_k\mathbf{V}_k\right)\left(\mathbf{I} - \mathbf{U}_k^H\mathbf{H}_k\mathbf{V}_k\right)^H \\ &+ \sum_{m\neq k}^{K} \mathbf{U}_k^H\mathbf{H}_k\mathbf{V}_m\mathbf{V}_m^H\mathbf{H}_k^H\mathbf{U}_k + \sigma_k^2\mathbf{U}_k^H\mathbf{U}_k,\end{aligned} \tag{6}$$

where $\mathbf{U}_k \in \mathbb{C}^{N\times d}$ and $\mathbf{W}_k \in \mathbb{C}^{d\times d}$ are two introduced auxiliary matrices.

To obtain a stationary solution of problem (5), the block coordinate descent (BCD) scheme is used to sequentially solve $\mathbf{U}_k, \mathbf{W}_k$, and $\mathbf{V}_k$, which is known as the WMMSE algorithm with the following iterative expressions [2]

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

$$\mathbf{W}_k = \left(\mathbf{I} - \mathbf{U}_k{}^H\mathbf{H}_k\mathbf{V}_k\right)^{-1}, \tag{8}$$

$$\mathbf{V}_k = \alpha_k\left(\sum_{m=1}^{K} \alpha_m\mathbf{H}_m^H\mathbf{U}_m\mathbf{W}_m\mathbf{U}_m^H\mathbf{H}_m + \mu\mathbf{I}\right)^{-1}\mathbf{H}_k^H\mathbf{U}_k\mathbf{W}_k. \tag{9}$$

III. Deep-unfolding based Distributed WMMSE

A. Decentralized Architecture

We now upgrade the traditional centralized architecture into a distributed architecture, thus serving for the following distributed networks.

In particular, partition the M antennas at BS into B ($1 \le B \le M$) DUs, and then each DU has C transmit antennas (i.e. $BC = M$), its own dedicated RF circuitry, and baseband signal processing units. Consequently, the local channel matrix between DU b and user k is $\mathbf{H}_{k,b} \in \mathbb{C}^{N\times C}$, and $\mathbf{H}_{:,b} = [\mathbf{H}_{1,b}^H, \cdots, \mathbf{H}_{K,b}^H]^H \in \mathbb{C}^{KN\times C}$ represents the local channel state information (CSI) stored in the b-th DU. Similarly, the local precoding matrix in the b-th DU is $\mathbf{V}_{:,b} = [\mathbf{V}_{1,b}, \cdots, \mathbf{V}_{K,b}] \in \mathbb{C}^{C\times Kd}$, where $\mathbf{V}_{k,b} \in \mathbb{C}^{C\times d}$ denotes the precoding matrix for user k. Moreover, the relationship between the local channel matrix $\mathbf{H}_{k,b}$ and the global channel matrix $\mathbf{H}_k$ is $\mathbf{H}_k = [\mathbf{H}_{k,1}, \cdots, \mathbf{H}_{k,B}]$, and similarly we have $\mathbf{V}_k^H = [\mathbf{V}_{k,1}^H, \cdots, \mathbf{V}_{k,B}^H]^H$. To be more specific, as shown in Fig. 1, the received signal vector $\mathbf{y}_k$ in this decentralized architecture can be written as

$$\mathbf{y}_k = \sum_{b=1}^{B} \mathbf{H}_{k,b}\mathbf{V}_{k,b}\mathbf{s}_k + \sum_{m\neq k}^{\mathcal{K}}\sum_{b=1}^{B} \mathbf{H}_{k,b}\mathbf{V}_{m,b}\mathbf{s}_m + \mathbf{n}_k. \tag{10}$$

Different from solving (4) in a centralized architecture, each DU in the decentralized architecture needs to independently solve own precoding matrix $\mathbf{V}_{:,b}$ with local CSI $\mathbf{H}_{:,b}$ which is only stored in the local storage unit.

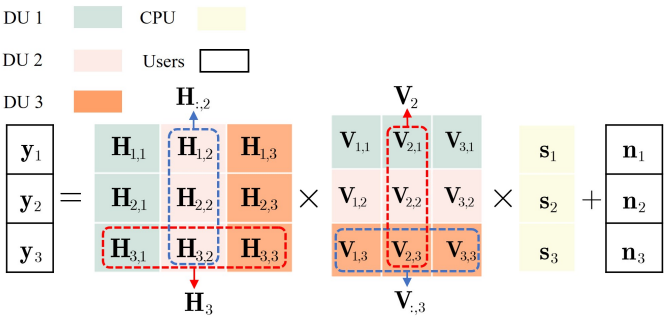

Fig. 1. An example of a downlink communication model ($B = 3$, $K = 3$).

B. Distributed WMMSE

According to the iterative expressions in (7)–(9), since the global CSI $\mathbf{H}_k$ is required to solve $\mathbf{V}_k$, the operations of the traditional WMMSE can not well suit this decentralized architecture, which leads to the following distributed WMMSE accordingly.

Specifically, we divide the problem in (5) into B distributed subproblems, where each DU corresponds to a subproblem shown below

$$\min_{\{\mathbf{U}_{k,b},\mathbf{W}_{k,b}\}_{k\in\mathcal{K}},\mathbf{V}_{:,b}} \sum_{k=1}^{K} \alpha_k \Big(\mathrm{Tr}\left(\mathbf{W}_{k,b}\mathbf{E}_{k,b}\right) - \log\det\left(\mathbf{W}_{k,b}\right) \Big)$$

$$\text{s.t.} \quad \mathrm{Tr}\left(\sum_{k=1}^{K} \mathbf{V}_{k,b}\mathbf{V}_{k,b}^H\right) \le P/B \tag{11}$$

with fixed $\mathbf{V}_{:,\bar{b}}, \bar{b} \neq b$ and

$$\begin{aligned}\mathbf{E}_{k,b} = &\left(\mathbf{I} - \mathbf{U}_{k,b}^H\mathbf{H}_k\mathbf{V}_k\right)\left(\mathbf{I} - \mathbf{U}_{k,b}^H\mathbf{H}_k\mathbf{V}_k\right)^H \\ &+ \sum_{m\neq k}^{K} \mathbf{U}_{k,b}^H\mathbf{H}_k\mathbf{V}_m\mathbf{V}_m^H\mathbf{H}_k^H\mathbf{U}_{k,b} + \sigma_k^2\mathbf{U}_{k,b}^H\mathbf{U}_{k,b}.\end{aligned} \tag{12}$$

Here, due to the independence of calculating $\mathbf{U}_k$ and $\mathbf{W}_k$ among different DUs, we use $\mathbf{U}_{k,b} \in \mathbb{C}^{N\times d}$ and $\mathbf{W}_{k,b} \in \mathbb{C}^{d\times d}$ to represent the results of the b-th DU. Meanwhile, the power budget is evenly distributed to each DU, i.e. the local power budget of each DU is P/B.

Then, with respect to the subproblem in (11), each DU applies BCD method as follows

$$\mathbf{U}_{k,b} = \left(\sum_{m=1}^{K} \boldsymbol{\xi}_{k,m}\boldsymbol{\xi}_{k,m}^H + \sigma_k^2\mathbf{I}\right)^{-1}\boldsymbol{\xi}_{k,k}, \tag{13}$$

$$\mathbf{W}_{k,b} = \left(\mathbf{I} - \mathbf{U}_{k,b}^H\boldsymbol{\xi}_{k,k}\right)^{-1}, \tag{14}$$

$$\begin{aligned}\mathbf{V}_{k,b} = &\left(\sum_{m=1}^{K} \alpha_m\mathbf{H}_{m,b}^H\mathbf{A}_{m,b}\mathbf{H}_{m,b} + \mu_b\mathbf{I}\right)^{-1}\Bigg(\alpha_k\mathbf{H}_{k,b}^H\mathbf{U}_{k,b}\mathbf{W}_{k,b} \\ &- \sum_{m=1}^{K} \alpha_m\mathbf{H}_{m,b}^H\mathbf{A}_{m,b}\left(\boldsymbol{\xi}_{m,k} - \mathbf{H}_{m,b}\mathbf{V}_{k,b}\right)\Bigg)\end{aligned} \tag{15}$$

with $\boldsymbol{\xi}_{i,j} = \sum_{b=1}^{B} \mathbf{H}_{i,b}\mathbf{V}_{j,b}$ and $\mathbf{A}_{m,b} = \mathbf{U}_{m,b}\mathbf{W}_{m,b}\mathbf{U}_{m,b}^H$. By iteratively calculating (13)–(15), each DU can solve the distributed problem in (11) independently. Moreover, by solving these B subproblems among all DUs in a sequential order, the

Algorithm 1 Distributed WMMSE

Input: $\mathbf{H}_{k,b}$, ε, $\{\sigma_k\}_{k\in\mathcal{K}}$, P, $\forall b,k$, L.
Output: $\mathbf{U}_{k,b}$,$\mathbf{W}_{k,b}$,$\mathbf{V}_{k,b}$, $\forall b,k$.
Initialize: Set $\mathbf{V}_{k,b}$ to satisfy $\mathrm{Tr}(\mathbf{V}_{k,b}\mathbf{V}_{k,b}^H)=P/(BK)$.
repeat(Outer loop)
  for $b=1$ to B **do**
    for $l=1$ to L **do** (Inner loop)
      update $\mathbf{U}_{k,b}$ via (13), $\forall k$ in b-th DU.
      update $\mathbf{W}_{k,b}$ via (14), $\forall k$ in b-th DU.
      update $\mathbf{V}_{k,b}$ via (15), $\forall k$ in b-th DU.
    end for
  end for
until Reaching convergence conditions

stationary point of the following problem can be approached [11]

$$\min_{\{\mathbf{U}_k,\mathbf{W}_k,\mathbf{V}_k\}_{k\in\mathcal{K}}} \sum_{k=1}^{K}\alpha_k\Big(\mathrm{Tr}\left(\mathbf{W}_k\mathbf{E}_k\right)-\log\det\left(\mathbf{W}_k\right)\Big) \quad (16a)$$

$$\text{s.t.}\quad \mathrm{Tr}\left(\sum_{k=1}^{K}\mathbf{V}_{k,b}\mathbf{V}_{k,b}^H\right)\le P/B,\forall b. \quad (16b)$$

Clearly, the difference between the problems in (16) and (5) lies in the different constraints. Assuming f_1^* is the minimum value of problem in (5) and f_2^* is the minimum value of problem of (16), since the constraint range in (16b) is included in the constraint range in (5b), the difference between these two minimum values satisfies $\Delta f=f_2^*-f_1^*\ge 0$, which can be well controlled with the appropriate power allocation. More specifically, it has been shown in [12] that a negligible Δf can be reached with the average power allocation.

To summarize, the proposed distributed WMMSE algorithm is given by Algorithm 1. However, although it can achieve the near performance as the centralized one, the sequential computations among DUs are required for the computations in (13)–(15).

IV. Proposed Deep-Unfolding-based Distributed WMMSE Design

Based on the above distributed WMMSE, we now propose a model-driven deep-unfolding-based distributed WMMSE network (MDD-WMMSE), where each DU can operate in a more parallel way with less other DUs' information. By doing this, the sequential implementation of the proposed distributed WMMSE can be effectively avoided without performance loss.

Specifically, as shown in Fig.2, the proposed MDD-WMMSE mainly consists of two stages. The first stage includes T layers without information exchange, and the second stage has T_e layers with information exchange supported by CU.

A. The First Stage of MDD-WMMSE

The first stage of MDD-WMMSE is built by reformulating the T iterations from $\mathbf{U}_{k,b}$ to $\mathbf{V}_{k,b}$ in (13)-(15), where superscript t denotes the t-th layer at the first stage.

To avoid the information exchange, a trainable matrix $\mathbf{O}_b^t\in\mathbb{C}^{KN\times Kd}$ is set to approximate the information of other DUs for the b-th DU. Then, given $\hat{\mathbf{P}}_b^t=\mathbf{O}_b^t+\mathbf{H}_{:,b}\mathbf{V}_{:,b}^{t-1}$, the update of $\mathbf{U}_{k,b}^t$ and $\mathbf{W}_{k,b}^t$ can be written as

$$\mathbf{U}_{k,b}^t=\left(\hat{\boldsymbol{\xi}}_{k,:}^t(\hat{\boldsymbol{\xi}}_{k,:}^t)^H+\sigma^2\mathbf{I}\right)^{-1}\hat{\boldsymbol{\xi}}_{k,k}^t, \quad (17)$$

$$\bar{\mathbf{W}}_{k,b}^t=(\mathbf{W}_{k,b}^t)^{-1}=\mathbf{I}-(\mathbf{U}_{k,b}^t)^H\hat{\boldsymbol{\xi}}_{k,k}^t. \quad (18)$$

Here, $\hat{\boldsymbol{\xi}}_{k,m}^t=\hat{\mathbf{P}}_b^t[(k-1)N+1:kN,(m-1)d+1:md]\in\mathbb{C}^{N\times d}$, which is an approximate version of $\boldsymbol{\xi}_{k,m}$, and $\hat{\boldsymbol{\xi}}_{k,:}=\hat{\mathbf{P}}_b^t[(k-1)N+1:kN,:]\in\mathbb{C}^{N\times Kd}$.

As for the update of the local precoding matrix $\mathbf{V}_{:,b}$ in MDD-WMMSE, the design of this part does not rely on the expression in (15), but adopts its equivalent expression as

$$\begin{aligned}\mathbf{V}_{:,b}^t=&\mathbf{H}_{:,b}^H\mathbf{U}_{:,b}^t\left((\mathbf{U}_{:,b}^t)^H\mathbf{H}_{:,b}\mathbf{H}_{:,b}^H\mathbf{U}_{:,b}^t+\mu_b^t\bar{\mathbf{W}}_{:,b}^t\right)^{-1}\Big(\mathbf{I}\\&-(\mathbf{U}_{:,b}^t)^H\sum_{\bar{b}\neq b}\mathbf{H}_{:,\bar{b}}\mathbf{V}_{:,\bar{b}}\Big),\end{aligned} \quad (19)$$

with $\mathbf{U}_{:,b}^t$ =blkdiag($\mathbf{U}_{1,b}^t$, $\cdots$, $\mathbf{U}_{K,b}^t$) and $\bar{\mathbf{W}}_{:,b}^t$ =blkdiag ($\alpha_1\bar{\mathbf{W}}_{1,b}^t,\cdots,\alpha_K\bar{\mathbf{W}}_{K,b}^t$), where blkdiag($\mathbf{A}_1$, $\mathbf{A}_2$) represents a block diagonal matrix with $\mathbf{A}_1$ and $\mathbf{A}_2$ as the diagonal blocks. Here, the equivalence is guaranteed by $(\mathbf{I}+\mathbf{AB})^{-1}\mathbf{A}=\mathbf{A}(\mathbf{I}+\mathbf{BA})^{-1}$.

According to (19), we firstly set the Lagrange multiplier μ_b^t as a trainable parameter, since the eigenvalue decomposition and bisection search for searching μ_b^t are not amenable to be mapped to the network. Meanwhile, in (19), the matrix $\mathbf{J}_b^t=(\mathbf{U}_{:,b}^t)^H\mathbf{H}_{:,b}\mathbf{H}_{:,b}^H\mathbf{U}_{:,b}^t+\mu_b^t\bar{\mathbf{W}}_{:,b}^t$ also requires the inverse operation. To reduce the complexity, inspired by [6], an approximate expression containing trainable parameters is used as an alternative, i.e. $(\mathbf{J}_b^t)^{-1}\approx[\mathbf{J}_b^t]_{\text{diag}}^{-1}\mathbf{X}_b^t+\mathbf{J}_b^t\mathbf{Y}_b^t+\mathbf{Z}_b^t$, where $\mathbf{X}_b^t$, $\mathbf{Y}_b^t$, $\mathbf{Z}_b^t\in\mathbb{C}^{Kd\times Kd}$ are trainable matrices, and $[\mathbf{J}_b^t]_{\text{diag}}^{-1}$ is the inverse matrix of diagonal elements of $\mathbf{J}_b^t$. Therefore, we have

$$\begin{aligned}\hat{\mathbf{V}}_{:,b}^t=&\mathbf{H}_{:,b}^H\mathbf{U}_{:,b}^t\Big([\mathbf{J}_b^t]_{\text{diag}}^{-1}\mathbf{X}_b^t+\mathbf{J}_b^t\mathbf{Y}_b^t+\mathbf{Z}_b^t\Big)\left(\mathbf{I}-(\mathbf{U}_{:,b}^t)^H\mathbf{O}_b^t\right)\\&+\mathbf{H}_{:,b}^H\mathbf{F}_b^t\end{aligned} \quad (20)$$

with a trainable compensation error matrix $\mathbf{F}_b^t\in\mathbb{C}^{KN\times Kd}$, which leads to the following expression by power constraint

$$\mathbf{V}_{:,b}^t=\sqrt{\frac{P}{B}}*\frac{\hat{\mathbf{V}}_{:,b}^t}{\|\hat{\mathbf{V}}_{:,b}^t\|_F}. \quad (21)$$

In general, the first stage of MDD-WMMSE in each DU contains T rounds of calculations from (17) to (21) (except for (19)) without any information exchange, so the calculations among DUs can be carried out in parallel. Note that MDD-WMMSE is based on (19) rather than (15) for promoting the learning efficiency of neural networks. More specifically, this comes from the following fact about the target problem in (4).

Proposition 1. *(Low-Dimensional Subspace Property) [13]: Any nontrivial stationary point $\mathbf{V}_k^*$ of problem (4) must lie in the range space of $\mathbf{H}^H=[\mathbf{H}_{:,1},\cdots,\mathbf{H}_{:,B}]^H\in\mathbb{C}^{KN\times M}$, i.e. $\mathbf{V}^*=[\mathbf{V}_1^*,\cdots,\mathbf{V}_B^*]=\mathbf{H}^H\hat{\mathbf{X}}$ with $\hat{\mathbf{X}}\in\mathbb{C}^{KN\times Kd}$.*

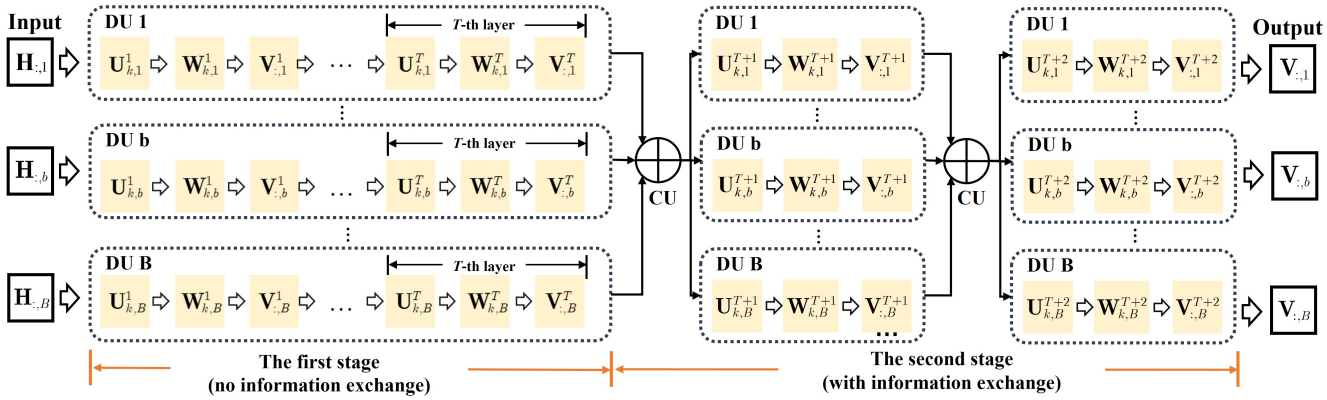

Fig. 2. The illustration of MDD-WMMSE with $T_e = 2$.

Expanding Proposition 1 to distributed cases, the local precoding matrix $\mathbf{V}_{:,b}$ for DU b that maximizes the global sum rate must lie in the range space of $\mathbf{H}_{:,b}^H$. Compared to the expression in (15), the one in (20) obeys this proposition, which introduces a more accurate coverage range to MDD-WMMSE with boosted training speed and performance.

B. The Second Stage of MDD-WMMSE

In the first stage, the information exchange is avoided by setting the trainable parameter $\mathbf{O}_b^t$. For the consideration of performance improvement, the second stage including T_e layers introduces information exchange with $T > T_e$.

Before the calculation of the t_e-th layer in the second stage, each DU calculates $\mathbf{H}_{:,b}\mathbf{V}_{:,b}^{T+t_e-1}$ and sends it to CU. Then, CU performs the accumulation operation, i.e. $\mathbf{P}^{t_e} = \sum_{b=1}^{B} \mathbf{H}_{:,b}\mathbf{V}_{:,b}^{T+t_e-1}$, and then feeds it back to each DU. Next, with the participation of $\mathbf{P}^{t_e}$ into the local network, each DU completes the following calculations

$$\mathbf{U}_{k,b}^{T+t_e} = \left(\boldsymbol{\xi}_{k,:}^{t_e}(\boldsymbol{\xi}_{k,:}^{t_e})^H + \sigma_k^2\mathbf{I}\right)^{-1}\boldsymbol{\xi}_{kk}^{t_e}, \tag{22}$$

$$\bar{\mathbf{W}}_{k,b}^{T+t_e} = \mathbf{I} - (\mathbf{U}_{k,b}^{T+t_e})^H\boldsymbol{\xi}_{kk}^{t_e}, \tag{23}$$

$$\bar{\mathbf{V}}_{:,b}^{T+t_e} = \mathbf{H}_{:,b}^H\mathbf{U}_{:,b}^{T+t_e}\left((\mathbf{U}_{:,b}^{T+t_e})^H\mathbf{H}_{:,b}\mathbf{H}_{:,b}^H\mathbf{U}_{:,b}^{T+t_e} + \mu_b^{T+t_e}\bar{\mathbf{W}}_{:,b}^{T+t_e}\right)^{-1}$$
$$\left(\mathbf{I} - (\mathbf{U}_b^{T+t_e})^H(\mathbf{P}^{t_e} - \mathbf{H}_{:,b}\mathbf{V}_{:,b}^{T+t_e-1})\right), \tag{24}$$

where $\boldsymbol{\xi}_{kk}^{t_e}$, $\boldsymbol{\xi}_{k,:}^{t_e}$ and $\mathbf{P}^{t_e}$ have the same relationship as $\hat{\boldsymbol{\xi}}_{kk}^{t}$, $\hat{\boldsymbol{\xi}}_{k,:}^{t}$ and $\hat{\mathbf{P}}_b^t$. To ensure the performance, the matrix inversion operation is remained here, where only $\mu_b^{T+t_e}$ is a trainable parameter in the above expressions.

In addition, due to the parallel calculations between DUs, an additional weighted correction operation is added, i.e.,

$$\hat{\mathbf{V}}_{:,b}^{T+t_e} = \mathbf{V}_{:,b}^{T+t_e-1} + \beta_b^{t_e}(\bar{\mathbf{V}}_{:,b}^{T+t_e} - \mathbf{V}_{:,b}^{T+t_e-1}) \tag{25}$$

with a trainable parameter $\beta_b^{t_e}$. Finally, perform the power normalization operation on $\hat{\mathbf{V}}_{:,b}^{T+t_e}$ by (21) to obtain $\mathbf{V}_{:,b}^{T+t_e}$.

C. Back Propagation and Complexity

In MDD-WMMSE, after the calculations in the above two stages of the forward propagation, each DU outputs its local precoding matrix $\mathbf{V}_{:,b}^{T+T_e}$ and then sends $\mathbf{H}_{:,b}\mathbf{V}_{:,b}^{T+T_e}$ to CU for the back propagation. Then, after CU accumulates the results, it returns $\mathbf{P}^{T_e} = \sum_{b=1}^{B} \mathbf{H}_{:,b}\mathbf{V}_{:,b}^{T+T_e}$ to each DU. Next, each DU independently calculates the sum rate R of the entire system based on $\mathbf{P}^{T_e}$ by

$$\begin{aligned} R = &\sum_{k=1}^{K} \alpha_k \log\det\left(\boldsymbol{\xi}_{k,:}^{T_e}(\boldsymbol{\xi}_{k,:}^{T_e})^H\right) \\ &- \sum_{k=1}^{K} \alpha_k \log\det\left(\boldsymbol{\xi}_{k,:}^{T_e}(\boldsymbol{\xi}_{k,:}^{T_e})^H - \boldsymbol{\xi}_{k,k}^{T_e}(\boldsymbol{\xi}_{k,k}^{T_e})^H + \sigma_k^2\mathbf{I}\right), \end{aligned} \tag{26}$$

which leads to the loss function for each DU as

$$Loss_b = -R, \forall b \in \{1, 2, \cdots, B\}. \tag{27}$$

Although the loss function is independently calculated by each DU, each DU actually shares the same loss function value so as to the global training performance.

Overall, during the training process, each DU first computes parallel based on its local CSI, then transmits $\mathbf{H}_{:,b}\mathbf{V}_{:,b}^{T}$ to the CU starting the second stage of the network. In the second stage stage, the DU completes relevant operations with the assistance of the CU. Eventually, each DU outputs the final result and sends $\mathbf{H}_{:,b}\mathbf{V}_{:,b}^{T+T_e}$ to the CU. After the CU aggregates $\mathbf{H}_{:,b}\mathbf{V}_{:,b}^{T+T_e}$, the result is returned to each DU. In back propagation, by correlating $\mathbf{P}_{T_e}$ with local $\mathbf{H}_{:,b}\mathbf{V}_{:,b}^{T+T_e}$, each DU computes the loss function via (26) and then updates local trainable parameters via the local updater, such as the Adam optimizer, dependently.

As for the complexity of MDD-WMMSE, it is $O(TBK^{2.37} + T_eBK^3 + (T+T_e)BKN^3 + (T+T_e)K^2MN)$, since the matrix multiplication operation with $O(K^{2.37})$ replaces matrix inversion with $O(K^3)$ in the first stage. Meanwhile, for centralized WMMSE, its complexity is $O(LKM^3 + LKN^3 + LK^2M^2N)$. Due to $M \gg K$ and T, $T_e \ll L$, the complexity of MDD-WMMSE is significantly lower than that of WMMSE.

V. Simulation Results

Fig. 3 shows the sum rates achieved by the proposed distributed WMMSE (D-WMMSE) in Algorithm 1 and the

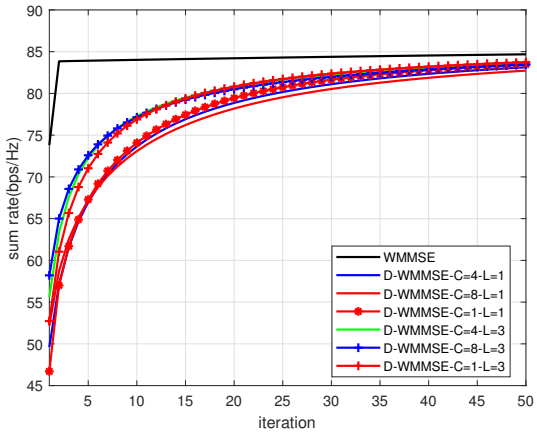

Fig. 3. The convergence curve of the WSR in a $M = 128$ BS antennas $K = 10$ UEs massive MU-MIMO system with $N = 2, d = 1$, SNR $= 15dB$ and $\alpha_k = 1, \forall k$.

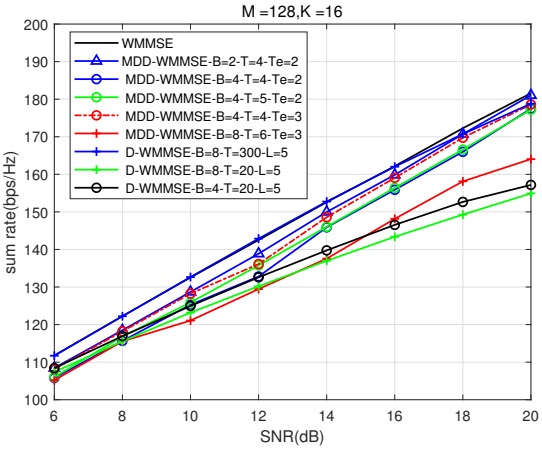

Fig. 4. WSR comparison in a $M = 128$ BS antennas $K = 16$ UEs massive MU-MIMO system with $N = 2, d = 1$ and $\alpha_k = 1, \forall k$.

centralized WMMSE under the different numbers of transmit antennas C and the numbers of iterations L in each DU. Intuitively, regardless of the values of C or L, the distributed version of WMMSE has negligible performance loss compared to the traditional WMMSE algorithm. In addition, the increases of C and L lead to an acceleration of convergence speed.

The comparison of the sum rates achieved by MDD-WMMSE and the centralized WMMSE under different B, T and T_e is shown in Fig. 4. The results of MDD-WMMSE are the average of 500 tests using the trained network. During network training, since deep learning models lack support for complex number training, all complex matrices are converted to real form. The training set consists of 24800 randomly generated channel matrices, and training uses the mini-batch mode, where the batch size is 128. Here, a random normal distribution, i.e. $\mathcal{N}(0, \omega^2)$ where ω is the scaler factor set as 0.01, is used to initialize each element in $\mathbf{X}_b^t$, $\mathbf{Y}_b^t$, $\mathbf{Z}_b^t$ and $\mathbf{F}_b^t$ with $\mu_b^t, \beta_b^{t_e} = 1$. Compared with the D-WMMSE, the MDD-WMMSE can achieve higher performance with lower layers. Obviously, an increase in the number of DUs B leads to an increase in performance loss, which means that more layers of MDD-WMMSE are needed. Moreover, increasing the number of layers T_e in the second stage can effectively improve the performance of MDD-WMMSE.

TABLE I
THE CPU TIME AND WSR OF MDD-WMMSE WITH $SNR = 10dB$.

(M, K)	(B,T,T_e)	CPU time(s)	WSR(%)
$(128, 16)$	WMMSE	16.77	131.55(bps/Hz)
	(4,4,2)	1.79	95.23%
	(4,4,3)	1.83	97.04%
	(4,5,2)	1.81	95.25%
	(8,6,3)	2.54	92.04%

Table I compares the CPU time of the prediction stage for different schemes in various scenarios. It demonstrates that the MDD-WMMSE can reach a performance close to that of the centralized WMMSE but with much lower computational complexity. In summary, the MDD-WMMSE based on the decentralized architecture achieves negligible performance loss with less complexity cost compared to the centralized WMMSE.

REFERENCES

[1] S. S. Christensen, R. Agarwal, E. De Carvalho, and J. M. Cioffi, "Weighted sum-rate maximization using weighted MMSE for MIMO-BC beamforming design," *IEEE Trans. Wireless Commun.*, vol. 7, no. 12, pp. 4792–4799, 2008.

[2] Q. Shi, M. Razaviyayn, Z.-Q. Luo, and C. He, "An Iteratively Weighted MMSE Approach to Distributed Sum-Utility Maximization for a MIMO Interfering Broadcast Channel," *IEEE Trans. Signal Process.*, vol. 59, no. 9, pp. 4331–4340, 2011.

[3] H. Sun, X. Chen, Q. Shi, M. Hong, X. Fu, and N. D. Sidiropoulos, "Learning to optimize: Training Deep Neural Networks for interference management," *IEEE Trans. Signal Process.*, vol. 66, no. 20, pp. 5438–5453, 2018.

[4] H. Huang, Y. Peng, J. Yang, W. Xia, and G. Gui, "Fast beamforming design via deep learning," *IEEE Trans. Veh. Technol.*, vol. 69, no. 1, pp. 1065–1069, 2020.

[5] L. Pellaco, M. Bengtsson, and J. Jaldn, "Deep weighted MMSE downlink beamforming," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, 2021, pp. 4915–4919.

[6] Q. Hu, Y. Cai, Q. Shi, K. Xu, G. Yu, and Z. Ding, "Iterative algorithm induced deep-unfolding neural networks: Precoding design for multiuser MIMO systems," *IEEE Trans. Wireless Commun.*, vol. 20, no. 2, pp. 1394–1410, 2021.

[7] M. Wu, Z. Gao, Y. Huang, Z. Xiao, D. W. K. Ng, and Z. Zhang, "Deep Learning-Based Rate-Splitting Multiple Access for Reconfigurable Intelligent Surface-Aided Tera-Hertz Massive MIMO," *IEEE Journal on Selected Areas in Communications*, vol. 41, no. 5, pp. 1431–1451, 2023.

[8] L. Ning and F. Pingzhi, "Distributed Cell-Free Massive MIMO Versus Cellular Massive MIMO Under UE Hardware Impairments," *Chinese Journal of Electronics*, vol. 33, no. 5, pp. 1274–1285, 2024.

[9] D. Wang, M. Tao, X. Zeng, and J. Liang, "Federated learning for precoding design in cell-free massive MIMO systems," *IEEE Open J. Commun. Soc.*, vol. 4, p. 15671582, 2023.

[10] R. Wang, W. Dai, and Y. Jiang, "Distributed learning for uplink cell-free Massive MIMO networks," *IEEE Trans. Commun.*, vol. 71, no. 9, p. 55955606, Sep. 2023.

[11] L. Grippo and M. Sciandrone, "On the convergence of the block nonlinear Gauss–Seidel method under convex constraints," *Operations research letters*, vol. 26, no. 3, pp. 127–136, 2000.

[12] N. Zhou, Z. Wang, C. Ma, Y. Huang, and Q. Shi, "Distributed Precoder Based on Weighted MMSE With Low Complexity for Massive MIMO Systems," *IEEE Communications Letters*, vol. 29, no. 3, pp. 482–486, 2025.

[13] X. Zhao, S. Lu, Q. Shi, and Z.-Q. Luo, "Rethinking WMMSE: Can Its Complexity Scale Linearly With the Number of BS Antennas?" *IEEE Trans. Signal Process.*, vol. 71, pp. 433–446, 2023.