SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

August 2018, Vol. 61 082306:1–082306:13
https://doi.org/10.1007/s11432-017-9275-y

Precoder design in downlink CoMP-JT MIMO network via WMMSE and asynchronous ADMM

Zhikun WU & Zesong FEI*

School of Information Science and Engineering, Beijing Institute of Technology, Beijing 100081, China

Received 1 June 2017/Revised 1 September 2017/Accepted 20 October 2017/Published online 24 May 2018

Abstract Precoder design of coordinated multi-point joint transmission (CoMP-JT) multiple-input and multiple-output (MIMO) network aimied at throughput maximization is a challenging problem. In this paper, we propose an asynchronous distributed iterative method to solve this problem. We transform the original throughput maximizing problem to the weighted minimum mean square error (WMMSE) problem, then decompose the problem into a series of subproblems. Based on alternation direction method of multipliers (ADMM), the proposed algorithm can solve the optimal precoder in a distributed manner. With asynchronous information exchange mechanism considered, the convergence rate of our algorithm can be accelerated further. Numerical results demonstrate the increase of throughput and the optimality of the precoding scheme provided by our algorithm.

Keywords multiple-input and multiple-output, coordinated multi-point joint transmission, weighted minimum mean square error, block coordinate descent, alternation direction method of multipliers, asynchronous method

Citation Wu Z K, Fei Z S. Precoder design in downlink CoMP-JT MIMO network via WMMSE and asynchronous ADMM. Sci China Inf Sci, 2018, 61(8): 082306, https://doi.org/10.1007/s11432-017-9275-y

1 Introduction

Coordinated multi-point (CoMP) transmission is a promising technology, that can improve the throughput, especially for the cell edge throughput, of a network [1]. Two main CoMP strategies are defined by 3GPP, namely CoMP coordinated scheduling/beamforming (CoMP-CS/CB) and CoMP joint processing (CoMP-JP) [2]. As a case of CoMP-JP, CoMP joint transmission (CoMP-JT) defines the scenario in which a single UE may receive data from several base stations (BSs) simultaneously in the CoMP cooperation set. For CoMP-CB, the scenario in which a single user can only receive data from the serving BS it specifies, but the precoder is jointly decided by several BSs in the CoMP cooperation set [2,3]. Although CoMP-JT can typically achieve higher throughput than CoMP-CB, the process is more complicated, and larger backhaul capacity is needed. Various CoMP strategies are compared comprehensively in [4–6].

Numerous studies have concentrated on the precoder design of the CoMP-JT network in recent years. Some centralized algorithms such as block diagonalization (BD) [7] can be used directly in this scenario, as long as the number of total transmitter antennas is no less than the number of the receiver antennas [8]. Owing to high computation complexity, centralized algorithms are impractical when the network size is large. Therefore, many studies have attempted to design the precoder of the CoMP-JT network in a distributed manner. Considering the cluster size, the authors of [9] propose an algorithm based on

* Corresponding author (email: feizesong@bit.edu.cn)

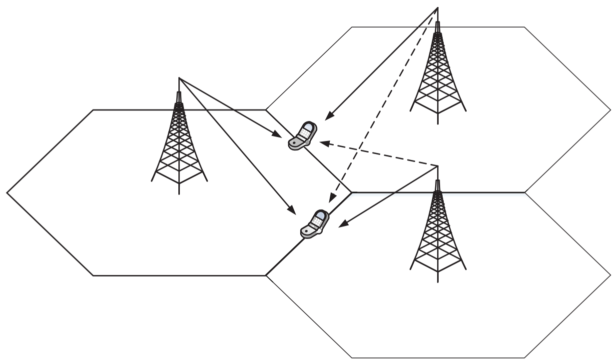

Figure 1 System model.

block coordinate descent (BCD) procedure. The multiplier of penalty terms can be used to control the cluster size; a similar method is also used in [10]. Owing to the non-convexity of throughput maximizing problem, heuristic algorithms can be used to solve the problem of CoMP-JT precoder design [11]. Another alternative way is to design algorithms based on minimum mean square error approach (MMSE), which is closely related to throughput maximization [9, 10, 12]. This method is firstly applied to the CoMP-CB network [13–16], and then extended to the CoMP-JT network. However, most of the studies on the CoMP-JT network consider the total power constraint of the transmitter [17–20], which is not as practical as the individual antenna power constraint.

In distributed computation systems, computation complexity varies in different calculation tasks, and calculation ability also varies in different computation nodes. The convergence rate of conventional distributed algorithms can be accelerated theoretically if we enable the asynchronous information exchange mechanism in those algorithms. Although few studies have been published in this field, this topic has attracted the attention of researchers [21–25].

In this paper, we propose an asynchronous distributed iterative algorithm for the optimal precoder design of the CoMP-JT network. Different from previous studies, our approach considers per-antenna power constraint instead of per-BS power constraint, and we design distributed iterative algorithm based on this constraint. Conventional distributed algorithms in previous studies are mainly synchronous iterative algorithms. By contrast, we consider and realize asynchronous information exchange in our algorithm, which is particularly important in large-scale network. We compare these algorithms comprehensively. The excellent performance of the proposed algorithm can be proven according to simulation results.

This paper is organized as follows. We model our system and formulate the problem in Section 2. The review of previous works is presented in Section 3. The details of the proposed algorithm are summarized in Section 4. In Section 5, the details of our numerical simulation are provided. Finally, this paper is concluded in Section 6.

Notations. Matrices and vectors are represented by boldface. $(\cdot)^{\mathrm{H}}$ denotes for Hermitian transpose. $\mathrm{Tr}(\cdot)$ indicates the trace and $\det(\cdot)$ the determinant. $\boldsymbol{A} \preceq \boldsymbol{B}$ indicates that $\boldsymbol{A}-\boldsymbol{B}$ is negative semidefinite. $\boldsymbol{I}_s$ denotes a identity matrix which has s dimensions. $\mathrm{diag}(\boldsymbol{A})$ means that the diagonal elements of $\boldsymbol{A}$ are taken to form a diagonal matrix.

2 System model and problem formulation

In this paper, we consider the downlink scenario of the CoMP-JT MIMO network, in which N_b BSs serve N_c users. The system model is demonstrated in Figure 1. The set of users is denoted by $\mathcal{U} = \{1, \ldots, N_c\}$, and the set of BSs is denoted by $\mathcal{T} = \{1, \ldots. N_b\}$. The set of users who were served by BS p in the CoMP network is represented by $\mathcal{U}_p$. $\mathcal{U}_p$ represents the set of BSs that serve the user i, and U_p denotes the number of users in $\mathcal{U}_p$. To improve the network throughput, especially for cell-edge users, several BSs may serve a user simultaneously. We use $\mathcal{T}_i$ to denote the set of BSs that serve the user i concurrently, and T_i to denote the number of BSs in $\mathcal{T}_i$. Each BS is equipped with N_t transmit antennas, and each user is equipped with N_r receive antennas. Each user can receive $d_s \leqslant \min(N_t, N_r)$ data streams from corresponding BSs. Let $\boldsymbol{s}_i$ denotes the transmit data streams to the i-th user in the CoMP network and $\mathrm{E}[\boldsymbol{s}_i\boldsymbol{s}_i^{\mathrm{H}}] = \boldsymbol{I}$. The precoding matrix $\boldsymbol{W}_{ip} \in \mathbb{C}^{N_t \times d_s}$ is defined for transmit data streams $\boldsymbol{s}_i$ of BS p. Based on above notations, the receive signals of user i in the CoMP network can be written as

$$\boldsymbol{y}_i = \sum_{p\in\mathcal{T}_i} \boldsymbol{H}_{ip}\boldsymbol{W}_{ip}\boldsymbol{s}_i + \sum_{j\in\mathcal{U}\backslash i}\sum_{p\in\mathcal{T}_j} \boldsymbol{H}_{ip}\boldsymbol{W}_{jp}\boldsymbol{s}_j + \boldsymbol{n}_i, \tag{1}$$

where $\boldsymbol{H}_{ip} \in \mathbb{C}^{N_r\times N_t}$ denotes the channel matrix between transmitter p and user i. $\boldsymbol{n}_i \in \mathbb{C}^{N_r\times 1}$ denotes the additive white Gaussian, of which the distribution is $\boldsymbol{n}_i \sim \mathcal{CN}(0, \sigma_i^2\boldsymbol{I}_{N_r})$. Considering individual antenna power constraint, the throughput maximizing problem of the CoMP-JT MIMO network can be formulated as

$$\begin{aligned} &\underset{\{\boldsymbol{W}_{ip}|i\in\mathcal{U},p\in\mathcal{T}_i\}}{\text{maximize}} \sum_{i=1}^{N_c} R_i \\ &\text{s.t.}\ \ \mathrm{diag}\left(\sum_{j\in\mathcal{U}_p} \boldsymbol{W}_{jp}\boldsymbol{W}_{jp}^{\mathrm{H}}\right) \preceq \frac{P_t}{N_t}\boldsymbol{I}_{N_t},\ \ \forall p\in\mathcal{T}, \end{aligned} \tag{2}$$

where

$$\begin{aligned} R_i =& \log\det\left[\boldsymbol{I}_{N_r} + \left(\sum_{p\in\mathcal{T}_i}\boldsymbol{H}_{ip}\boldsymbol{W}_{ip}\right)\left(\sum_{p\in\mathcal{T}_i}\boldsymbol{H}_{ip}\boldsymbol{W}_{ip}\right)^{\mathrm{H}}\right. \\ &\left.\times\left[\sum_{j\in\mathcal{U}\backslash i}\left(\sum_{p\in\mathcal{T}_j}\boldsymbol{H}_{ip}\boldsymbol{W}_{jp}\right)\left(\sum_{p\in\mathcal{T}_j}\boldsymbol{H}_{ip}\boldsymbol{W}_{jp}\right)^{\mathrm{H}} + \sigma_i^2\boldsymbol{I}_{N_r}\right]^{-1}\right]. \end{aligned} \tag{3}$$

No-convexity of (3) hinders solving it by standard convex optimization methods. However, Ref. [13] has proven the feasibility of transforming (3) to weighted MMSE (WMMSE) minimizing problem which can compute the local optimal solutions of (3). This method can also be used to solve the weighted sum-rate maximization problem [26]. Linear receive filter $\boldsymbol{A}_i$ is applied at user i to recover original data from received signals. The estimated MSE matrix of desired signals received by user i can be written as

$$\begin{aligned} \boldsymbol{E}_i &= \boldsymbol{E}\left[(\boldsymbol{A}_i\boldsymbol{y}_i - \boldsymbol{s}_i)(\boldsymbol{A}_i\boldsymbol{y}_i - \boldsymbol{s}_i)^{\mathrm{H}}\right] \\ &= \boldsymbol{A}_i\boldsymbol{C}_i\boldsymbol{A}_i^{\mathrm{H}} - \boldsymbol{Q}_i - \boldsymbol{Q}_i^{\mathrm{H}} + \sigma_i^2\boldsymbol{A}_i\boldsymbol{A}_i^{\mathrm{H}}, \end{aligned} \tag{4}$$

where

$$\boldsymbol{C}_i = \sum_{j\in\mathcal{U}}\left(\sum_{p\in\mathcal{T}_j}\boldsymbol{H}_{ip}\boldsymbol{W}_{jp}\right)\left(\sum_{p\in\mathcal{T}_j}\boldsymbol{H}_{ip}\boldsymbol{W}_{jp}\right)^{\mathrm{H}}, \tag{5}$$

and

$$\boldsymbol{Q}_i = \boldsymbol{A}_i\sum_{p\in\mathcal{T}_i}\boldsymbol{H}_{ip}\boldsymbol{W}_{ip}. \tag{6}$$

Let $\boldsymbol{G}_i \succeq \boldsymbol{0}$ indicates the weight matrix of user i. Let $\boldsymbol{W} = \{\boldsymbol{W}_{ip}|i\in\mathcal{U}, p\in\mathcal{T}_i\}$, $\boldsymbol{E} = \{\boldsymbol{E}_i|i\in\mathcal{U}\}$ and $\boldsymbol{G} = \{\boldsymbol{G}_i|i\in\mathcal{U}\}$. The corresponding WMMSE minimizing problem can be formulated as

$$\begin{aligned} &\underset{\boldsymbol{A},\boldsymbol{G},\boldsymbol{W}}{\text{minimize}} \sum_{i\in\mathcal{U}} \mathrm{Tr}(\boldsymbol{G}_i\boldsymbol{E}_i) - \log\det(\boldsymbol{G}_i) \\ &\text{s.t.}\ \ \mathrm{diag}\left(\sum_{i\in\mathcal{U}_p}\boldsymbol{W}_{ip}\boldsymbol{W}_{ip}^{\mathrm{H}}\right) \preceq \frac{P_t}{N_t}\boldsymbol{I}_{N_t},\ \ \forall p\in\mathcal{T}. \end{aligned} \tag{7}$$

Similar to previous studies, the BCD procedure is applied to solve $\boldsymbol{A}$, $\boldsymbol{G}$, $\boldsymbol{W}$ iteratively. Given $\boldsymbol{W}$, the optimal $\boldsymbol{A}^\star$ and $\boldsymbol{G}^\star$ can be written as

$$\boldsymbol{A}_i^\star = \frac{\sum_{p\in\mathcal{T}_i} \boldsymbol{W}_{ip}^{\mathrm{H}} \boldsymbol{H}_{ip}^{\mathrm{H}}}{\boldsymbol{C}_i + \sigma_i^2 \boldsymbol{I}_{N_r}}, \tag{8a}$$

$$\boldsymbol{G}_i^\star = \boldsymbol{E}_i^{-1} = \left(\boldsymbol{I}_{d_s} - \boldsymbol{A}_i^\star \sum_{p\in\mathcal{T}_i} \boldsymbol{H}_{ip}\boldsymbol{W}_{ip}\right)^{-1}. \tag{8b}$$

Given $\boldsymbol{A}$ and $\boldsymbol{G}$, the optimization problem of $\boldsymbol{W}$ can be expressed as

$$\begin{aligned} &\underset{\boldsymbol{W}}{\text{minimize}} \sum_{i\in\mathcal{U}} \mathrm{Tr}\left[\boldsymbol{G}_i\left(\boldsymbol{A}_i\boldsymbol{C}_i\boldsymbol{A}_i^{\mathrm{H}}\right)\right] - \sum_{i\in\mathcal{U}} \mathrm{Tr}\left[\boldsymbol{G}_i\left(\boldsymbol{Q}_i + \boldsymbol{Q}_i^{\mathrm{H}}\right)\right] \\ &\text{s.t.}\ \ \mathrm{diag}\left(\sum_{i\in\mathcal{U}_p} \boldsymbol{W}_{ip}\boldsymbol{W}_{ip}^{\mathrm{H}}\right) \preceq \frac{P_t}{N_t}\boldsymbol{I}_{N_t}, \quad \forall p \in \mathcal{T}. \end{aligned} \tag{9}$$

Obviously that (9) is convex with respect to $\boldsymbol{W}$. Therefore, standard convex optimization methods such as CVX can be applied to compute the optimal $\boldsymbol{W}$. However, being characterized with centralized and inefficient, CVX is impractical in handling large-scale network optimization. Therefore, distributed optimization algorithms have to be developed to solve this problem.

3 Precoder design based on BCD and ADMM

Although the coupling between precoder matrixes hinders obtaining the closed-form of optimal $\boldsymbol{W}$, convex optimization methods can still be applied to solve (9) iteratively. If we divide the precoder matrixes of each BS to several blocks of variables, namely, $\boldsymbol{W}_p = \{\boldsymbol{W}_{ip} | p \in \mathcal{T}, \forall i \in \mathcal{U}_p\}$, we can observe the fact that constraints can be decomposed among these blocks of variables. This division method is called "BS-centric divide". As the optimization goal cannot be divided according to the BS-centric divide, so BCD procedure (also termed as the nonlinear Gauss-Seidel algorithm) is applied by [9,10] to solve similar problems. We can also divide the precoder matrices according to different users, namely $\hat{\boldsymbol{W}}_i = \{\boldsymbol{W}_{ip} | i \in \mathcal{U}, \forall p \in \mathcal{T}_i\}$. We find that the optimization goal can be decomposed among these blocks of variables. This division method is called "user-centric divide", which is discussed in detail in Section 4. In this section, we extend previous studies to an individual situation with antenna power constraint.

For the block p of variables, namely $\boldsymbol{W}_p$, we have to solve the sub-problem as follows:

$$\begin{aligned} &\underset{\boldsymbol{W}_p}{\text{minimize}}\ \mathrm{Tr}\left[\sum_{j\in\mathcal{U}_p} \boldsymbol{T}_{pip} + \sum_{q\in\mathcal{T}_i\backslash p} \left(\boldsymbol{T}_{piq} + \boldsymbol{T}_{piq}^{\mathrm{H}}\right)\right] - \mathrm{Tr}\sum_{i\in\mathcal{U}_p}\left(\boldsymbol{Q}_{ip}\boldsymbol{W}_{ip} + \boldsymbol{W}_{ip}^{\mathrm{H}}\boldsymbol{Q}_{ip}^{\mathrm{H}}\right) \\ &\text{s.t.}\ \ \mathrm{diag}\left(\sum_{i\in\mathcal{U}_p} \boldsymbol{W}_{ip}\boldsymbol{W}_{ip}^{\mathrm{H}}\right) \preceq \frac{P_t}{N_t}\boldsymbol{I}_{N_t}, \end{aligned} \tag{10}$$

where

$$\begin{aligned} \boldsymbol{T}_{piq} &= \sum_{j\in\mathcal{U}} \boldsymbol{W}_{ip}^{\mathrm{H}}\boldsymbol{H}_{jp}^{\mathrm{H}}\boldsymbol{A}_j^{\mathrm{H}}\boldsymbol{G}_j\boldsymbol{A}_j\boldsymbol{H}_{jq}\boldsymbol{W}_{iq}, \\ \boldsymbol{Q}_{ip} &= \boldsymbol{G}_i\boldsymbol{A}_i\boldsymbol{H}_{ip}. \end{aligned}$$

Different blocks of variables are coupled in the optimization goal, and we take the variables in $\boldsymbol{W} \setminus \boldsymbol{W}_p$ as constants. Obviously (10) cannot be solved in closed-form, but the iterative method, such as the alternation direction method of multipliers (ADMM) algorithm [27], can be used to solve it.

To apply the ADMM algorithm, auxiliary variables $\boldsymbol{L} = \{\boldsymbol{L}_{ip} | \forall p \in \mathcal{T}, i \in \mathcal{U}_p\}$ are introduced. We similarly define $\boldsymbol{L}_p = \{\boldsymbol{L}_{ip} | p \in \mathcal{T}, \forall i \in \mathcal{U}_p\}$. Then (10) can be written as

$$\underset{\boldsymbol{W}_p, \boldsymbol{L}_p}{\text{minimize}}\ g_p(\boldsymbol{W}_p) + h_p(\boldsymbol{L}_p) \quad \text{s.t.} \ \ \boldsymbol{W}_{ip} = \boldsymbol{L}_{ip}, \quad \forall i \in \mathcal{U}_p, \tag{11}$$

where $g_p(\cdot)$ can be written as follows:

$$g_p(\boldsymbol{W}_p) = \text{Tr}\left[\sum_{i\in\mathcal{U}_p} \boldsymbol{T}_{pip} + \sum_{q\in\mathcal{T}_i \backslash p} \left(\boldsymbol{T}_{piq} + \boldsymbol{T}_{piq}^{\text{H}}\right)\right] - \text{Tr}\sum_{i\in\mathcal{U}_p}\left(\boldsymbol{Q}_{ip}\boldsymbol{W}_{ip} + \boldsymbol{W}_{ip}^{\text{H}}\boldsymbol{Q}_{ip}^{\text{H}}\right), \tag{12}$$

$h_p(\cdot)$ is a function, that indicates whether $\boldsymbol{L}_p$ satisfies the constraint

$$\text{diag}\left(\sum_{i\in\mathcal{U}_p} \boldsymbol{L}_{ip}\boldsymbol{L}_{ip}^{\text{H}}\right) \preceq \frac{P_t}{N_t}\boldsymbol{I}_{N_t}. \tag{13}$$

The Lagrange formulation of (12) is

$$\mathcal{L}_p(\boldsymbol{W}_p, \boldsymbol{L}_p, \boldsymbol{\Phi}_p) = g_p(\boldsymbol{W}_p) + h_p(\boldsymbol{L}_p) + \sum_{i\in\mathcal{U}_p} \rho\|\boldsymbol{L}_{ip} - \boldsymbol{W}_{ip} - \boldsymbol{\Phi}_{ip}\|_F^2, \tag{14}$$

where $\boldsymbol{\Phi}_{ip}$ is the normalized Lagrange multiplier corresponding to constraint $\boldsymbol{W}_{ip} = \boldsymbol{L}_{ip}$, $\boldsymbol{\Phi}_p = \{\boldsymbol{\Phi}_{ip} | p \in \mathcal{T}, \forall i \in \mathcal{U}_p\}$ and ρ is the penalty parameter. According to the update rules of ADMM, we sequentially update $\boldsymbol{L}_p$, $\boldsymbol{W}_p$ and $\boldsymbol{\Phi}_p$ until the algorithm converge or the maximum iteration number is reached. The update rules are detailed as follows.

(1) Procedure of $\boldsymbol{L}_p$ update. Given $\boldsymbol{W}_{ip}^{(t)}$ and $\boldsymbol{\Phi}_{ip}^{(t)}$, the update of $\boldsymbol{L}_p^{(t+1)}$ can be conducted according to

$$\begin{aligned} &\boldsymbol{L}_p^{(t+1)} = \underset{\boldsymbol{L}_p}{\text{minimize}} \sum_{i\in\mathcal{U}_p}\sum_{q\in\mathcal{T}_i} \rho\|\boldsymbol{L}_{ip} - \boldsymbol{W}_{ip}^{(t)} - \boldsymbol{\Phi}_{ip}^{(t)}\|_F^2, \\ &\text{s.t.} \ \ \text{diag}\left(\sum_{i\in\mathcal{U}_p} \boldsymbol{L}_{ip}\boldsymbol{L}_{ip}^{\text{H}}\right) \preceq \frac{P_t}{N_t}\boldsymbol{I}_{N_t}. \end{aligned} \tag{15}$$

Inspired by [13, 14], we further decompose $(\boldsymbol{W}_{ip}^{(t)} + \boldsymbol{\Phi}_{ip}^{(t)})^{\text{H}}$ and $\boldsymbol{L}_{ip}^{\text{H}}$ into

$$\begin{aligned} &\left(\boldsymbol{W}_{ip}^{(t)} + \boldsymbol{\Phi}_{ip}^{(t)}\right)^{\text{H}} = \left[\boldsymbol{w}_{ip}^{1\ (t)}, \ldots, \boldsymbol{w}_{ip}^{N_t\ (t)}\right], \\ &\boldsymbol{L}_{ip}^{\text{H}} = \left[\boldsymbol{l}_{ip}^{1}, \ldots, \boldsymbol{l}_{ip}^{N_t}\right]. \end{aligned} \tag{16}$$

So (15) can be written as

$$\begin{aligned} &\boldsymbol{L}_p^{(t+1)} = \underset{\boldsymbol{L}_p}{\text{minimize}} \sum_{i\in\mathcal{U}_p}\sum_{s=1}^{N_t} \rho\|\boldsymbol{l}_{ip}^{s} - \boldsymbol{w}_{ip}^{s\ (t)}\|_F^2, \\ &\text{s.t.} \ \ \sum_{i\in\mathcal{U}_p} \|\boldsymbol{l}_{ip}^{s}\|_2^2 \leqslant \frac{P_t}{N_t}. \end{aligned} \tag{17}$$

According to the Karush-Kuhn-Tucker (KKT) conditions, we can solve the optimal $\boldsymbol{l}_{ip}^{s}$ as follows:

$$\boldsymbol{l}_{ip}^{s\ (t+1)} = \begin{cases} \sqrt{\dfrac{P_t}{N_t \sum_{i\in\mathcal{U}_p} \|\boldsymbol{w}_{ip}^{s\ (t)}\|_2^2}}\,\boldsymbol{w}_{ip}^{s\ (t)}, & \displaystyle\sum_{i\in\mathcal{U}_p} \|\boldsymbol{w}_{ip}^{s\ (t)}\|_2^2 \geqslant \frac{P_t}{N_t}, \\ \boldsymbol{w}_{ip}^{s\ (t)}, & \text{otherwise}. \end{cases} \tag{18}$$

(2) Procedure of $\boldsymbol{W}_p$ update. For $\boldsymbol{W}_p$ with $\boldsymbol{L}_{ip}^{(t+1)}$ and $\boldsymbol{\Phi}_{ip}^{(t)}$ given, we update it as follows:

$$\boldsymbol{W}_p^{(t+1)} = \underset{\boldsymbol{W}_p}{\text{minimize}} \;\; g_p\left(\boldsymbol{W}_p\right) + \sum_{i \in \mathcal{U}_p} \rho \left\| \boldsymbol{L}_{ip}^{(t+1)} - \boldsymbol{W}_{ip} - \boldsymbol{\Phi}_{ip}^{(t)} \right\|_F^2 . \tag{19}$$

Let the derivative of (19) about $\boldsymbol{W}_{ip}$ to be zero, and then we can get the optimal $\boldsymbol{W}_{ip}$. Let $\boldsymbol{S}_{pq} = \sum_{i \in \mathcal{U}} \boldsymbol{H}_{ip}^{\mathrm{H}} \boldsymbol{A}_i^{\mathrm{H}} \boldsymbol{G}_i \boldsymbol{A}_i \boldsymbol{H}_{iq}$, and the optimal $\boldsymbol{W}_{ip}$ can be written as

$$\boldsymbol{W}_{ip}^{(t+1)} = \left(\boldsymbol{S}_{pp} + \rho\right)^{-1} \left[\boldsymbol{Q}_{ip}^{\mathrm{H}} + \rho\left(\boldsymbol{L}_{ip}^{(t+1)} - \boldsymbol{\Phi}_{ip}^{(t)}\right) - \sum_{q \in \mathcal{T}_i \backslash p} \boldsymbol{S}_{pq} \boldsymbol{W}_{iq}^{(t)} \right]. \tag{20}$$

(3) Procedure of $\boldsymbol{\Phi}_p$ update. The update of the Lagrange multiplier $\boldsymbol{\Phi}_{jp}$ is given by

$$\boldsymbol{\Phi}_{ip}^{(t+1)} = \boldsymbol{\Phi}_{ip}^{(t)} + \mu\left(\boldsymbol{W}_{ip}^{(t+1)} - \boldsymbol{L}_{ip}^{(t+1)}\right), \tag{21}$$

where μ is the step size. We name the proposed iterative algorithm as WMMSE-BCD-ADMM (WBA) algorithm. This algorithm can be operated in a parallel or distributed manner. In this paper, we only discuss how to conduct the WBA algorithm as a distributed algorithm here. Similar to [13], our study makes the following assumptions.

Assumption 1. Each BS knows it is channel state information (CSI) between itself and the users it serves.

Assumption 2. Each BS is linked to neighbor BSs, so only neighbouring BSs can exchange information, including CSI and calculation results.

We denote neighbouring set $\mathcal{N}_p = \{q | \forall i \in \mathcal{U}_p, \forall q \in \mathcal{T}_i\}$ for BS p. By our definition, the BS p itself also belongs to it is neighbor set. By our assumption, the CSI and calculation results exchange only happens between BS p and the BSs in it is neighbor set.

Assumption 3. Each terminal can accomplish several basic measurement and calculation tasks, and feed calculation results back to corresponding BS via an upload link.

Based on the preceding mentioned above, we provide details of the algorithm in Algorithms 1–3, where n_i indicates the iteration number of WMMSE iteration and $n_{\max}$ denotes the maximum iteration number.

Algorithm 1 The WMMSE algorithm for solving problem (7)

Terminal i;
Initialize $\boldsymbol{A}_i^{(0)}$, $\boldsymbol{G}_i^{(0)}$, $\boldsymbol{E}_i^{(0)}$, $n_i = 0$;
repeat
    Update $\boldsymbol{A}_i^{(n_i+1)}$ according to (8a);
    Update $\boldsymbol{G}_i^{(n_i+1)}$ according to (8b);
    Feed $\boldsymbol{A}_i^{(n_i+1)}$ and $\boldsymbol{G}_i^{(n_i+1)}$ to BSs in $\mathcal{T}_i$;
    Receive $\boldsymbol{W}_p^{(n_i+1)}$ from BSs in $\mathcal{T}_i$ via pilot frequency;
    Calculate $\boldsymbol{E}_i^{(n_i+1)}$ according to (4);
    $n_i = n_i + 1$;
until $n_i < n_{\max}$.

---

4 Precoder design based on asynchronous ADMM

The WBA algorithm proposed in Section 3 can solve (9) theoretically. However, a possible drawback of this algorithm is that the BCD procedure (Algorithm 2) cannot be conducted in parallel. In the ultra-dense CoMP-JT network, the practical applicability of the WBA algorithm is a problem because the time overhead increases linearly according to the scale of the network. A possible solution is the nonlinear Jacobi algorithm, which can be conducted in a parallel way. However, the (9) is not strongly

Algorithm 2 The BCD algorithm for solving problem (9)

```
Initialize W_p^(0), ∀p ∈ T, r_p = 0;
repeat
    Circularly pick up q ∈ T;
    BS p;
    if p = q;
        Update W_p^(r_p) according to Algorithm 3;
    else
        W_q^(r_p+1) = W_q^(r_p);
    r_p = r_p + 1;
until r_p < r_max, ∀p ∈ T.
```

Algorithm 3 The ADMM algorithm for solving problem (10)

```
Initialize L_p^(0), Φ_p^(0), t_p = 0;
repeat
    Update L_p^(t_p+1) according to (18);
    Update W_p^(t_p+1) according to (20);
    Update Φ_p^(t_p+1) according to (21);
until t_p < t_max.
```

convex with respected to $\boldsymbol{W}$, so a unique optimal solution can not be guaranteed [28]. Therefore, the convergence of the nonlinear Jacobi algorithm becomes troublesome. Inspired by the parallel structure of the ADMM algorithm, we employ the ADMM algorihtm to solve (9) directly.

Firstly, we write (9) as another form, though which we can decompose the original optimization in a user-centric way, and which can be formulate as

$$\begin{aligned} &\underset{\boldsymbol{W}}{\text{minimize}} \sum_{i\in\mathcal{U}} \operatorname{Tr}(\boldsymbol{D}_i) - \sum_{i\in\mathcal{U}} \operatorname{Tr}\left[\boldsymbol{G}_i\left(\boldsymbol{Q}_i + \boldsymbol{Q}_i^{\mathrm{H}}\right)\right], \\ &\text{s.t.}\ \ \operatorname{diag}\left(\sum_{i\in\mathcal{U}_p} \boldsymbol{W}_{ip}\boldsymbol{W}_{ip}^{\mathrm{H}}\right) \preceq \frac{P_t}{N_t}\boldsymbol{I}_{N_t},\quad \forall p\in\mathcal{T}, \end{aligned} \tag{22}$$

where

$$\boldsymbol{D}_i = \sum_{j\in\mathcal{U}} \boldsymbol{G}_j\boldsymbol{A}_j\left(\sum_{p\in\mathcal{T}_i}\boldsymbol{H}_{jp}\boldsymbol{W}_{ip}\right)\left(\sum_{p\in\mathcal{T}_i}\boldsymbol{H}_{jp}\boldsymbol{W}_{ip}\right)^{\mathrm{H}}\boldsymbol{A}_j^{\mathrm{H}}. \tag{23}$$

We can further decompose (23) to sub-problems according to different users. Through this decomposition method, we can solve different optimal $\hat{\boldsymbol{W}}_i = \{\boldsymbol{W}_{ip} | i\in\mathcal{U}, \forall p\in\mathcal{T}_i\}$ a parallel manner and in a closed form. Similar to the procedure we performed in Section 3, we define

$$f_i(\hat{\boldsymbol{W}}_i) = \operatorname{Tr}(\boldsymbol{D}_i) - \operatorname{Tr}\sum_{p\in\mathcal{T}_i}\left(\boldsymbol{Q}_{ip}\boldsymbol{W}_{ip} + \boldsymbol{W}_{ip}^{\mathrm{H}}\boldsymbol{Q}_{ip}^{\mathrm{H}}\right). \tag{24}$$

Then, Eq. (22) can be written as

$$\begin{aligned} &\underset{\hat{\boldsymbol{W}}_i,\boldsymbol{L}_p}{\text{minimize}} \sum_{j\in\mathcal{U}} f_i(\hat{\boldsymbol{W}}_i) + \sum_{p\in\mathcal{T}} h_p(\boldsymbol{L}_p), \\ &\text{s.t.}\ \ \boldsymbol{W}_{ip} = \boldsymbol{L}_{ip},\quad \forall p\in\mathcal{T},\quad \forall i\in\mathcal{U}_p. \end{aligned} \tag{25}$$

The procedure of the ADMM algorithm is detailed in Section 3. Evidently the WMMSE-ADMM (WA) algorithm, the procedure of $\boldsymbol{L}_p$ update and $\boldsymbol{\Phi}_p$ update are the same as that of the WBA algorithm. Therefore, we omit the details and only formulate the update method of $\hat{\boldsymbol{W}}_i$.

Procedure of the $\hat{\boldsymbol{W}}_i$ update. Similar to the procedures in Section 3, the optimal $\boldsymbol{W}_{ip}^{(t+1)}$ can be written as

$$\boldsymbol{W}_{ip}^{(t+1)} = (\boldsymbol{S}_{pp} + \rho)^{-1} \left[\boldsymbol{Q}_{ip}^{\mathrm{H}} + \rho \left(\boldsymbol{L}_{ip}^{(t+1)} - \boldsymbol{\Phi}_{ip}^{(t)} \right) - \sum_{q \in \mathcal{T}_i \backslash p} \boldsymbol{S}_{pq} \boldsymbol{W}_{iq}^{(t+1)} \right]. \tag{26}$$

The only difference between (20) and (26) is that in (20), $\boldsymbol{W}_{iq}$ is known, whereas in (26), it is a variable to be calculated. In (26), the optimal $\boldsymbol{W}_{ip}^{(t+1)}$ is coupled with other elements in $\hat{\boldsymbol{W}}_i^{(t+1)}$. Furthermore, we can list T_i similar equations in total. We further combine those equations so that we can solve the optimal $\boldsymbol{W}_{ip}^{(t+1)}$ in a closed form. We denote $\mathcal{T}_i = \{\mathcal{T}_i^1, \mathcal{T}_i^2, \ldots, \mathcal{T}_i^{T_i}\}$, $\tilde{\boldsymbol{W}}_i = [\boldsymbol{W}_{i\mathcal{T}_i^1}^{\mathrm{H}}, \boldsymbol{W}_{i\mathcal{T}_i^2}^{\mathrm{H}}, \ldots, \boldsymbol{W}_{i\mathcal{T}_i^{T_i}}^{\mathrm{H}}]^{\mathrm{H}}$. The update procedure of $\tilde{\boldsymbol{W}}_i$ can be written as

$$\tilde{\boldsymbol{W}}_i^{(t+1)} = (\boldsymbol{K}_i + \rho \boldsymbol{I}_{N_r \times T_i})^{-1} \boldsymbol{M}_i^{(t)}, \tag{27}$$

where

$$\boldsymbol{K}_i = \begin{bmatrix} \boldsymbol{S}_{\mathcal{T}_i^1 \mathcal{T}_i^1} & \boldsymbol{S}_{\mathcal{T}_i^1 \mathcal{T}_i^2} & \ldots & \boldsymbol{S}_{\mathcal{T}_i^1 \mathcal{T}_i^{T_i}} \\ \boldsymbol{S}_{\mathcal{T}_i^2 \mathcal{T}_i^1} & \boldsymbol{S}_{\mathcal{T}_i^2 \mathcal{T}_i^2} & \ldots & \boldsymbol{S}_{\mathcal{T}_i^2 \mathcal{T}_i^{T_i}} \\ \vdots & \vdots & & \vdots \\ \boldsymbol{S}_{\mathcal{T}_i^{T_i} \mathcal{T}_i^1} & \boldsymbol{S}_{\mathcal{T}_i^{T_i} \mathcal{T}_i^2} & \ldots & \boldsymbol{S}_{\mathcal{T}_i^{T_i} \mathcal{T}_i^{T_i}} \end{bmatrix}, \tag{28}$$

and

$$\boldsymbol{M}_i^{(t)} = \begin{bmatrix} \boldsymbol{Q}_{i\mathcal{T}_i^1} + \rho(\boldsymbol{L}_{i\mathcal{T}_i^1}^{(t+1)} - \boldsymbol{\Phi}_{i\mathcal{T}_i^1}^{(t)}) \\ \boldsymbol{Q}_{i\mathcal{T}_i^2} + \rho(\boldsymbol{L}_{j\mathcal{T}_i^2}^{(t+1)} - \boldsymbol{\Phi}_{i\mathcal{T}_i^2}^{(t)}) \\ \vdots \\ \boldsymbol{Q}_{i\mathcal{T}_i^{T_i}} + \rho(\boldsymbol{L}_{i\mathcal{T}_i^{T_i}}^{(t+1)} - \boldsymbol{\Phi}_{i\mathcal{T}_i^{T_i}}^{(t)}) \end{bmatrix}. \tag{29}$$

4.1 Parallel implementation of the WA algorithm

We assume that a fusion center exists in the network. All CSI information is collected and computation tasks are operated in this center. Parallel computation is enabled in this center by multithreading technology. The WMMSE algorithm can certainly be conducted in parallel for different users. Then we should calculate $(\boldsymbol{K}_i + \rho \boldsymbol{I}_{N_r \times T_i})^{-1}$ and $\boldsymbol{M}_i^{(t)}$ in parallel for different users. Evidently Eqs. (18), (26) and (21) can also be conducted in parallel. In other words, the WA algorithm can be operated efficiently if we treat it as a centralized algorithm.

4.2 Distributed implementation of WA algorithm

Different from parallel implementation, in distributed implementation, each calculation node can access a fraction of CSI information and calculation results of other BSs in order to reduce signaling overheads. As $\boldsymbol{K}_i$ needs all the CSI information of the BSs in $\mathcal{T}_i$, obviously, we cannot update (27) in terminals. For terminal i, we choose a specific BS, noted by BS p_j to calculate $(\boldsymbol{K}_i + \rho \boldsymbol{I}_{N_r \times T_i})^{-1}$. Let $(\boldsymbol{K}_i + \rho \boldsymbol{I}_{N_r \times T_i})^{-1} = [\boldsymbol{F}_{i\mathcal{T}_i^1}^{\mathrm{H}}, \boldsymbol{F}_{i\mathcal{T}_i^2}^{\mathrm{H}}, \ldots, \boldsymbol{F}_{i\mathcal{T}_i^{T_i}}^{\mathrm{H}}]^{\mathrm{H}}$, in which $\boldsymbol{F}_{ip} \in \mathbb{C}^{N_t \times T_i N_t}$. Then the update of $\boldsymbol{W}_{jp}$ can be formulated as

$$\boldsymbol{W}_{ip}^{(t+1)} = \boldsymbol{F}_{ip} \times \boldsymbol{M}_i^{(t)}. \tag{30}$$

As we update $\boldsymbol{L}_p$ update and $\boldsymbol{\Phi}_p$ in a BS-centric way, we can also update $\boldsymbol{W}_{jp}^{(t+1)}$ in a BS-centric way. We detail the proposed WA algorithm for the CoMP-JT network in Algorithms 4–6. A slight change of the WMMSE algorithm in the WA algorithm is that $\boldsymbol{A}_i^{(n_i+1)}$ and $\boldsymbol{G}_i^{(n_i+1)}$ are feed to BS i_p, rather than BSs in $\mathcal{T}_i$.

4.3 Asynchronous implementation of WA algorithm

As mentioned in Subsections 4.1 and 4.2, the WA algorithm can be conducted in a parallel or distributed manner. If we conducted the WA algorithm in the distributed way, we have to consider that the computation ability, computation load, computation complexity and communication delay of different computation nodes vary from one another. Therefore, for each iteration of the WA algorithm, time overheads of different computation nodes may vary in a wide range. The covergence rate of the synchronous WA algorithm will be limited by the "slowest" computation node. We can accelerate the coverage rate by enabling asynchronous information exchange mechanism in the WA algorithm, and we call it WaA algorithm. In other words, in Algorithm 6, we can update $\boldsymbol{W}_p^{(t_p+1)}$ for BS p before we receive all $(\boldsymbol{L}_{iq}^{(t_q+1)}, \boldsymbol{\Phi}_{iq}^{(t_q+1)})$ from BSs in $\mathcal{N}_p$. $\chi_p^{t_p}$ denotes the set of BSs whose computation results arrive timely for (t_p+1)-th iteration of $\boldsymbol{W}_p$ and $\eta_p^{\min}$ is the minimum num of such BSs. To guarantee the convergence of the asynchronous ADMM (aADMM) algorithm, we set up Assumption 4 according to [29].

Algorithm 4 The WMMSE algorithm to solve problem (7)

Terminal i;
  Initialize $\boldsymbol{A}_i^{(0)}$, $\boldsymbol{G}_i^{(0)}$, $\boldsymbol{E}_i^{(0)}$, $n_i=0$;
repeat
  Update $\boldsymbol{A}_i^{(n_i+1)}$ according to (8a);
  Update $\boldsymbol{G}_i^{(n_i+1)}$ according to (8b);
  Feed $\boldsymbol{A}_i^{(n_i+1)}$ and $\boldsymbol{G}_i^{(n_i+1)}$ to BS i_p;
  Receive $\boldsymbol{W}_p^{(n_i+1)}$ from BSs in $\mathcal{T}_i$ via pilot frequency;
  Calculate $\boldsymbol{E}_i^{(n_i+1)}$ according to (4);
  $n_i = n_i + 1$;
until $n_i < n_{\max}$.

Algorithm 5 Auxiliary calculation to solve (9)

BS i_p;
  Wait until all users in $\mathcal{U}_{i_p}$ feed their $\boldsymbol{A}_i^{(n_i+1)}$ and $\boldsymbol{G}_i^{(n_i+1)}$ to it;
  Calculate $\{\boldsymbol{F}_{ip} | \forall p \in \mathcal{T}_i\}$ for each user i and feed the $\boldsymbol{F}_{ip}$ to BS p.

Algorithm 6 ADMM algorithm to solve problem (10)

BS p;
  Initialize $\boldsymbol{W}_p^{(0)}$, $\boldsymbol{L}_p^{(0)}$, $\boldsymbol{\Phi}_p^{(0)}$ $p \in \mathcal{T}$, $t_p = 0$;
repeat
  Update $\boldsymbol{L}_p^{(t_p+1)}$ according to (18);
  Update $\boldsymbol{\Phi}_p^{(t_p+1)}$ according to (21);
  Feed $(\boldsymbol{L}_{ip}^{(t_p+1)}, \boldsymbol{\Phi}_{ip}^{(t_p+1)})$ back to corresponding BS in $\mathcal{N}_p$;
  Wait until all $(\boldsymbol{L}_{iq}^{(t_q+1)}, \boldsymbol{\Phi}_{iq}^{(t_q+1)})$ is received from BSs in $\mathcal{N}_p$;
  Let $(\boldsymbol{L}_{iq}^{(t_p+1)}, \boldsymbol{\Phi}_{iq}^{(t_p+1)}) = (\boldsymbol{L}_{iq}^{(t_q+1)}, \boldsymbol{\Phi}_{iq}^{(t_q+1)})$;
  Update $\boldsymbol{W}_p^{(t_p+1)}$ according to (30);
  $t_p = t_p + 1$;
until $t_p < t_{\max}$.

Assumption 4. The renewal of $(\boldsymbol{L}_{iq}^{(t_p+1)}, \boldsymbol{\Phi}_{iq}^{(t_p+1)})$ of BS p can be skipped at most $\tau_{\max}$ times.

Assumption 4 indicates that $\mathcal{N}_p \subseteq \chi_p^{t_p} \cup \chi_p^{t_p+1} \cdots \cup \chi_p^{t_p+\tau_{\max}-1}$. Let $\psi_p^{t_p} = \{q | \tau_{pq}^{t_p} \geqslant \tau_{\max} - 1\}$ indicates the set of those BSs whose $(\boldsymbol{L}_{iq}^{(t_q+1)}, \boldsymbol{\Phi}_{iq}^{(t_q+1)})$ must be provided for the (t_p+1)-th iteration of BS p, in case of the violation of Assumption 4. The details of the aADMM algorithm are summarized in Alogrithm 7.

4.4 Computational complexity

For the convenience of computational complexity analysis, we assume that T_i is the same for all users, and U_p is the same for all BSs. Therefore, $T_i N_c = U_p N_b$. We also assume $N_b \times N_t \geqslant N_c \times N_r$, which

Algorithm 7 aADMM algorithm to solve problem (10)

```
BS p;
    Initialize W_p^(0), L_p^(0), Φ_p^(0) p ∈ T, t_p = 0;
repeat
    Update L_p^(t_p+1) according to (18);
    Update Φ_p^(t_p+1) according to (21);
    Feed (L_jp^(t_p+1), Φ_jp^(t_p+1)) back to corresponding BSs in N_p;
    Wait until at least η_p^min BSs in N_p provide their (L_jq^(t_q+1), Φ_jq^(t_q+1)) to BS p;
    If q ∈ χ_p^(t_p+1)
        set (L_jq^(t_p+1), Φ_jq^(t_p+1)) = (L_jq^(t_q+1), Φ_jq^(t_q+1));
        set τ_pq^(t_p+1) = 0;
    else
        set (L_jq^(t_p+1), Φ_jq^(t_p+1)) = (L_jq^(t_p), Φ_jq^(t_p));
        set τ_pq^(t_p+1) = τ_pq^(t_p+1) + 1;
    Update W_p^(t_p+1) according to (26);
    t_p = t_p + 1;
until Stop.
```

means that total transmit antennas are more than the total receive antennas.

For the computational complexity of the WBA algorithm, we firstly analyze the calculation complexity of solving (10). The calculation complexity of solving (10) mainly depends on (20). For BS p and user i, the computational complexity of (20) is $O(N_t^2 N_r T_i)$. Therefore, the computational complexity of Algorithms 2 and 3, and WBA algorithm are $O(t_{\max} N_t^2 N_r T_i^2 U_p)$, $O(r_{\max} t_{\max} N_t^2 N_r T_i^3 N_c)$, and $O(n_{\max} r_{\max} t_{\max} N_t^2 N_r T_i^3 N_c)$, respectively.

For the WA algorithm, we firstly analyze the computational complexity of Algorithm 6. Therefore, for Algorithm 6, the computational complexity mainly depends on (30), of which the computational complexity is $O(N_t^2 N_r)$. Therefore, for Algorithm 6, the computational complexity is $O(t_{\max} N_t^2 N_r T_i^2 N_c)$. For Algorithm 5, the calculation complexity mainly lies in the calculation of $(\boldsymbol{K}_i + \rho \boldsymbol{I}_{N_r \times T_i})^{-1}$, of which the calculation complexity is $O(N_c (T_i N_t)^3)$. Therefore, the computational complexity of the WA algorithm is $O(n_{\max}(N_c (T_i N_t)^3 + t_{\max} N_t^2 N_r T_i^2 N_c))$. For the WaA algorithm, the computational complexity is the same as that of the WA algorithm.

We take full coordination zero-forcing algorithm for comparison, of which the computational complexity is $O(N_c (N_t N_b)^3)$. Evidently, the calculation complexity of WBA, WA and WaA increase linearly with increasing network size. However, for full coordination zero-forcing algorithm, there will be a cubic increase with increasing network size. A similar situation happens when we take CVX to solve can solve (9) via interior-point method, of which the computational complexity also increases cubic with increasing network size. Therefore, the WBA, WA and WaA algorithm are suitable for solving (1) in the large-scale CoMP-JT network.

5 Numerical simulation and results

We consider the downlink scenario of a CoMP-JT MIMO network in which N_b BSs and N_c users are randomly placed in a square area of $20N_b \times 20N_b$ square metres. The minimum distance between BSs is 30 m. Users are randomly located in the cell edge area of each BS. Path loss is modeled by ITU urban macro model. We assume that the channel gain obeys the complex Gaussian distribution $\mathcal{CN}(0,1)$. Each BS is equipped with $N_t = 4$ transmit antennas. Each user is equipped with $N_r = 2$ receive antennas, and receives $d_s = 2$ data streams. We define $\sigma_n^2 = 10$ mW and SNR $= P_t/\sigma_n^2$. Various criteria can be used to ensure the BSs cooperation set of a specific user. For instance, the maximum BSs number of the BSs cooperation set is limited, or the minimum SINR of the signals received from the BSs in the cooperation set should be larger than a specific threshold. Although we don not concentrate on how to ensure the BSs cooperation set of each user, our algorithm can be applied to an arbitrary combination of users and BSs. We consider two situations in our simulation: (1) CoMP-CB, in which we assume

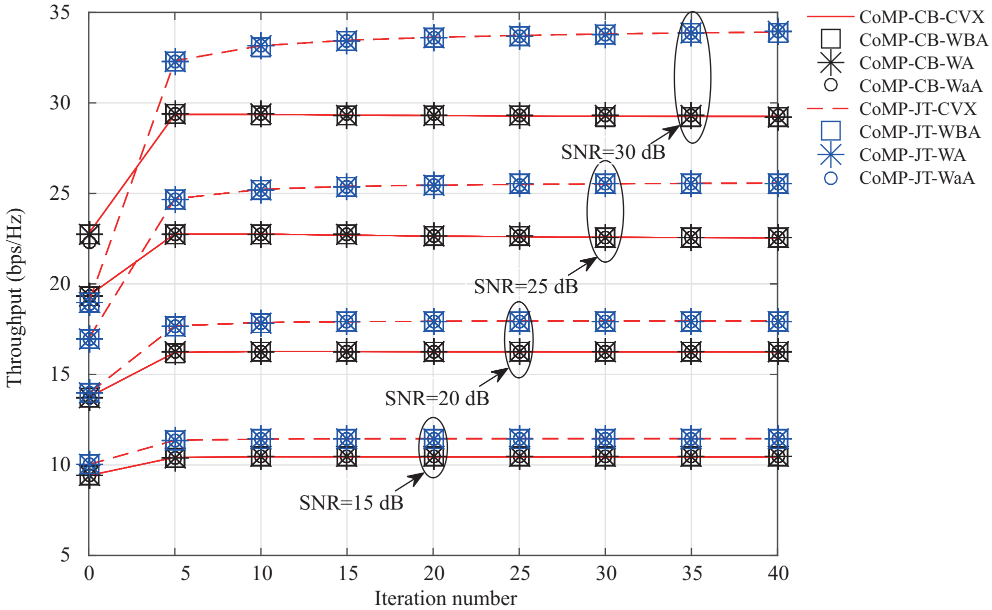

Figure 2 (Color online) Coverage examples of CVX, WBA, WA, and WaA algorithm.

that each user is assigned to the BS with highest instantaneous SINR; (2) CoMP-JT, in which each user is served by available BSs in the CoMP-JT MIMO network. CVX is also used to verify the optimality of our algorithm. We denote the simulation results of CVX as “CoMP-CB-CVX” and “CoMP-JT-CVX”, respectively. The simulation results of our algorithm are represented by “CoMP-CB-WA” and “CoMP-JT-WA”, respectively. Similarly, we define “CoMP-CB-WaA”, “CoMP-JT-WaA”, “CoMP-CB-WBA” and “CoMP-JT-WBA”. We also consider full coordination zero-forcing beamforming method in our simulation, which is denoted by “CoMP-JT-ZF”.

We assume that the time overhead for each calculation task distributes uniformly between four and eight time units. BS p can update $\boldsymbol{W}_p^{(t_p+1)}$ with at least $\eta_p^{\min} = 2$ BSs in $\mathcal{N}_p$ provide their $(\boldsymbol{L}_{iq}^{(t_q+1)}, \boldsymbol{\Phi}_{iq}^{(t_q+1)})$ to BS p and we set $\tau_{\max} = 4$.

In Figure 2, we demonstrate a convergence example of CVX and the WBA, WA, and WaA algorithms in terms of different SNR values. In this situation, the throughput of CoMP-JT outperforms that of CoMP-CB by nearly at least 10.1%–15.3% in terms of different SNRs. The optimum of the WBA algorithm, WA algorithm, WaA algorithm can also be proven according to Figure 2. All the algorithms can converge at no more than 40 WMMSE iterations.

Figure 3 shows the throughput of the CoMP-JT network and CoMP-CB networks in terms of different SNRs. Whether in the high SNR region or in low SNR region, the throughput of the CoMP-JT network can outperform CoMP-CB significantly. Generally speaking, CoMP-JT can improve the throughput of the network by 10%–17% in terms of different SNRs. The results of the WBA algorithm, WA algorithm and WaA algorithm are very close to the calculation results of CVX, which prove the optimality of those algorithms. Furthermore, we compare the full coordination zero-forcing method with our algorithms. Simulations results show that our algorithms significantly outperforms the full coordination zero-forcing beamforming method significantly.

As the algorithms proposed in this article are mainly iterative, the influence of different initial points should be analyzed. Figure 4 shows the influence of different initial points of the WaA algorithm in terms of different SNRs. “Boundary point”, “Exterior point”, and “Interior point” mean that the initial point is boundary point, exterior point and interior point of feasible region respectively. Evidently, the convergence results of the interior point outperform that of the exterior and boundary points, especially when SNR is relatively high. Therefore, the interior point should be chosen for the initial point of those algorithms.

Figure 5 shows the convergence rate difference of asynchronous ADMM and synchronous ADMM in

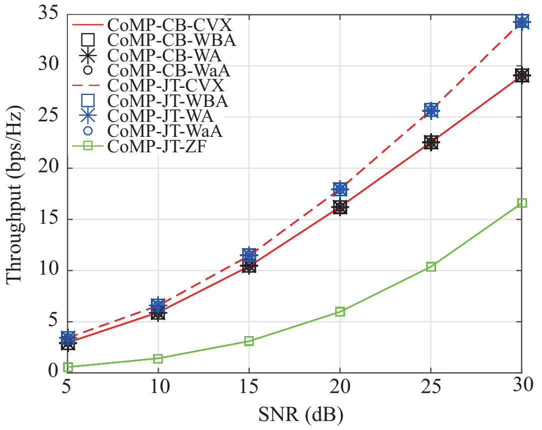

Figure 3 (Color online) Throughput of CoMP-JT vs. CoMP-CB network in terms of different SNRs.

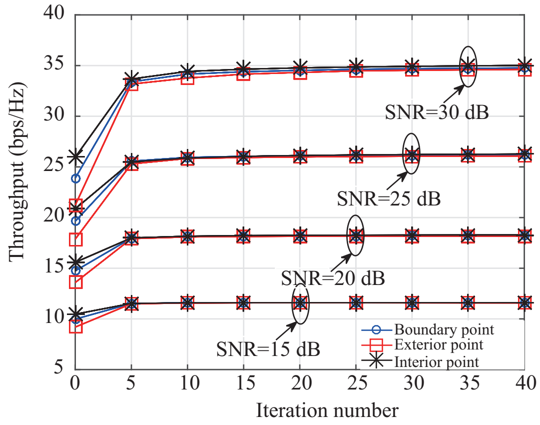

Figure 4 (Color online) Influence of different initial points of WaA algorithm in terms of different SNRs.

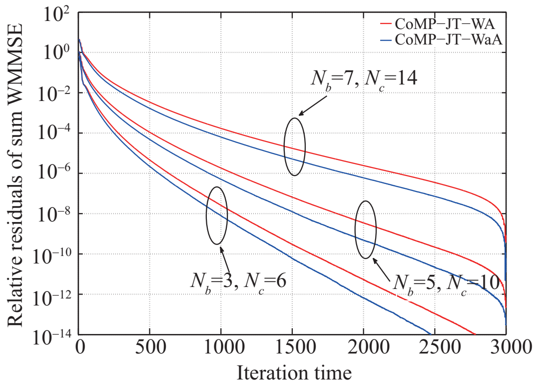

Figure 5 (Color online) Coverage speed of asynchronous ADMM and synchronous ADMM in terms of different network sizes, SNR = 15 dB.

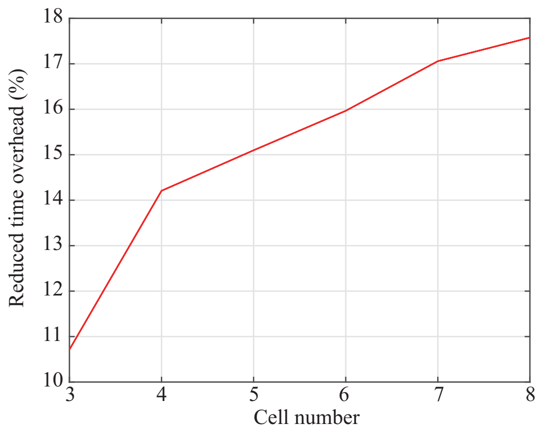

Figure 6 (Color online) Reduced time overhead of asynchronous ADMM in terms of different network sizes, SNR = 15 dB.

terms of various network sizes. The converge speed of both asynchronous ADMM and synchronous ADMM decrease with increasing network size. Furthermore, asynchronous ADMM converges faster than synchronous ADMM in terms of network size. The reduced degree of time overhead brought by asynchronous information exchange in terms of different network sizes is detailed in Figure 6. We set $\eta_p^{\min} = \lfloor 2/3 \times T_p \rfloor$. We can observe an increasing trend when the network size increases. When $N_b = 8$ and $N_c = 16$, asynchronous ADMM can coverage nearly 17.5% faster than synchronous ADMM.

6 Conclusion

CoMP-JT is a promising technology for future wireless communication system. To achieve maximum throughput, we propose an iterative method to handle the precoder design problem in the CoMP-JT network. Based on WMMSE, BCD and ADMM, previous study is extended to the scenario of individual antenna power constraint. Then, we propose an asynchronous distributed iterative algorithm based on WMMSE, ADMM and asynchronous calculation. The proposed algorithm we proposed can also be conducted in a parallel manner, which is practical for a large-scale wireless network. Numerical results show a gain in throughput, decrease in time overhead, and the optimality of the precoding scheme provided by our algorithm.

Acknowledgements This work was supported in part by Beijing Natural Science Foundation (Grant No. 4152047) and National Natural Science Foundation of China (Grant Nos. 61671058, 61371075).

References

1 Irmer R, Droste H, Marsch P, et al. Coordinated multipoint: concepts, performance, and field trial results. IEEE Commun Mag, 2011, 49: 102–111
2 Access EUTR. Further advancements for E-UTRA physical layer aspects. 3GPP Technical Report TR36.814. 2010
3 Gong S Q, Xing C W, Fei Z S, et al. Cooperative beamforming design for physical-layer security of multi-hop MIMO communications. Sci China Inf Sci, 2016, 59: 062304
4 Lee D, Seo H, Clerckx B, et al. Coordinated multipoint transmission and reception in LTE-advanced: deployment scenarios and operational challenges. IEEE Commun Mag, 2012, 50: 148–155
5 Sawahashi M, Kishiyama Y, Morimoto A, et al. Coordinated multipoint transmission/reception techniques for LTE-advanced [coordinated and distributed MIMO]. IEEE Wirel Commun, 2010, 17: 26–34
6 Cui Q M, Wang H, Hu P X, et al. Evolution of limited-feedback CoMP systems from 4G to 5G: CoMP features and limited-feedback approaches. IEEE Veh Technol Mag, 2014, 9: 94–103
7 Shim S, Kwak J S, Heath R W, et al. Block diagonalization for multi-user MIMO with other-cell interference. IEEE Trans Wirel Commun, 2008, 7: 2671–2681
8 Zhang J, Chen R, Andrews J G, et al. Networked MIMO with clustered linear precoding. IEEE Trans Wirel Commun, 2009, 8: 1910–1921
9 Hong M Y, Sun R, Baligh H, et al. Joint base station clustering and beamformer design for partial coordinated transmission in heterogeneous networks. IEEE J Sel Areas Commun, 2013, 31: 226–240
10 Lagen S, Agustin A, Vidal J, et al. Distributed user-centric clustering and precoding design for CoMP joint transmission. In: Proceedings of IEEE Global Communications Conference, San Diego, 2015
11 Bjornson E, Zakhour R, Gesbert D, et al. Cooperative multicell precoding: rate region characterization and distributed strategies with instantaneous and statistical CSI. IEEE Trans Signal Process, 2010, 58: 4298–4310
12 Liao W C, Hong M Y, Liu Y F, et al. Base station activation and linear transceiver design for optimal resource management in heterogeneous networks. IEEE Trans Signal Process, 2014, 62: 3939–3952
13 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. IEEE Trans Signal Process, 2011, 59: 4331–4340
14 Kim T M, Sun F, Paulraj A J. Low-complexity MMSE precoding for coordinated multipoint with per-antenna power constraint. IEEE Signal Process Lett, 2013, 20: 395–398
15 Gong S Q, Xing C W, Fei Z S, et al. Millimeter-wave secrecy beamforming designs for two-way amplify-and-forward MIMO relaying networks. IEEE Trans Veh Technol, 2017, 66: 2059–2071
16 Guo S Z, Xing C W, Fei Z S, et al. Robust capacity maximization transceiver design for MIMO OFDM systems. Sci China Inf Sci, 2016, 59: 062301
17 Kaleva J, Berry R, Honig M, et al. Decentralized sum MSE minimization for coordinated multi-point transmission. In: Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing, Florence, 2014. 469–473
18 Li J, Wu J X, Peng M G, et al. Queue-aware energy-efficient joint remote radio head activation and beamforming in cloud radio access networks. IEEE Trans Wirel Commun, 2016, 15: 3880–3894
19 Cui Q M, Gu Y, Ni W, et al. Effective capacity of licensed-assisted access in unlicensed spectrum for 5G: from theory to application. IEEE J Sel Areas Commun, 2017, 35: 1754–1767
20 Cui Q M, Yuan T P, Ni W. Energy-efficient two-way relaying under non-ideal power amplifiers. IEEE Trans Veh Technol, 2017, 66: 1257–1270
21 Kumar S, Jain R, Rajawat K. Asynchronous optimization over heterogeneous networks via consensus ADMM. IEEE Trans Signal Inf Process Networks, 2017, 3: 114–129
22 Mota J F C, Xavier J M F, Aguiar P M Q, et al. D-ADMM: a communication-efficient distributed algorithm for separable optimization. IEEE Trans Signal Process, 2013, 61: 2718–2723
23 Hong M Y. A distributed, asynchronous and incremental algorithm for nonconvex optimization: an ADMM approach. IEEE Trans Control Netw Syst, 2017. doi: 10.1109/TCNS.2017.2657460
24 Chang T H, Hong M Y, Liao W C, et al. Asynchronous distributed ADMM for large-scale optimization part I: algorithm and convergence analysis. IEEE Trans Signal Process, 2016, 64: 3118–3130
25 Chang T H, Liao W C, Hong M Y, et al. Asynchronous distributed ADMM for large-scale optimization-part II: linear convergence analysis and numerical performance. IEEE Trans Signal Process, 2016, 64: 3131–3144
26 Christensen S S, Agarwal R, Carvalho E, et al. Weighted sum-rate maximization using weighted MMSE for MIMO-BC beamforming design. IEEE Trans Wirel Commun, 2008, 7: 4792–4799
27 Boyd S. Distributed optimization and statistical learning via the alternating direction method of multipliers. FNT Mach Learn, 2010, 3: 1–122
28 Palomar D P, Chiang M. A tutorial on decomposition methods for network utility maximization. IEEE J Sel Areas Commun, 2006, 24: 1439–1451
29 Bertsekas D P, Tsitsiklis J N. Parallel and Distributed Computation: Numerical Methods. Englewood Cliffs: Prentice Hall, 1989