2022 IEEE the 14th International Conference on Wireless Communications and Signal Processing

A New Low-Complexity WMMSE Algorithm for Downlink Massive MIMO Systems

Ningxin Zhou and Zheng Wang
School of Information Science and Engineering
Southeast University
Nanjing, China
ningxin_zhou@seu.edu.cn, z.wang@ieee.org

Lanxin He and Yang Huang
College of Electronic and Information Engineering
Nanjing University of Aeronautics and Astronautics
Nanjing, China
{lanxin_he, yang.huang.ceie}@nuaa.edu.cn

2022 14th International Conference on Wireless Communications and Signal Processing (WCSP) | 978-1-6654-5085-0/22/$31.00 ©2022 IEEE | DOI: 10.1109/WCSP55476.2022.10039088

***Abstract*—Precoding is one of the most important technologies for downlink massive multiple-input multiple-output (MIMO) systems. Although the popular weighted minimum mean-square error (WMMSE) algorithm is guaranteed to converge to at least a local optimum of the weighted sum rate (WSR) maximization problem, its computational complexity is still too high due to the complicated matrix inversion. To this end, an iterative method based WMMSE algorithm (IM-WMMSE) is proposed in this paper. By adopting the traditional iterative methods to bypass the matrix inversion, the complexity of IM-WMMSE is significantly less than WMMSE with negligible performance loss. Meanwhile, inspired by deep learning, a deep neural network named as IM-WMMSE-Net is also designed for further complexity reduction. Simulation results demonstrate that IM-WMMSE is able to effectively reduce the computational complexity while the designed IM-WMMSE-Net achieves high performance with low computational complexity.**

***Index Terms*—Massive MIMO, precoding, weighted MMSE, iteration methods, deep unfolding neural network.**

I. Introduction

Massive multiple-input multiple-output (MIMO) has attracted great interest since it can dramatically increase the channel capacity without extra bandwidth. In massive MIMO systems, precoding plays an important role in determining the downlink performance. To maximize the weighted sum rate (WSR), a number of advanced precoding algorithms based on optimization methods have been proposed [1]–[3]. In [3], the weighted minimum mean-square error (WMMSE) algorithm is proposed, which is able to converge to at least a local optimum of WSR maximization problem. However, due to the complicated matrix inversion, it is rather challenging to widely implement WMMSE in practice, especially in high-dimensional systems. For this reason, a lot of effort has been made to reduce the computational complexity of WMMSE.

Specifically, the work in [4] proposes a FAST-WMMSE algorithm. By exploiting the relationship between the precoding matrix and channel matrix, it has fewer iterations than WMMSE. In [5], a block diagonal zero forcing (BD-ZF) beamforming algorithm is proposed, where the number of iterations can be dramatically reduced with BD-ZF initialization. A variant of WMMSE algorithm called R-WMMSE algorithm is also proposed in [6]. R-WMMSE uses the low-dimensional subspace property to reduce the matrix operation dimension, thus leading to a much less computational complexity than WMMSE. On the other hand, many studies have applied deep learning to reduce the complexity of WMMSE. The numerical results in [7] show that deep neural network (DNN) is able to significantly reduce the calculation time of WMMSE. Moreover, one of the model-driven deep learning methods known as deep unfolding is also applied in WMMSE [8]–[10], where different variants of WMMSE with trainable parameters are unfolded into a layer-wise structure. Both [8] and [9] use first-order Taylor expansion with several trainable matrices to approximate the matrix inversion. But the latter proposes a generalized chain rule (GCR) to depict the recurrence relation between two adjacent layers in Back Propagation in order to train the network. Besides, the authors in [10] also apply the projected gradient descent (PGD) with trainable step sizes to avoid the matrix inversion.

In this paper, to achieve a lower complexity for precoding, iterative method based WMMSE (IM-WMMSE) algorithm is proposed by adopting the classic iterative methods into WMMSE for matrix inversion approximation. Then, we also design a deep neural network named as IM-WMMSE-Net, which unfolds the proposed IM-WMMSE in a deep learning way. Although both IM-WMMSE and IM-WMMSE-Net are described based on the Gauss-Seidel (GS) iteration, they are well suited to various low-complexity iteration methods, making them flexible to different cases of massive MIMO systems.

II. System Model

Considering a downlink massive MIMO system with K cells, the base station (BS) in cell k is equipped by M_k transmit antennas to serve I_k users. We define i_k as the ith user equipped with N_{i_k} receive antennas in cell k. The set of all users in the whole system is defined as $\mathcal{I}$. Denote $\mathbf{v}_{i_k} \in \mathbb{C}^{M_k}$ as the precoding vector with the transmitted signal $s_{i_k} \sim \mathcal{CN}(0,1)$ to user i_k, then the signal sent at BS k can be expressed as

$$\mathbf{x}_k = \sum_{i=1}^{I_k} \mathbf{v}_{i_k} s_{i_k}. \tag{1}$$

The precoding vector must satisfy power constraint $\sum_{i_k=1}^{I_k} \mathrm{Tr}(\mathbf{v}_{i_k}\mathbf{v}_{i_k}^H) \leq P_k$, where $\mathrm{Tr}(\cdot)$ and P_k represent the trace operator and the transmit power budget at the BS k, respectively.

Corresponding author: Zheng Wang (e-mail: z.wang@ieee.org)

Given the transmitted signal in (1), then the received signal $\mathbf{y}_{i_k} \in \mathbb{C}^{N_{i_k}}$ at receiver i_k turns out to be

$$\mathbf{y}_{i_k} = \mathbf{H}_{i_k k}\mathbf{v}_{i_k}s_{i_k} + \sum_{m=1,m\neq i}^{I_k} \mathbf{H}_{i_k k}\mathbf{v}_{m_k}s_{m_k} + \sum_{j\neq k,j=1}^{K}\sum_{\ell=1}^{I_j}\mathbf{H}_{i_k j}\mathbf{v}_{j_l}s_{j_l} + \mathbf{n}_{i_k}. \tag{2}$$

Here, $\mathbf{H}_{i_k j} \in \mathbb{C}^{N_{i_k}\times M_j}$ denotes the massive MIMO channel matrix from the BS j to user i_k, $\mathbf{n}_{i_k} \in \mathbb{C}^{N_{i_k}}$ represents the additive white Gaussian noise (AWGN) with $\mathcal{CN}(0,\sigma_{i_k}^2\mathbf{I})$, and $\sigma_{i_k}^2$ is the average noise power at user i_k.

Theoretically, precoding aims to find the set of all precoding vectors[1] $\mathbf{V}$ which maximizes the weighted sum rate (WSR) as follows

$$\begin{aligned}&\max_{\{\mathbf{V}\}}\sum_{k=1}^{K}\sum_{i_k=1}^{I_k}\alpha_{i_k}R_{i_k}\\ &\text{s.t.}\sum_{i_k=1}^{I_k}\mathrm{Tr}(\mathbf{v}_{i_k}\mathbf{v}_{i_k}^H)\leq P_k, k=1,2,...,K.\end{aligned} \tag{3}$$

Here, α_{i_k} stands for the priority of user i_k in the system, and R_{i_k} is the rate of user i_k written as

$$R_{i_k} = \log\det\left(\mathbf{I}+\mathbf{H}_{i_k k}\mathbf{v}_{i_k}\mathbf{v}_{i_k}^H\mathbf{H}_{i_k k}^H\left(\sum_{(l,j)\neq(i,k)}\mathbf{H}_{i_k j}\mathbf{v}_{l_j}\mathbf{v}_{l_j}^H\mathbf{H}_{i_k j}^H+\sigma_k^2\mathbf{I}\right)^{-1}\right). \tag{4}$$

However, because WSR maximization problem in (3) is non-convex, it is difficult to solve especially in high-dimensional systems. To this end, an effective alternative named as WMMSE is given, which solves the equivalent problem as [3]

$$\begin{aligned}&\min_{\{\mathbf{V},\mathbf{W},\mathbf{U}\}}\sum_{k=1}^{K}\sum_{i_k=1}^{I_k}\alpha_{i_k}\left(w_{i_k}E_{i_k}-\log(w_{i_k})\right)\\ &\text{s.t.}\sum_{i_k=1}^{I_k}\mathrm{Tr}(\mathbf{v}_{i_k}\mathbf{v}_{i_k}^H)\leq P_k, k=1,2,...,K.\end{aligned} \tag{5}$$

Here, w_{i_k} and $\mathbf{u}_{i_k} \in \mathbb{C}^{N_{i_k}}$ are auxiliary variables, which represent the positive weight variable and the receiving vector of receiver i_k. The mean squared error (MSE) E_{i_k} is given by

$$E_{i_k} = (1-\mathbf{u}_{i_k}^H\mathbf{H}_{i_k k}\mathbf{v}_{i_k})(1-\mathbf{u}_{i_k}^H\mathbf{H}_{i_k k}\mathbf{v}_{i_k}^H) + \sum_{(l,j)\neq(i,k)}\mathbf{u}_{i_k}\mathbf{H}_{i_k j}\mathbf{v}_{l_j}\mathbf{v}_{l_j}^H\mathbf{H}_{i_k j}^H\mathbf{u}_{i_k}^H+\sigma_{i_k}^2\mathbf{u}_{i_k}^H\mathbf{u}_{i_k}. \tag{6}$$

Intuitively, since the problem in (5) is convex in each optimization variable of $\mathbf{U},\mathbf{W},\mathbf{V}$, the block coordinate descent (BCD) method is applied by WMMSE, which sequentially updates each one of $\mathbf{U},\mathbf{W},\mathbf{V}$ by fixing the other two. More specifically, as shown in Algorithm 1, $\mu_k \geq 0$ in step 6 is a Lagrange multiplier to make the precoding vector $\mathbf{v}_{i_k}$ satisfy the power constraint, where bisection search is applied to search it [3], and ε is the tolerance of accuracy. However, due to the operations of matrix inversion in updating $\mathbf{U}$ and $\mathbf{V}$, the implementation of WMMSE is still challenging in practice.

[1]The notation $\mathbf{V}$ is short for $\{\mathbf{v}_{i_k}\}_{i_k\in\mathcal{I}}$, $\mathbf{U}=\{\mathbf{u}_{i_k}\}_{i_k\in\mathcal{I}}$, $\mathbf{W}=\{w_{i_k}\}_{i_k\in\mathcal{I}}$, $\mathbf{H}=\{\mathbf{H}_{i_k j}\}_{i_k\in\mathcal{I},j\in\mathcal{K}}$, $\mathbf{J}=\{\mathbf{J}_k\}_{k\in\mathcal{K}}$ and $\mathbf{C}=\{\mathbf{c}_{i_k}\}_{i_k\in\mathcal{I}}$.

Algorithm 1 WMMSE Algorithm

Input: $\mathbf{H}$, ε, $\{\alpha_{i_k}\}_{i_k\in\mathcal{I}}$, $\{\sigma_{i_k}\}_{i_k\in\mathcal{I}}$, $\{P_k\}_{k\in\mathcal{K}}$.
Output: $\mathbf{u}_{i_k}$,w_{i_k},$\mathbf{v}_{i_k}$, $\forall i_k$.
1: **Initialize:** set $\mathbf{v}_{i_k}$ to satisfy $\mathrm{Tr}(\mathbf{v}_{i_k}\mathbf{v}_{i_k}^H)=P_k/I_k$, and $\mathbf{u}_{i_k}=\mathbf{0}$ and $w_{i_k}=0$, $\forall i_k$.
2: **repeat**
3: $w'_{i_k}=w_{i_k}$, $\forall i_k$
4: $\mathbf{u}_{i_k}=(\sum_{(l,j)}\mathbf{H}_{i_k j}\mathbf{v}_{l_j}\mathbf{v}_{l_j}^H\mathbf{H}_{i_k j}^H+\sigma_{i_k}^2\mathbf{I})^{-1}\mathbf{H}_{i_k k}\mathbf{v}_{i_k}$, $\forall i_k$
5: $w_{i_k}=(1-\mathbf{u}_{i_k}^H\mathbf{H}_{i_k k}\mathbf{v}_{i_k})^{-1}$, $\forall i_k$
6: $\mathbf{v}_{i_k}=\alpha_{i_k}(\sum_{(l,j)}\alpha_{l_j}w_{l_j}\mathbf{H}_{l_j k}^H\mathbf{u}_{l_j}\mathbf{u}_{l_j}^H\mathbf{H}_{l_j k}+\mu_k\mathbf{I})^{-1}w_{i_k}\mathbf{H}_{i_k k}^H\mathbf{u}_{i_k}$, $\forall i_k$
7: **until** $|\sum_{(l,j)}\log(w_{l_j})-\sum_{(l,j)}\log(w'_{l_j})|<\varepsilon$

III. Iteration Method Based WMMSE

In this section, the classic low-complex iterative methods are introduced into WMMSE for matrix inversion approximation, which leads to the proposed iterative method based WMMSE (IM-WMMSE).

A. Iterative Methods for Linear Systems

Consider a linear equation as follows

$$\mathbf{A}\mathbf{z}=\mathbf{b}, \tag{7}$$

where $\mathbf{A}\in\mathbb{C}^{m\times m}$ is an invertible matrix, $\mathbf{b}\in\mathbb{C}^m$. Clearly, $\mathbf{z}=\mathbf{A}^{-1}\mathbf{b}\in\mathbb{C}^m$ is the exact solution of (7). However, due to the matrix inversion, the complexity of computing $\mathbf{z}$ is $\mathcal{O}(m^3)$, which is unfavorable in practice, especially when m is large enough.

To bypass the matrix inversion, many traditional iterative methods are proposed [11]. By dividing $\mathbf{A}$ into $\mathbf{G}\in\mathbb{C}^{m\times m}$ and $\mathbf{F}\in\mathbb{C}^{m\times m}$, i.e. $\mathbf{A}=\mathbf{G}+\mathbf{F}$, the general form of the iterative expression of $\mathbf{z}$ is as follows

$$\mathbf{z}^{n+1}=\mathbf{M}\mathbf{z}^n+\mathbf{N}\mathbf{b}. \tag{8}$$

Here, $\mathbf{z}^n\in\mathbb{C}^m$ represents the estimated vector of $\mathbf{z}$ at the nth iteration , $\mathbf{z}^0$ is the initial vector which is usually set as zero vector, the matrix $\mathbf{M}=-\mathbf{G}^{-1}\mathbf{F}\in\mathbb{C}^{m\times m}$ is the iterative matrix which determines the convergence of the iterative method, and the matrix $\mathbf{N}=\mathbf{G}^{-1}\in\mathbb{C}^{m\times m}$. In theory, when the spectral radius of $\mathbf{M}$ satisfies $\rho(\mathbf{M})<1$, the iterative method is convergent, and the convergence speed is inversely proportional to the spectral radius [11].

Apparently, different choices of $\mathbf{M}$ and $\mathbf{N}$ produce different iterative methods. For Jacobi iteration, the values of $\mathbf{M}$ and $\mathbf{N}$ are set as $\mathbf{M}=-\mathbf{D}^{-1}(\mathbf{L}+\mathbf{T}),\mathbf{N}=\mathbf{D}^{-1}$, which leads to the following iteration

$$\mathbf{z}^{n+1}=\mathbf{D}^{-1}(\mathbf{b}-(\mathbf{L}+\mathbf{T})\mathbf{z}^n). \tag{9}$$

Here, $\mathbf{D}\in\mathbb{C}^{m\times m}$, $\mathbf{L}\in\mathbb{C}^{m\times m}$ and $\mathbf{T}\in\mathbb{C}^{m\times m}$ represent the diagonal, the strict lower and the strict upper matrices of $\mathbf{A}$, respectively. Next, when $\mathbf{M}=-(\mathbf{D}+\mathbf{L})^{-1}\mathbf{T}$ and $\mathbf{N}=$

TABLE I
THE PROPORTION OF COMPLEXITY REQUIRED TO UPDATE EACH VECTOR IN DIFFERENT SYSTEMS

$M_k \times N_{i_k}$	U	W	V
4×4	40.75%	0.78%	58.47%
16×4	18.42%	0.35%	81.22%
32×4	9.03%	0.17%	90.80%
64×4	3.54%	0.07%	96.40%

$(\mathbf{D}+\mathbf{L})^{-1}$, the iterative method is known as Gauss-Seidel (GS) iteration with

$$\mathbf{z}^{n+1} = (\mathbf{D}+\mathbf{L})^{-1}(\mathbf{b}-\mathbf{T}\mathbf{z}^n). \tag{10}$$

In addition, based on GS iteration, successive overrelaxation (SOR) iteration is proposed, and its corresponding iterative expression is as follows

$$\mathbf{z}^{n+1} = \left(\frac{1}{\omega}\mathbf{D}+\mathbf{L}\right)^{-1}\left(\mathbf{b}+\left(\left(\frac{1}{\omega}-1\right)\mathbf{D}-\mathbf{T}\right)\mathbf{z}^n\right). \tag{11}$$

Here, $0 < \omega < 2$ is the relaxation factor, which affects the convergence of SOR. It is obvious that SOR is equivalent to GS when $\omega = 1$.

Among these iterative methods, Jacobi iteration only works when $\mathbf{A}$ is diagonally dominant, where the spectral radius of $\mathbf{M}$ is less than 1 [12]. Different from Jacobi iteration, GS is always convergent. As for SOR, the convergence is guaranteed when the relaxation factor ω satisfies $0 < \omega < 2$ [13]. Moreover, due to a smaller spectral radius, GS iteration has a better convergence performance than Jacobi iteration [14]. For SOR, when the relaxation factor ω takes the optimal value $\omega_{\text{opt}} = 2/(1+\sqrt{1-\lambda_{\max}^2})$ ($\lambda_{\max}$ is the maximum eigenvalue of $(\mathbf{A}-\mathbf{D})$), the convergence speed is faster than that of GS [15]. On the contrary, it is also clear to see that Jacobi iteration has the lowest computational cost among them while SOR iteration requires the highest complexity.

B. Iterative Method Based WMMSE

Clearly, the updates of $\mathbf{u}_{i_k}$ and $\mathbf{v}_{i_k}$ in WMMSE both involve matrix inversions, where the dimensions of matrix inversion are N_{i_k} and M_k respectively. To make it more specific, the proportion of complexity required to update each of $\mathbf{U}, \mathbf{W}$ and $\mathbf{V}$ under different system dimensions is shown in Table I with $I_k = 10, \forall k$. Apparently, the complexity of computing $\mathbf{v}_{i_k}$ is dominant, compared to those of $\mathbf{u}_{i_k}$ and w_{i_k}. Meanwhile, because WMMSE is an iterative algorithm, the approximation error will be accumulated iteratively if the matrix inversion approximation is applied. Therefore, to balance the trade-off between complexity reduction and error propagation, we only apply the iterative methods to update the vector $\mathbf{v}_{i_k}$.

Typically, regarding to the step 6 in Algorithm 1, we define

$$\mathbf{J}_k = \sum_{(l,j)} \alpha_{l_j} w_{l_j} \mathbf{H}_{l_j k}^H \mathbf{u}_{l_j} \mathbf{u}_{l_j}^H \mathbf{H}_{l_j k} + \mu_k \mathbf{I} \in \mathbb{C}^{M_k \times M_k}, \tag{12}$$

$$\mathbf{c}_{i_k} = \alpha_{i_k} w_{i_k} \mathbf{H}_{i_k k}^H \mathbf{u}_{i_k} \in \mathbb{C}^{M_k}, \tag{13}$$

where the Lagrange multiplier μ_k is still searched by bisection search. Apparently, $\mathbf{v}_{i_k}$ is the exact solution of $\mathbf{J}_k \mathbf{v}_{i_k} = \mathbf{c}_{i_k}$, which has the same formation with $\mathbf{z}$ in (7). Therefore, the iterative methods for solving (7) can be applied to get the precoding vector $\mathbf{v}_{i_k}$. Here, for simplicity, we describe the proposed IM-WMMSE algorithm based on GS iteration, where the extensions to other iterative methods are straightforward.

Algorithm 2 IM-WMMSE Algorithm

Input: $\mathbf{H}$, ε, Q, $\{\alpha_{i_k}\}_{i_k \in \mathcal{I}}$, $\{\sigma_{i_k}\}_{i_k \in \mathcal{I}}$, $\{P_k\}_{k \in \mathcal{K}}$.
Output: $\mathbf{u}_{i_k}$,w_{i_k},$\mathbf{v}_{i_k}$, $\forall i_k$.

Initialize: set $\mathbf{v}_{i_k}$ to satisfy $\text{Tr}(\mathbf{v}_{i_k}\mathbf{v}_{i_k}^H) = P_k/I_k$, and $\mathbf{u}_{i_k} = \mathbf{0}$ and $w_{i_k} = \mathbf{0}$, $\forall i_k$.
repeat
  $w'_{i_k} = w_{i_k}$, $\forall i_k$
  $\mathbf{u}_{i_k} = (\sum_{(l,j)} \mathbf{H}_{i_k j}\mathbf{v}_{l_j}\mathbf{v}_{l_j}^H\mathbf{H}_{i_k j}^H + \sigma_{i_k}^2\mathbf{I})^{-1}\mathbf{H}_{i_k k}\mathbf{v}_{i_k}$, $\forall i_k$
  $w_{i_k} = (1 - \mathbf{u}_{i_k}^H\mathbf{H}_{i_k k}\mathbf{v}_{i_k})^{-1}$, $\forall i_k$
  $\mathbf{J}_k = \sum_{(l,j)} \alpha_{l_j} w_{l_j} \mathbf{H}_{l_j k}^H \mathbf{u}_{l_j} \mathbf{u}_{l_j}^H \mathbf{H}_{l_j k} + \mu_k \mathbf{I}$, $\forall k$
  $\mathbf{c}_{i_k} = \alpha_{i_k} w_{i_k} \mathbf{H}_{i_k k}^H \mathbf{u}_{i_k}$, $\forall i_k$
  Divide $\mathbf{J}_k$ into $\mathbf{E}_k$, $\mathbf{P}_k$ and $\mathbf{P}_k^H$, $\forall i_k$
  $\mathbf{v}_{i_k}^0 = \mathbf{0}$, $\forall i_k$
  for $q = 1, 2, ..., Q$ **do**
    $\mathbf{v}_{i_k}^q = (\mathbf{E}_k + \mathbf{P}_k)^{-1}(\mathbf{c}_{i_k} - \mathbf{P}_k^H \mathbf{v}_{i_k}^{q-1})$, $\forall i_k$
  end for
  $\mathbf{v}_{i_k} = \Pi\{\mathbf{v}_{i_k}^Q\}$, $\forall i_k$
until $|\sum_{(l,j)} \log(w_{l_j}) - \sum_{(l,j)} \log(w'_{l_j})| < \varepsilon$

According to GS iteration, $\mathbf{J}_k$ is divided by diagonal matrix $\mathbf{E}_k$, strictly lower triangular matrix $\mathbf{P}_k$ and strictly upper triangular matrix $\mathbf{P}_k^H$. Thus, we can update $\mathbf{v}_{i_k}$ as follows

$$\mathbf{v}_{i_k}^q = (\mathbf{E}_k + \mathbf{P}_k)^{-1}(\mathbf{c}_{i_k} - \mathbf{P}_k^H \mathbf{v}_{i_k}^{q-1}), \tag{14}$$

where $\mathbf{v}_{i_k}^q \in \mathbb{C}^{M_k}$ represents the precoding vector $\mathbf{v}_{i_k}$ at the qth iteration and the initial vector $\mathbf{v}_{i_k}^0$ is set as zero vector.

Then, due to the approximation error in iterative methods, the approximated precoding vector $\mathbf{v}_{i_k}^Q$ obtained by GS iteration needs to be normalized to avoid exceeding the transmission power budget P_k. Therefore, we have

$$\mathbf{v}_{i_k} = \Pi\{\mathbf{v}_{i_k}^Q\} = \frac{\mathbf{v}_{i_k}^Q}{\sum_{i_k=1}^{I_k} \text{Tr}(\mathbf{v}_{i_k}^Q \mathbf{v}_{i_k}^{Q H})}\sqrt{P_k}, \tag{15}$$

where the function $\Pi\{\cdot\}$ represents the normalization operation. In this way, the whole algorithm iterates until satisfying the convergence criteria. To sum up, the proposed IM-WMMSE algorithm is outlined in Algorithm 2.

Now, we consider the computational complexity of proposed IM-WMMSE. Denote $I = |\mathcal{I}|$ as the total number of users in the system. The total complexity of computing $\mathbf{U}$ and $\mathbf{W}$ is $\mathcal{O}(I^2 M_k N_{i_k}^2 + I M_k N_{i_k}^2 + I N_{i_k}^3)$, while that of computing $\mathbf{J}$ and $\mathbf{C}$ is $\mathcal{O}(IK M_k^2 N_{i_k} + I M_k^2 N_{i_k})$. When GS iteration or Jacobi iteration is applied to computing $\mathbf{V}$, the complexity is $\mathcal{O}(IQM_k^2)$, and it is $\mathcal{O}(IQM_k^2 + IQM_k)$ when using SOR iteration. Beside, the complexity of computing (15) is $\mathcal{O}(IM_k^2)$. With respect to WMMSE, the complexities of computing $\mathbf{U}$, $\mathbf{W}$, $\mathbf{J}$ and $\mathbf{C}$ are same as IM-WMMSE, but the complexity of matrix inversion used to update $\mathbf{V}$ in WMMSE is $\mathcal{O}(IM_k^3)$.

Overall, the computational complexity of proposed IM-

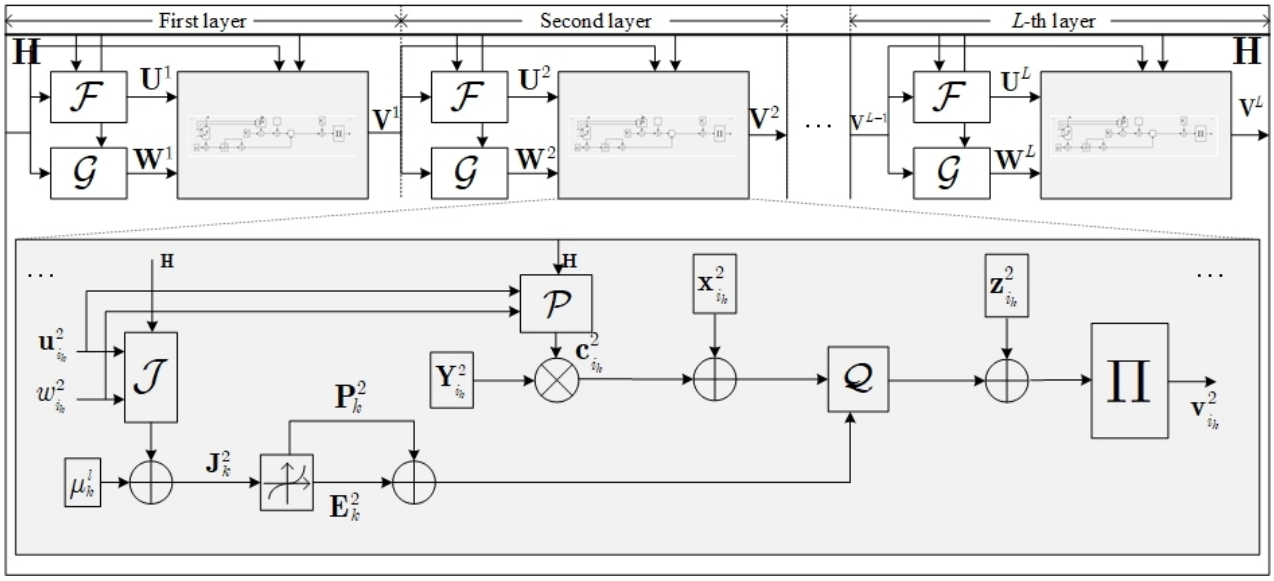

Fig. 1. Network architecture of the L layer IM-WMMSE-Net (The superscript l represents the parameters of the l-th layer, $\mathcal{F}$, $\mathcal{G}$, $\mathcal{J}$ and $\mathcal{P}$ denote the updating of $\mathbf{u}_{i_k}$, w_{i_k}, $\mathbf{J}_k$ and $\mathbf{c}_{i_k}$ in Algorithm 2 respectively, Π is the function in (15)), and $\mathcal{Q}$ means multiplying the inverse matrix of the square matrix and the vector.

WMMSE is $\mathcal{O}(L_a(I^2M_kN_{i_k}^2 + IKM_k^2N_{i_k} + IM_k^2N_{i_k} + IM_kN_{i_k}^2 + IN_{i_k}^3 + IQM_k^2 + IM_k^2))$, where L_a denotes the number of iterations. At the same time, the computational complexity of WMMSE is $\mathcal{O}(L_w(I^2M_kN_{i_k}^2 + IKM_k^2N_{i_k} + IM_k^2N_{i_k} + IM_kN_{i_k}^2 + IN_{i_k}^3 + IM_k^3))$, where L_w is the iteration number of WMMSE. Apparently, the number of transmitting antennas M_k is much larger than other system parameters N_{i_k}, I and K in massive MIMO systems. So the complexity of computing $\mathbf{V}$ accounts for the majority of the total, which is accordance with the results shown in Table I. Compared to WMMSE, IM-WMMSE reduces the complexity of updating $\mathbf{V}$ from $\mathcal{O}(IM_k^3)$ to $\mathcal{O}(IQM_k^2 + IM_k^2)$. Therefore, IM-WMMSE has much lower computational complexity than WMMSE especially when M_k is large enough.

IV. Deep Unfolding IM-WMMSE

Recently, a model-driven deep learning method called deep unfolding has attracted lots of research attentions, which unfolds the iteration algorithm via the layers of neural network. Inspired by deep unfolding, we design a deep neural network named as IM-WMMSE-Net for the further complexity reduction, where the iterative processes of IM-WMMSE is unfolded with some trainable parameters.

In particular, the designed IM-WMMSE-Net has L layers while each layer has the same structure. At each layer of IM-WMMSE-Net, the same steps in Algorithm 2 are used to update the $\mathbf{u}_{i_k}, w_{i_k}, \mathbf{J}_k$ and $\mathbf{c}_{i_k}$.

In IM-WMMSE, we use GS iteration (or other iterative methods) to solve $\mathbf{J}_k\mathbf{v}_{i_k} = \mathbf{c}_{i_k}$. However, if each layer has Q times iterations to bypass the matrix inversion, the network structure will be rather complicated. This makes us update the precoding matrix $\mathbf{v}_{i_k}$ in the following way

$$\mathbf{v}_{i_k} = (\mathbf{E}_k + \mathbf{P}_k)^{-1}(\mathbf{Y}_{i_k}\mathbf{c}_{i_k} + \mathbf{x}_{i_k}) + \mathbf{z}_{i_k}, \tag{16}$$

where diagonal matrix $\mathbf{Y}_{i_k} \in \mathbb{C}^{M_k \times M_k}$, $\mathbf{x}_k \in \mathbb{C}^{M_k}$ and $\mathbf{z}_{i_k} \in \mathbb{C}^{M_k}$ are trainable parameters in the network. When these trainable parameters are well-trained, the operations in steps 9-12 of Algorithm 2 can be achieved by the computation of $\mathbf{v}_{i_k}$ in (16). Here, we point out that using (16) instead of GS iteration in (14) to update $\mathbf{v}_{i_k}$ makes the complexity reduction from $\mathcal{O}(QM_k^2)$ to $\mathcal{O}(M_k^2 + M_k)$.

After (16), the function $\Pi\{\cdot\}$ in (15) is still applied to make $\mathbf{v}_{i_k}$ satisfy power constraint. Moreover, to further reduce the complexity and simplify the neural network, the optimal Lagrangian multiplier μ_k is also chosen as the trainable parameter. To summarize, the trainable parameters of layer l in IM-WMMSE-Net are $\mathbf{Y}^l = \{\mathbf{Y}_{i_k}^l | i_k \in \mathcal{I}\}$, $\mu^l = \{\mu_k^l | k \in \mathcal{K}\}$, $\mathbf{X}^l = \{\mathbf{x}_{i_k}^l | i_k \in \mathcal{I}\}$ and $\mathbf{z}^l = \{\mathbf{z}_{i_k}^l | i_k \in \mathcal{I}\}$. For a better understanding, the L-layer IM-WMMSE-Net network is depicted in Figure 1 in detail.

On the other hand, because IM-WMMSE-Net aims to find the precoding vector to maximize WSR, the WSR of the precoding vector outputted from the last layer can be directly selected as the loss function of the network, i.e.,

$$\text{Loss} = -\frac{1}{N_s}\sum_{n=1}^{N_s} f_{\text{WSR}}\left(\mathbf{H}_n, \mathbf{V}_n^L\right). \tag{17}$$

Here, $f_{\text{WSR}}\left(\mathbf{H}_n, \mathbf{V}_n^L\right)$ stands for the corresponding WSR of the output of the Lth iteration, subscript n represents the nth training data, and N_s denotes batch size. Besides, the Adam optimizer is applied to minimize (17) in the back propagation.

Considering the complexity of the designed IM-WMMSE-Net, the complexity of updating $\mathbf{V}$ is reduced from $\mathcal{O}(IM_k^3)$ to $\mathcal{O}(IM_k^2 + IM_k)$ by adopting the matrix multiplication in (16) to replace the matrix inversion. Meanwhile, the step of searching Lagrange multiplier μ_k, which involves bisection search, eigen-decomposition and other operations, is also removed, which results in further complexity reduction.

To summarize, the overall complexity of IM-WMMSE-Net is given by $\mathcal{O}(L(I^2M_kN_{i_k}^2 + IKM_k^2N_{i_k} + IM_k^2N_{i_k} + IM_kN_{i_k}^2 + IN_{i_k}^3 + IM_k^2))$. With respect to IM-WMMSE-Net,

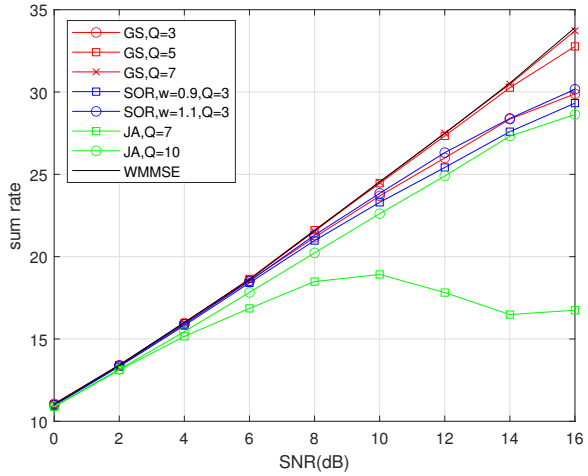

Fig. 2. sum rate versus average SNR for $M_k \times N_{i_k} = 4 \times 4$ MIMO system with 2 cells. ($K = 2$, $I_k = 10$, $\varepsilon = 0.01$)

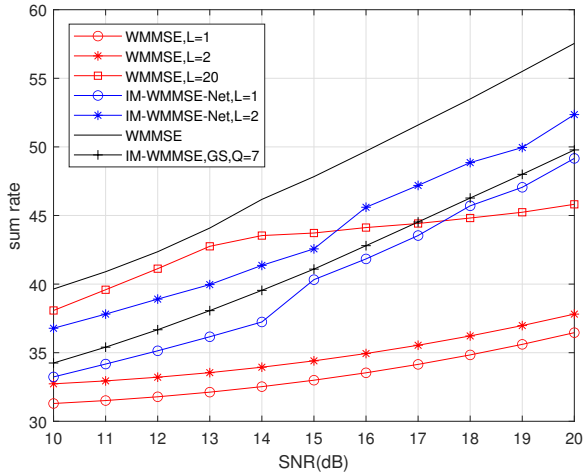

Fig. 3. sum rate versus average SNR for $M_k \times N_{i_k} = 64 \times 16$ MIMO system. ($K = 1$, $I_k = 10$, $\varepsilon = 0.01$, the legend WMMSE,L=l represents the result of only iterating WMMSE algorithm for l times)

the complexity of computing $\mathbf{V}$ is less than WMMSE. Beside, due to the introduced trainable parameters the numbers of layers is far less than the number of iterations required for WMMSE algorithm, i.e, $L \ll L_a, L_w$. In general, the overall complexity of IM-WMMSE-Net is less than WMMSE and proposed IM-WMMSE.

V. Simulations

In this section, we compare the simulation performance of the proposed IM-WMMSE based on different iterative methods, where the performance of the designed IM-WMMSE-Net is also shown. To simplify the system model, we assume that the base stations in different cells have the same number of antennas and serve the same number of users, i.e. $\forall k, j, M_k = M_j$ and $I_k = I_j$. Meanwhile, the receiving antennas for each user are the same, i.e.$\forall i_k, l_j, N_{i_k} = N_{l_j}$, and each user has the same priority, i.e. $\forall i_k, \alpha_{i_k} = 1$.

TABLE II
Sum rate and CPU times for different iteration methods and system models

$M_k \times N_{i_k}$	WMMSE		GS, $Q=3$		GS, $Q=5$		GS, $Q=7$	
	WSR	time	WSR	time	WSR	time	WSR	time
4×4	32.1742	4.3339	92.56%	68.62%	98.64%	87.48%	99.57%	102.55%
16×4	80.3982	12.6681	73.34%	24.97%	85.80%	32.82%	92.19%	57.07%
32×4	138.0862	22.1745	67.19%	16.75%	79.29%	22.88%	85.08%	18.86%
64×4	220.2730	34.6544	72.94%	20.54%	83.98%	39.92%	86.73%	51.39%
$M_k \times N_{i_k}$	SOR, $Q=5, \omega=0.9$		SOR, $Q=5, \omega=1.1$		JA,$Q=7$		JA, $Q=10$	
	WSR	time	WSR	time	WSR	time	WSR	time
4×4	97.43%	84.50%	99.06%	96.55%	63.82%	221.69%	91.69%	93.27%
16×4	84.15%	37.91%	86.22%	35.41%	-	-	-	-
32×4	78.30%	30.34%	79.47%	19.73%	-	-	-	-
64×4	84.17%	62.87%	82.48%	58.96%	-	-	-	-

Fig. 2 shows the performance comparison of different iterative methods based IM-WMMSE in model with $K = 2$, $I_k = 10$ and $M_k \times N_{i_k} = 4 \times 4$. It is clear to see that the higher Q leads to the higher sum rate because of smaller approximate error. When Q is same, the sum rate of IM-WMMSE based GS and SOR is close, i.e, the influence of relax factor ω is small, and the sum rate of IM-WMMSE based Jacobi iteration is much less than others due to the lower convergence speed. At the same time, the required Q in IM-WMMSE algorithm increases to get a close result of WMMSE as SNR increases.

Table II jointly compares the sum rate and CPU time for different iteration methods and system models with $K = 4$ and $I_k = 10$, where the results of WMMSE are shown as real value while others give the proportion compared with the results of WMMSE. Except for the case of 4×4, the results of Jacobi iteration-based IM-WMMSE is not given in the table, because the matrix $\mathbf{J}_k$ which is not diagonally dominant in other cases does not satisfy the convergence condition of Jacobi iteration. We can observe that both the sum rate and CPU time improve as Q increases, and a higher Q leads to excess CPU time. Therefore, choosing the appropriate Q plays an important role to make a better trade-off between performance and complexity. When $M_k = 64$, the CPU time of SOR-based IM-WMMSE is higher than that of GS-based IM-WMMSE with a small gap in sum rate. According to Table II, it can be found that GS is more suitable for IM-WMMSE among these iterative methods.

We also compare the sum rate of IM-WMMSE-Net, IM-WMMSE and WMMSE in $M_k \times N_{i_k} = 64 \times 16$ MIMO system with $K = 1$ and $I_k = 10$ in Fig. 3. We assume the batch size is 8 when $L = 1$, the batch size is 16 with $L = 2$, the number of training batches is 10000 and learning rate decreases exponentially. It can be found that, even if $L = 2$, the sum rate of IM-WMMSE-Net is close to WMMSE and greater than IM-WMMSE. Meanwhile, when SNR $= 20$dB, the sum rate is close between IM-WMMSE-Net with $L = 1$ and WMMSE after 20 iterations. Specifically, the CPU time of IM-WMMSE-Net with $L = 2$ is 1.34% of WMMSE and the sum rate of IM-WMMSE-Net is 90.98% of WMMSE, which illustrates that IM-WMMSE-Net has much less complexity than WMMSE.

VI. Conclusion

In this work, we propose an iterative method based WMMSE algorithm (IM-WMMSE), where the traditional it-

erative method is adopted to bypass the matrix inversion. Based on IM-WMMSE, a deep neural network named as IM-WMMSE-Net is also designed for further complexity reduction. Our results show that both the proposed IM-WMMSE and IM-WMMSE-Net are able to effectively reduce the computational complexity. Besides GS iteration, others low-complexity iterative methods are also suitable to them.

REFERENCES

[1] S.-J. Kim and G. B. Giannakis, "Optimal resource allocation for MIMO Ad Hoc cognitive radio networks," *IEEE Transactions on Information Theory*, vol. 57, no. 5, pp. 3117–3131, 2011.

[2] 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 Transactions on Wireless Communications*, vol. 7, no. 12, pp. 4792–4799, 2008.

[3] 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 Transactions on Signal Processing*, vol. 59, no. 9, pp. 4331–4340, 2011.

[4] K. Bao and Y. Lu, "A FAST-WMMSE approach to distributed sum-rate maximization for a MIMO-BC channel," in *2021 13th International Conference on Wireless Communications and Signal Processing (WCSP)*, 2021, pp. 1–5.

[5] S. Lu, S. Zhao, and Q. Shi, "Block-Diagonal Zero-Forcing beamforming for weighted sum-rate maximization in Multi-User Massive MIMO Systems," in *2020 IEEE Symposium on Computers and Communications (ISCC)*, 2020, pp. 1–6.

[6] X. Zhao, S. Lu, Q. Shi, and Z.-Q. Luo, "Rethinking WMMSE: Can Its Complexity Scale Linearly With the Number of BS Antennas?" arXiv, Tech. Rep. arXiv:2205.06225, May 2022. [Online]. Available: http://arxiv.org/abs/2205.06225

[7] 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 Transactions on Signal Processing*, vol. 66, no. 20, pp. 5438–5453, 2018.

[8] Q. Hu, Y. Liu, Y. Cai, G. Yu, and Z. Ding, "Joint deep reinforcement learning and unfolding: Beam selection and precoding for mmWave Multiuser MIMO with lens arrays," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 8, pp. 2289–2304, 2021.

[9] 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 Transactions on Wireless Communications*, vol. 20, no. 2, pp. 1394–1410, 2021.

[10] L. Pellaco, M. Bengtsson, and J. Jaldén, "Deep weighted MMSE downlink beamforming," in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 4915–4919.

[11] Y. Saad, *Iterative methods for sparse linear systems*. SIAM, 2003.

[12] B. Y. Kong and I.-C. Park, "Low-complexity symbol detection for massive MIMO uplink based on Jacobi method," in *2016 IEEE 27th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, 2016, pp. 1–5.

[13] X. Gao, L. Dai, Y. Hu, Z. Wang, and Z. Wang, "Matrix inversion-less signal detection using SOR method for uplink large-scale MIMO systems," in *2014 IEEE Global Communications Conference*, 2014, pp. 3291–3295.

[14] M. A. Albreem, W. Salah, A. Kumar, M. H. Alsharif, A. H. Rambe, M. Jusoh, and A. N. Uwaechia, "Low complexity linear detectors for Massive MIMO: A Comparative Study," *IEEE Access*, vol. 9, pp. 45 740–45 753, 2021.

[15] B. A. Carre, "The Determination of the Optimum Accelerating Factor for Successive Over-relaxation," vol. 4, no. 1, pp. 73–78.