

Low Complexity SLP: An Inversion-Free, Parallelizable ADMM Approach

Junwen Yang, Ang Li, *Senior Member, IEEE*, Xuewen Liao, *Member, IEEE*,
and Christos Masouros, *Senior Member, IEEE*

Abstract

We propose a parallel constructive interference (CI)-based symbol-level precoding (SLP) approach for massive connectivity in the downlink of multiuser multiple-input single-output (MU-MISO) systems, with only local channel state information (CSI) used at each processor unit and limited information exchange between processor units. By reformulating the power minimization (PM) SLP problem and exploiting the separability of the corresponding reformulation, the original problem is decomposed into several parallel subproblems via the ADMM framework with closed-form solutions, leading to a substantial reduction in computational complexity. The sufficient condition for guaranteeing the convergence of the proposed approach is derived, based on which an adaptive parameter tuning strategy is proposed to accelerate the convergence rate. To avoid the large-dimension matrix inverse operation, an efficient algorithm is proposed by employing the standard proximal term and by leveraging the singular value decomposition (SVD). Furthermore, a prox-linear proximal term is adopted to fully eliminate the matrix inversion, and a parallel inverse-free SLP (PIF-SLP) algorithm is finally obtained. Numerical results validate our derivations above, and demonstrate that the proposed PIF-SLP algorithm can significantly reduce the computational complexity compared to the state-of-the-arts.

Index Terms

Manuscript received XXXX; revised XXXX. (*Corresponding author: Xuewen Liao.*)

J. Yang, A. Li are with the School of Information and Communications Engineering, Faculty of Electronic and Information Engineering, Xi'an Jiaotong University, Xi'an, Shaanxi 710049, China (e-mail: jwyang@stu.xjtu.edu.cn; ang.li.2020@xjtu.edu.cn).

X. Liao is with the School of Information and Communications Engineering, Faculty of Electronic and Information Engineering, Xi'an Jiaotong University, Xi'an, Shaanxi 710049, China, and also with the National Mobile Communications Research Laboratory, Southeast University, Nanjing 210096, China (e-mail: yeplos@mail.xjtu.edu.cn).

C. Masouros is with the Department of Electronic and Electrical Engineering, University College London, London WC1E 7JE, U.K. (e-mail: c.masouros@ucl.ac.uk).

I. INTRODUCTION

Massive multiuser multiple-input multiple-output (M-MU-MIMO) has been foreseen as one of the key enablers for future wireless communication systems [1]–[3], as it has the potential to offer tremendous multiplexing gain and array gain, thereby meeting the boosting requirements of spectral efficiency and energy efficiency [4], [5]. As a fundamental factor that affects system performance, interference plays a central role in reaping the benefits of M-MU-MIMO, and needs to be dealt with carefully. As an effective interference management technique in the downlink, precoding has attracted extensive attention [6].

Maximum ratio transmission (MRT) precoding is the simplest strategy that maximizes the received signal-to-noise ratio (SNR) [7]. MRT is devised for noise-limited scenarios, when it comes to interference-limited scenarios, zero-forcing (ZF) precoding is a preferable choice [8], which employs the channel inversion to eliminate the multiuser interference at the price of augmenting noise. As a regularized form of channel inversion, regularized ZF (RZF) precoding is proposed to alleviate the noise-amplifying effect of ZF [9]. The aforementioned linear precoding methods are close to optimal only when the number of transmit antennas is far greater than the number of users [10], because in such case the channels of users are asymptotically orthogonal and favorable propagation can be achieved [1]. On the other hand, as the number of users keeps increasing, there will be a large spread in the singular value of the channel matrix [9], which will dramatically deteriorate the performance of linear precoding methods.

Except for closed-form linear precoding methods described above, there also exist a number of nonlinear precoding approaches in the literature. Dirty paper coding (DPC) is a capacity-achieving nonlinear precoding method that cancels known interference sequentially leveraging the full channel state information (CSI) [11]. Nevertheless, the unrealistic assumption of infinite codebook length hinders the practical implementation of DPC. An interference cancellation alternative is the Tomlinson-Harashima precoding (THP), which imposes an integer offset at the transmitter, and a modulo operation is required for the received signal [12]. Despite the near-capacity performance of THP, its encoding and decoding are of great complexity. Instead of sequentially calculating the offset in THP, another nonlinear precoding, the vector perturbation (VP) precoding, jointly selects a perturbation vector via the sphere encoding algorithm and

transmits the perturbed signals to the users, which is shown to achieve superior performance and requires a relatively simpler decoding procedure than THP [13].

In addition to devising the precoder heuristically or analytically, pursuing the optimal precoding strategy naturally resorts to optimization. For example, the SINR-constrained power minimization (PM) problem aims to minimize the total transmit power, subject to the received SINR target for each user [14]. This problem can be solved via the uplink-downlink duality [14] or conic programming [15], and also can be reformulated into a semidefinite optimization, for which the semidefinite relaxation approach is viable [16]. The inverse problem of PM is the max-min SINR balancing (SB) problem, which maximizes the minimum SINR subject to a total transmit power constraint [15], [17], [18]. The SB problem is nonconvex and cannot be reformulated to a convex form. Stimulated by the relationship between SB and PM, iterative algorithms that solve a series of PM problems in a bi-section search have been proposed in the literature to solve the SB problem [15], [19]. The nonlinear precoding methods are also known as symbol-level precoding (SLP), because their precoding matrices are jointly determined by the CSI and data symbols, and generally redesigned for each symbol slot. From a statistical perspective, the interference is uncontrollable and performs as a deterioration factor, and thereby BLP methods aim to mitigate or eliminate interference. On the other hand, from an instantaneous view, interference is controllable and can be manipulated to enhance signal detection by means of SLP. This was first discussed by the constructive interference (CI) precoding in the context of pre-decorrelation and Pre-Rake [20]. The same concept was introduced to ZF precoding later in [21]. As a step further, a correlation rotation precoding technique was designed to rotate both CI and destructive interference (DI) such that the phase of interference is aligned to the signal of interest, based on which DI can be transferred into CI [22]. The first work to combine CI precoding with optimization was proposed in the context of VP precoding with limited feedback. Standing on the concept of CI, the optimization-based PM-SLP and SB-SLP schemes were further studied, where the resulting interference is no longer strictly aligned to the signal of interest, but constrained by the CI regions [23], which provides further performance improvements. At the same time, this means that the SLP methods must solve a constrained optimization problem at each symbol slot to obtain the full benefits offered by CI-SLP, resulting substantial computational complexity, especially in M-MU-MIMO settings.

Towards low-complexity and low-latency CI-SLP solutions, plenty of works have endeavored to find efficient and practical SLP solutions. For PM-SLP, the virtual multicast formulation is

widely used, by which the optimization variable is shifted from the large-dimension precoding matrix to the small-dimension precoded signal vector [23]–[25]. Subsequently, Lagrange duality is applied to inspect the reformulated problem, whose Lagrangian dual is identified as a nonnegative least-squares (NNLS) problem. A gradient projection algorithm with line search was proposed to solve the NNLS problem [23]. With further inspection, the structure of the optimal solution for PM-SLP was analyzed via the Karush–Kuhn–Tucker (KKT) optimality conditions, which leads to a closed-form suboptimal solution for the NNLS problem [24]. To improve the approximation performance of the suboptimal solution, its improved alternative with an extra validation step was proposed [25]. For SB-SLP, its Lagrangian dual was shown to be a QP optimization over a probability simplex, and the optimal structure of the precoding matrix was derived, based on which a closed-form iterative algorithm with conditional optimality was developed [26], [27]. On the other hand, deep learning-based low-complexity SLP frameworks, such as CI-NN and SLP-DNet were also proposed [28], [29].

Based on the above descriptions, it can be summarized that most of the existing CI-based precoding approaches need sequential and centralized implementations, while the closed-form suboptimal solutions suffer from performance losses. More importantly, the resource-demanding matrix inverse operation is commonly required, so the resulting complexity of SLP is still high, which hinders its practical implementation. Motivated by these findings, in this paper, we propose a parallel inverse-free SLP (PIF-SLP) scheme for M-MU-MIMO downlink. The main contributions of the paper are summarized as follows.

- 1) We propose a parallel CI-SLP approach based on the proximal Jacobian alternating direction method of multipliers (PJ-ADMM) for the PM-SLP problem. For the first time in literature, we take advantage of the separable structure of the PM-SLP problem, and by transferring the inequality constraints into equality constraints with the introduced slack variable vector, the original problem is formulated into an unconstrained problem using the augmented Lagrangian method (ALM). The PJ-ADMM framework is adopted to decouple the unconstrained problem into a series of parallel subproblems, and closed-form solutions are obtained for each subproblem.
- 2) We analyze the convergence performance of the proposed parallel CI-SLP approach, and derive the sufficient condition for convergence, which indicates that the parallel SLP approach is guaranteed to converge to the global optimum as long as the proximal term is chosen sufficiently large. However, a larger proximal term will result in a slower

convergence rate. Accordingly, an adaptive parameter tuning strategy is developed to speed up the convergence.

- 3) Based on the above, we propose the PIF-SLP algorithm by adapting a prox-linear proximal term to avoid the matrix inverse operation for further complexity reduction. Specifically, the Hessian of the quadratic penalty term in the Lagrangian function is approximated with an identical proximal matrix, and hence the corresponding matrix inversion can be replaced by scalar division. Meanwhile, the required number of matrix multiplication is also reduced.
- 4) We further propose a low-coordination overhead decentralized scheme to alleviate the coordination overhead, at the cost of slightly increased computational overhead. By rearranging the closed-form solutions of the subproblems and incorporating the updates of global variables in the parallel processor units, the extra consensus node is removed, which reduces the coordination overhead. The computational complexity and the coordination overhead between processor units of the parallel SLP approach are also studied analytically.

Monte Carlo simulations are conducted to validate our analysis as well as the effectiveness of the proposed schemes, where it is demonstrated that the proposed PIF-SLP algorithm can greatly reduce the computational burden of the CI-SLP without performance loss. A scalable complexity-performance trade-off of the parallel SLP approach is also observed.

The remainder of this paper is organized as follows. Section II introduces the system model and CI, as well as the canonical PM-SLP problem formulation. Section III reformulates the canonical problem based on separability and slackness, where ALM and ADMM are further introduced. The proposed parallel SLP approach and its sufficient condition of convergence are presented in section IV, including the adaptive parameter tuning strategy and the final PIF-SLP algorithm. Section V provides the computational complexity and coordination overhead analysis. Numerical results are presented in Section VI, and Section VII concludes the paper.

Notation: Scalars, vectors, and matrices, are represented by plain lowercase, boldface lowercase, and boldface capital letters, respectively. $(\cdot)^T$, $(\cdot)^H$, and $(\cdot)^{-1}$ denote transpose, conjugate transpose, and inverse operators, respectively. $\mathbb{C}^{M \times N}$ and $\mathbb{R}^{M \times N}$ denote the sets of $M \times N$ matrices with complex and real entries, respectively. $|\cdot|$ represents the absolute value of a real scalar or the modulus of a complex scalar. $\|\cdot\|$ denotes the Euclidean norm of a vector or spectral norm of a matrix. $\Re\{\cdot\}$ and $\Im\{\cdot\}$ respectively denote the real part and imaginary part of a complex input. \succeq denotes element-wise inequality. $\mathbf{0}$, $\mathbf{1}$, and \mathbf{I} represent respectively, the

all-zeros vector, the all-ones vector, and the identity matrix with appropriate dimensions. $\max\{\cdot\}$ represents the elementwise maximum. \oslash denotes the element-wise division. $\text{diag}\{\cdot\}$ returns a vector consisting of the main diagonal elements of a input matrix.

II. SYSTEM MODEL AND PROBLEM FORMULATION

A. System Model

We consider a downlink massive MU-MISO system, where a base station (BS) equipped with N_t antennas serves K single-antenna users in the same time-frequency resource. The modulated data symbol vector $\tilde{\mathbf{s}} \triangleq [\tilde{s}_1, \dots, \tilde{s}_K]^T \in \mathbb{C}^K$ is composed of K independent symbols randomly drawn from a normalized \mathcal{M} -ary PSK constellation, which is mapped to the transmit signal $\tilde{\mathbf{x}} \triangleq [\tilde{x}_1, \dots, \tilde{x}_{N_t}]^T \in \mathbb{C}^{N_t}$ at the BS via SLP. The received signal of user k is expressed as

$$y_k = \tilde{\mathbf{h}}_k^T \tilde{\mathbf{x}} + n_k, \quad (1)$$

where $\tilde{\mathbf{h}}_k \in \mathbb{C}^{N_t}$ denotes the quasi-static Rayleigh flat-fading channel vector between BS and user k , and n_k is the circularly symmetric complex zero-mean Gaussian white noise with variance σ_k^2 at user k . The above signal model can be written in a more compact form as

$$\mathbf{y} = \tilde{\mathbf{H}}\tilde{\mathbf{x}} + \mathbf{n}, \quad (2)$$

where $\mathbf{y} \triangleq [y_1, \dots, y_K]^T \in \mathbb{C}^K$ and $\mathbf{n} \triangleq [n_1, \dots, n_K]^T \in \mathbb{C}^K$ denote the received signal and noise at all K users, respectively. $\tilde{\mathbf{H}} \triangleq [\tilde{\mathbf{h}}_1, \dots, \tilde{\mathbf{h}}_K]^T \in \mathbb{C}^{K \times N_t}$ is the channel matrix. To focus on the precoding design, perfect CSI is assumed throughout this paper.

B. Constructive Interference

CI precoding was first introduced in [20], which reveals that the constructive and destructive interference pattern of the noiseless received signal $\{\tilde{\mathbf{h}}_k^T \tilde{\mathbf{x}}\}$ is jointly determined by CSI and data symbols. Based on this fact, interference can be predicted and further exploited using SLP, which judiciously utilizes CSI and data symbols to optimize the transmit signal, such that all the multiuser interference add up constructively at receivers [30]. Therefore, the received instantaneous SINR at user k is given as $\text{SINR}_k \triangleq \frac{|\tilde{\mathbf{h}}_k^T \tilde{\mathbf{x}}|^2}{\sigma_k^2}$. Since all interference is exploited via SLP, the SINR is equivalent to the conventional signal-to-noise ratio (SNR).

Geometrically, CI is achieved as long as the noiseless received signal of each user lies in the symbol-specified CI region in the complex plane, where the CI region refers to a polyhedron

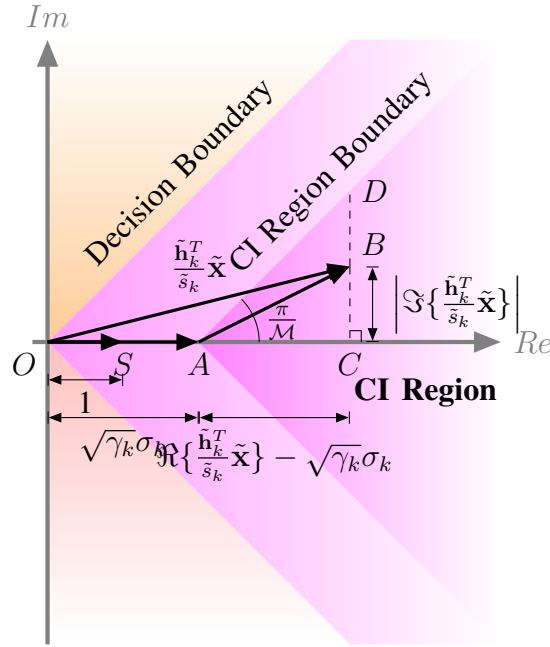


Fig. 1. Illustration of CI regions for a generic \mathcal{M} -PSK modulation.

bounded by hyperplanes parallel to decision boundaries or Voronoi edges [23], [31], and the only vertex of one CI region is the SINR threshold-dependent nominal constellation symbol, as depicted in Fig. 1. Without loss of generality, let \tilde{s}_k be the symbol of interest for user k , which is an arbitrary constellation point drawn from a normalized \mathcal{M} -PSK constellation as described in Section II-A. We rotate \tilde{s}_k to the positive real axis, thereby the rotated symbol is 1, which corresponds to \overrightarrow{OS} in Fig. 1. Other related signals are rotated by the same phase. Consequently, the received noiseless signal of user k , $\tilde{\mathbf{h}}_k^T \tilde{\mathbf{x}}$, turns out to $\frac{\tilde{\mathbf{h}}_k^T \tilde{\mathbf{x}}}{\tilde{s}_k}$, which is denoted by \overrightarrow{OB} in Fig. 1. For a given instantaneous SINR threshold γ_k for user k , the nominal constellation point is equivalent to $\sqrt{\gamma_k} \sigma_k \tilde{s}_k$. We introduce \overrightarrow{OA} as the rotated nominal constellation point, which is also the only vertex of the interested CI region. When \overrightarrow{OB} is located in the depicted CI region, the received signal is pushed away from decision boundaries and the instantaneous SINR is guaranteed to be no less than the prescribed threshold γ_k . One of the criteria that specifies the location of \overrightarrow{OB} in the CI region is $|\overrightarrow{CD}| \geq |\overrightarrow{CB}|$. Accordingly, the corresponding explicit mathematical formulation of CI constraints for \mathcal{M} -PSK signaling can be written as

$$\Re \left\{ \hat{\mathbf{h}}_k^T \tilde{\mathbf{x}} \right\} - \frac{\left| \Im \left\{ \hat{\mathbf{h}}_k^T \tilde{\mathbf{x}} \right\} \right|}{\tan \frac{\pi}{\mathcal{M}}} \geq \sqrt{\gamma_k} \sigma_k, \forall k, \quad (3)$$

where $\hat{\mathbf{h}}_k^T \triangleq \frac{\tilde{\mathbf{h}}_k^T}{\tilde{s}_k}$, γ_k denotes the pre-defined instantaneous SINR threshold for user k . It is worth noting that the CI constraint for each user already incorporates the SINR constraint.¹

C. SLP for Power Minimization

Throughout this paper, we are interested in minimizing the total transmit power subject to CI constraints, which is known as the PM-SLP problem. This optimization problem can be formulated as

$$\begin{aligned} \mathcal{P}_1 : \min_{\tilde{\mathbf{x}}} \|\tilde{\mathbf{x}}\|^2 \\ \text{s.t. } \Re \left\{ \hat{\mathbf{h}}_k^T \tilde{\mathbf{x}} \right\} - \frac{\left| \Im \left\{ \hat{\mathbf{h}}_k^T \tilde{\mathbf{x}} \right\} \right|}{\tan \frac{\pi}{\mathcal{M}}} \geq \sqrt{\gamma_k} \sigma_k, \forall k. \end{aligned} \quad (4)$$

The quadratic objective function and linear constraints indicate that this problem is convex, and hence can be handled via off-the-shelf solvers. Unfortunately, most generic solvers, e.g., SeDuMi and SDPT3, are based on the high-complexity interior-point method (IPM). To alleviate the computational burden, efficient algorithms based on gradient projection method [23], suboptimal closed-form solution [24], and improved suboptimal closed-form solution [25] were proposed. Existing works, however, focus on centralized iterative algorithms and ignore the separable nature of the PM-SLP problem. By exploiting such separability, we propose a parallel CI-SLP precoding approach in this paper based on the PJ-ADMM, as shown below.

III. ALM AND CONVENTIONAL ADMM FOR PM-SLP

In this section, we investigate the structure of the PM-SLP optimization problem and reveal its separable nature. ALM is used to tackle the reformulated problem subsequently. Conventional Gauss-Seidel ADMM and Jacobian ADMM are further employed to exploit the separability to arrive at sequential and parallel solutions. In the next section, we present the proposed PIF-SLP approach.

A. Separability and Slackness

The real-valued equivalence of \mathcal{P}_1 can be written as

$$\begin{aligned} \mathcal{P}_2 : \min_{\mathbf{x}} \|\mathbf{x}\|^2 \\ \text{s.t. } \mathbf{N}\mathbf{S}_k \mathbf{H}_k \mathbf{x} \succeq \sqrt{\gamma_k} \sigma_k \mathbf{1}, \forall k, \end{aligned} \quad (5)$$

¹The CI constraints can be readily extended to multi-level modulation such as QAM by employing the symbol-scaling metric [32].

where $\mathbf{x} \triangleq \begin{bmatrix} \Re\{\tilde{\mathbf{x}}\} \\ \Im\{\tilde{\mathbf{x}}\} \end{bmatrix} \in \mathbb{R}^{2N_t}$, $\mathbf{N} \triangleq \begin{bmatrix} 1 & -\frac{1}{\tan \frac{\pi}{M}} \\ 1 & \frac{1}{\tan \frac{\pi}{M}} \end{bmatrix} \in \mathbb{R}^{2 \times 2}$, $\mathbf{S}_k \triangleq \begin{bmatrix} \Re\left\{\frac{1}{\tilde{s}_k}\right\} & -\Im\left\{\frac{1}{\tilde{s}_k}\right\} \\ \Im\left\{\frac{1}{\tilde{s}_k}\right\} & \Re\left\{\frac{1}{\tilde{s}_k}\right\} \end{bmatrix} \in \mathbb{R}^{2 \times 2}$, and $\mathbf{H}_k \triangleq \begin{bmatrix} \Re\left\{\tilde{\mathbf{h}}_k^T\right\} & -\Im\left\{\tilde{\mathbf{h}}_k^T\right\} \\ \Im\left\{\tilde{\mathbf{h}}_k^T\right\} & \Re\left\{\tilde{\mathbf{h}}_k^T\right\} \end{bmatrix} \in \mathbb{R}^{2 \times 2N_t}$. We further introduce $\bar{\mathbf{A}}_k \triangleq \mathbf{N}\mathbf{S}_k\mathbf{H}_k$, and $\mathbf{b}_k \triangleq \sqrt{\gamma_k}\sigma_k\mathbf{1}$. Accordingly, the CI constraints become

$$\bar{\mathbf{A}}_k \mathbf{x} \succeq \mathbf{b}_k, \forall k. \quad (6)$$

Stacking the CI constraints, the compact formulation can be written as

$$\mathbf{A}\mathbf{x} \succeq \mathbf{b}, \quad (7)$$

where $\mathbf{A} \triangleq [\bar{\mathbf{A}}_1^T, \dots, \bar{\mathbf{A}}_K^T]^T \in \mathbb{R}^{2K \times 2N_t}$, $\mathbf{b} \triangleq [\mathbf{b}_1^T, \dots, \mathbf{b}_K^T]^T \in \mathbb{R}^{2K}$. We can identify that the left-hand side of (7) can be expressed as a linear combination of the columns of \mathbf{A} , i.e., $\sum_{i=1}^{2N_t} \mathbf{a}_i x_i$, where \mathbf{a}_i is the i -th column of \mathbf{A} , x_i is the i -th entry of \mathbf{x} . Accordingly, \mathcal{P}_2 can be rearranged as

$$\begin{aligned} \mathcal{P}_3 : \min_{\mathbf{x}_i} & \sum_{i=1}^N \|\mathbf{x}_i\|^2 \\ \text{s.t.} & \sum_{i=1}^N \mathbf{A}_i \mathbf{x}_i \succeq \mathbf{b}, \end{aligned} \quad (8)$$

where $\mathbf{x}_i \in \mathbb{R}^{n_i}$ with $\sum_{i=1}^N n_i = 2N_t$ is the i -th block of \mathbf{x} , composed of the adjacent and/or disadjacent elements of \mathbf{x} , and $\mathbf{A}_i \in \mathbb{R}^{2K \times n_i}$ is the i -th column block of \mathbf{A} , each column of which is uniquely taken from the columns of \mathbf{A} . Mathematically, for the adjacent case, $\mathbf{x} = [\mathbf{x}_1^T, \dots, \mathbf{x}_N^T]^T$, $\mathbf{A} = [\mathbf{A}_1, \dots, \mathbf{A}_N]$, while for the disadjacent case, $\mathbf{x}_i = \mathbf{E}_i^T \mathbf{x}$, $\mathbf{A}_i = \mathbf{A}\mathbf{E}_i$, where $\mathbf{E}_i \in \mathbb{R}^{2N_t \times n_i}$, and each column of $\{\mathbf{E}_i\}$ is uniquely picked from the columns of the $2N_t \times 2N_t$ identity matrix. With such formulation, \mathcal{P}_3 is partitioned into N blocks, here we do not confine the number of blocks, so long as N is a positive integer not greater than $2N_t$.

We reformulate \mathcal{P}_3 by introducing a slack variable vector $\mathbf{c} \in \mathbb{R}_+^{2K}$ to replace the original inequality constraints as follows:

$$\begin{aligned} \mathcal{P}_4 : \min_{\mathbf{x}_i, \mathbf{c}} & \sum_{i=1}^N \|\mathbf{x}_i\|^2 \\ \text{s.t.} & \sum_{i=1}^N \mathbf{A}_i \mathbf{x}_i = \mathbf{b} + \mathbf{c}, \\ & \mathbf{c} \succeq \mathbf{0}. \end{aligned} \quad (9)$$

Since the feasible region of the slack variable \mathbf{c} is \mathbb{R}_+^{2K} , by introducing an indicator function, the nonnegativity constraints can be incorporated into the objective function:

$$\begin{aligned} \mathcal{P}_5 : \min_{\mathbf{x}_i, \mathbf{c}} & \sum_{i=1}^N \|\mathbf{x}_i\|^2 + \mathcal{I}_{\mathbb{R}_+^{2K}}(\mathbf{c}) \\ \text{s.t.} & -\sum_{i=1}^N \mathbf{A}_i \mathbf{x}_i + \mathbf{b} + \mathbf{c} = \mathbf{0}, \end{aligned} \quad (10)$$

where $\mathcal{I}_{\mathbb{R}_+^{2K}}$ is the indicator function of \mathbb{R}_+^{2K} given by

$$\mathcal{I}_{\mathbb{R}_+^{2K}}(\mathbf{c}) = \begin{cases} 0, & \text{if } \mathbf{c} \in \mathbb{R}_+^{2K}, \\ +\infty, & \text{otherwise.} \end{cases} \quad (11)$$

B. ALM

The corresponding augmented Lagrangian function of \mathcal{P}_5 is given by

$$\begin{aligned} \mathcal{L}_\rho(\mathbf{x}, \mathbf{c}, \boldsymbol{\lambda}) &= \sum_{i=1}^N \|\mathbf{x}_i\|^2 + \mathcal{I}_{\mathbb{R}_+^{2K}}(\mathbf{c}) + \boldsymbol{\lambda}^T \left(-\sum_{i=1}^N \mathbf{A}_i \mathbf{x}_i + \mathbf{b} + \mathbf{c} \right) + \frac{\rho}{2} \left\| -\sum_{i=1}^N \mathbf{A}_i \mathbf{x}_i + \mathbf{b} + \mathbf{c} \right\|^2 \\ &= \sum_{i=1}^N \|\mathbf{x}_i\|^2 + \mathcal{I}_{\mathbb{R}_+^{2K}}(\mathbf{c}) + \frac{\rho}{2} \left\| -\sum_{i=1}^N \mathbf{A}_i \mathbf{x}_i + \mathbf{b} + \mathbf{c} + \frac{\boldsymbol{\lambda}}{\rho} \right\|^2 - \frac{1}{2\rho} \|\boldsymbol{\lambda}\|^2 \end{aligned} \quad (12)$$

where $\boldsymbol{\lambda} \in \mathbb{R}_+^{2K}$ is the Lagrange multiplier vector, ρ is a positive penalty parameter that determines the severity of the quadratic penalty on constraint violations. When the value of $\boldsymbol{\lambda}$ is close to the optimal Lagrange multiplier, or the penalty parameter ρ is large, the optimal transmit signal vector \mathbf{x} of \mathcal{P}_5 can be well approximated by the unconstrained minima of the augmented Lagrangian [33]. Therefore, the original PM-SLP problem can be solved via the augmented Lagrangian method (ALM).

Starting with an arbitrary $\boldsymbol{\lambda}^0$, the ALM aims to update the multiplier vector iteratively to approximate the optimal dual solution. A common choice of such approximation is the following gradient iteration:

$$\boldsymbol{\lambda}^{t+1} = \boldsymbol{\lambda}^t + \rho^t (-\mathbf{A}\mathbf{x}^{t+1} + \mathbf{b} + \mathbf{c}^{t+1}), \quad (13)$$

where the superscript denotes the iteration index, and $(\mathbf{c}^{t+1}, \mathbf{x}^{t+1})$ is any vector that minimizes $\mathcal{L}_{\rho^t}(\mathbf{x}, \mathbf{c}, \boldsymbol{\lambda}^t)$, namely,

$$(\mathbf{c}^{t+1}, \mathbf{x}^{t+1}) = \arg \min_{\mathbf{c}, \mathbf{x}} \mathcal{L}_{\rho^t}(\mathbf{x}, \mathbf{c}, \boldsymbol{\lambda}^t). \quad (14)$$

The standard ALM guarantees global convergence with a theoretical linear convergence rate, while its convergence rate is in general faster in practical problems. Despite the promising convergence performance, the ALM involves a joint optimization of primal variables, and hence can not take advantage of the separability.

C. Gauss-Seidel ADMM

Given the current iteration variables $(\mathbf{c}^t, \mathbf{x}^t, \boldsymbol{\lambda}^t)$, the ADMM generates new iteration variables $(\mathbf{c}^{t+1}, \mathbf{x}^{t+1}, \boldsymbol{\lambda}^{t+1})$ via alternating optimization. Applying standard ADMM to the separable PM-SLP problem, $(\mathbf{c}^{t+1}, \mathbf{x}^{t+1}, \boldsymbol{\lambda}^{t+1})$ is updated via the following steps:

$$\mathbf{c}^{t+1} = \arg \min_{\mathbf{c}} \mathcal{L}_{\rho}(\mathbf{x}_1^t, \dots, \mathbf{x}_N^t, \mathbf{c}, \boldsymbol{\lambda}^t), \quad (15a)$$

$$\mathbf{x}_i^{t+1} = \arg \min_{\mathbf{x}_i} \mathcal{L}_{\rho}(\mathbf{x}_{<i}^{t+1}, \mathbf{x}_i, \mathbf{x}_{>i}^t, \mathbf{c}^{t+1}, \boldsymbol{\lambda}^t), \forall i, \quad (15b)$$

$$\boldsymbol{\lambda}^{t+1} = \boldsymbol{\lambda}^t + \rho \left(- \sum_{i=1}^N \mathbf{A}_i \mathbf{x}_i^{t+1} + \mathbf{b} + \mathbf{c}^{t+1} \right). \quad (15c)$$

We can observe that the transmit signal \mathbf{x}_i^{t+1} is calculated by a sweep of Gauss-Seidel updates, namely, \mathbf{x}_i^{t+1} is sequentially updated one after another. While the direct extended Gauss-Seidel ADMM does not necessarily converge for $N \geq 3$ [34], it is still efficient at solving many practical problems. Thanks to the strong convexity of the objective function of PM-SLP, Gauss-Seidel ADMM is applicable. On the other hand, however, although Gauss-Seidel ADMM is able to partition the original PM-SLP problem into several subproblems and allow distributed processing, parallel processing is still not achievable. Therefore, Gauss-Seidel ADMM is inefficient for large-scale MIMO precoding.

D. Jacobian ADMM

To enable parallel processing, Jacobian ADMM can be adopted, which minimizes the augmented Lagrangian with respect to $\mathbf{x}_1, \dots, \mathbf{x}_N$ in a parallel fashion, while keeping the updates of the remaining variables unchanged, given by

$$\mathbf{x}_i^{t+1} = \arg \min_{\mathbf{x}_i} \mathcal{L}_{\rho}(\mathbf{x}_i, \mathbf{x}_{\neq i}^t, \mathbf{c}^{t+1}, \boldsymbol{\lambda}^t), \forall i. \quad (16)$$

The above full decomposition and parallelization, however, is achieved at the expense of a degraded practical convergence performance compared to the Gauss-Seidel ADMM. It was shown

in [35] that the Jacobian ADMM iterations may be divergent, thus the output of the preceding Jacobian updates may not be used in the next iteration directly.

To design a Jacobian ADMM with guaranteed convergence, [35] suggests inserting an under-relaxation step between every two adjacent Jacobian ADMM iterates, given by

$$\mathbf{u}^{t+1} = \mathbf{u}^t - \alpha (\mathbf{u}^t - \bar{\mathbf{u}}^t), \quad (17)$$

where $\mathbf{u}^t \triangleq (\mathbf{x}_1^t, \mathbf{x}_2^t \cdots, \mathbf{x}_N^t, \boldsymbol{\lambda}^t)$, $\bar{\mathbf{u}}^t \triangleq (\bar{\mathbf{x}}_1^t, \bar{\mathbf{x}}_2^t \cdots, \bar{\mathbf{x}}_N^t, \bar{\boldsymbol{\lambda}}^t)$ denotes the output of the original Jacobian ADMM with the input \mathbf{u}^t , $\alpha > 0$ is a chosen step size. Note that, as an exactly updated intermediate variable, the slack variable \mathbf{c} is excluded from \mathbf{u} . The motivation for the above underrelaxation step lies in the fact that the Jacobian decomposition may have poor accuracy to approximate the joint optimization step of ALM, and [35] proposes to compensate for the accuracy loss by combining the last iterate \mathbf{u}^t with $\bar{\mathbf{u}}^t$ approximately in aid of the step size α . Its worst-case $O(1/t)$ convergence rate measured by the iteration complexity in both the ergodic and nonergodic senses is established.

Another way to enhance convergence of the Jacobian ADMM is to regularize each decomposed problem by a proximal term [36]. By adopting a proximal Jacobian decomposition method of ALM, the iteration steps are given by

$$\mathbf{c}^{t+1} = \arg \min_{\mathbf{c}} \mathcal{L}_{\rho} (\mathbf{x}_1^t, \cdots, \mathbf{x}_N^t, \mathbf{c}, \boldsymbol{\lambda}^t), \quad (18a)$$

$$\mathbf{x}_i^{t+1} = \arg \min_{\mathbf{x}_i} \mathcal{L}_{\rho} (\mathbf{x}_{\neq i}^t, \mathbf{x}_i, \mathbf{c}^{t+1}, \boldsymbol{\lambda}^t) + \frac{\tau \rho}{2} \|\mathbf{A}_i (\mathbf{x}_i - \mathbf{x}_i^t)\|^2, \forall i, \quad (18b)$$

$$\boldsymbol{\lambda}^{t+1} = \boldsymbol{\lambda}^t + \rho \left(- \sum_{i=1}^N \mathbf{A}_i \mathbf{x}_i^{t+1} + \mathbf{b} + \mathbf{c}^{t+1} \right), \quad (18c)$$

where $\tau > 0$ is a proximal coefficient that controls the proximity of the new iterate to the last one. It is essentially one type of proximal Jacobian ADMM, but to discriminate it from the PJ-ADMM in the next section and follow the terminology in [36], we refer it to PJ-ALM. It was shown in [36] that if the proximal coefficient is sufficiently large, i.e., $\tau \geq N - 1$, the convergence of the proximal Jacobian decomposition of ALM can be guaranteed.

IV. PROPOSED PIF-SLP APPROACH

In the previous section, we have revealed the separability of the original PM-SLP optimization problem (4) by inspecting and rearranging its structure to facilitate distributed and parallel processing. In this section, we adopt a more general and flexible PJ-ADMM framework in [37]

to solve the reformulated PM-SLP problem with closed-form solutions for each subproblem. PJ-ADMM is similar to the preceding relaxation and regularization idea, while its relaxation step is split into primal and dual relaxation, where the primal relaxation is replaced by a flexible quadratic proximal regularization term for each subproblem. The PJ-ADMM procedure for PM-SLP is formulated as

$$\mathbf{c}^{t+1} = \arg \min_{\mathbf{c}} \mathcal{L}_{\rho}(\mathbf{x}_1^t, \dots, \mathbf{x}_N^t, \mathbf{c}, \boldsymbol{\lambda}^t), \quad (19a)$$

$$\mathbf{x}_i^{t+1} = \arg \min_{\mathbf{x}_i} \mathcal{L}_{\rho}(\mathbf{x}_{\neq i}^t, \mathbf{x}_i, \mathbf{c}^{t+1}, \boldsymbol{\lambda}^t) + \frac{1}{2} \|\mathbf{x}_i - \mathbf{x}_i^t\|_{\mathbf{P}_i}^2, \forall i, \quad (19b)$$

$$\boldsymbol{\lambda}^{t+1} = \boldsymbol{\lambda}^t + \beta \rho \left(- \sum_{i=1}^N \mathbf{A}_i \mathbf{x}_i^{t+1} + \mathbf{b} + \mathbf{c}^{t+1} \right), \quad (19c)$$

where $\beta > 0$ is a damping parameter, \mathbf{P}_i is a symmetric and positive semi-definite matrix and $\|\mathbf{x}_i\|_{\mathbf{P}_i}^2 \triangleq \mathbf{x}_i^T \mathbf{P}_i \mathbf{x}_i$. Based on the above derivations, the original PM-SLP problem is decomposed, and each subproblem can be calculated in a parallel and distributed manner with (19). The global convergence with $o(1/t)$ convergence rate under certain conditions on $\{\mathbf{P}_i\}$ and β of PJ-ADMM can be guaranteed, as established in [37]. In what follows, we derive closed-form solutions for each subproblem in the PJ-ADMM iteration.

A. Closed-Form Solution for Each Subproblem of PJ-ADMM

The update for the slack variable \mathbf{c} can be written as

$$\mathbf{c}^{t+1} = \arg \min_{\mathbf{c} \in \mathbb{R}_+^{2K}} \frac{\rho}{2} \left\| - \sum_{i=1}^N \mathbf{A}_i \mathbf{x}_i^t + \mathbf{b} + \mathbf{c} + \frac{\boldsymbol{\lambda}^t}{\rho} \right\|^2, \quad (20)$$

which is equivalent to projecting the vector $\sum_{i=1}^N \mathbf{A}_i \mathbf{x}_i^t - \mathbf{b} - \frac{\boldsymbol{\lambda}^t}{\rho}$ onto \mathbb{R}_+^{2K} , denoted by

$$P_{\mathbb{R}_+^{2K}} \left(\sum_{i=1}^N \mathbf{A}_i \mathbf{x}_i^t - \mathbf{b} - \frac{\boldsymbol{\lambda}^t}{\rho} \right).$$

Its closed-form solution is given by

$$\mathbf{c}^{t+1} = \max \left\{ \sum_{i=1}^N \mathbf{A}_i \mathbf{x}_i^t - \mathbf{b} - \frac{\boldsymbol{\lambda}^t}{\rho}, \mathbf{0} \right\}. \quad (21)$$

The iteration for \mathbf{x}_i^{t+1} is updated as follows:

$$\mathbf{x}_i^{t+1} = \arg \min_{\mathbf{x}_i} \|\mathbf{x}_i\|^2 + \frac{\rho}{2} \left\| -\mathbf{A}_i \mathbf{x}_i - \sum_{j \neq i}^N \mathbf{A}_j \mathbf{x}_j^t + \mathbf{b} + \mathbf{c}^{t+1} + \frac{\boldsymbol{\lambda}^t}{\rho} \right\|^2 + \frac{1}{2} \|\mathbf{x}_i - \mathbf{x}_i^t\|_{\mathbf{P}_i}^2, \forall i, \quad (22)$$

which is an unconstrained quadratic programming, whose optimal solution can be obtained by setting the gradient of the objective function with respect to \mathbf{x}_i to zero, i.e.,

$$2\mathbf{x}_i + \rho\mathbf{A}_i^T \left(\mathbf{A}_i\mathbf{x}_i + \sum_{j \neq i}^N \mathbf{A}_j\mathbf{x}_j^t - \mathbf{b} - \mathbf{c}^{t+1} - \frac{\boldsymbol{\lambda}^t}{\rho} \right) + \mathbf{P}_i (\mathbf{x}_i - \mathbf{x}_i^t) = 0, \forall i. \quad (23)$$

After some calculation, the closed-form solution for \mathbf{x}_i^{t+1} can be written as

$$\mathbf{x}_i^{t+1} = (2\mathbf{I} + \rho\mathbf{A}_i^T\mathbf{A}_i + \mathbf{P}_i)^{-1} \left[\mathbf{P}_i\mathbf{x}_i^t + \rho\mathbf{A}_i^T \left(-\sum_{j \neq i}^N \mathbf{A}_j\mathbf{x}_j^t + \mathbf{b} + \mathbf{c}^{t+1} + \frac{\boldsymbol{\lambda}^t}{\rho} \right) \right], \forall i. \quad (24)$$

Note that when we take $N = 2N_t$, i.e., the transmit signal vector \mathbf{x} is decomposed into $2N_t$ scalars, \mathbf{A}_i reduces to a column vector \mathbf{a}_i , and \mathbf{P}_i reduces to a scalar p_i , then the update of the transmit signal can be carried out via $2N_t$ parallel and distributed scalar operations, i.e.,

$$x_i^{t+1} = \frac{p_i x_i^t + \rho \mathbf{a}_i^T \left(-\sum_{j \neq i}^{2N_t} \mathbf{a}_j x_j^t + \mathbf{b} + \mathbf{c}^{t+1} + \frac{\boldsymbol{\lambda}^t}{\rho} \right)}{2 + \rho \mathbf{a}_i^T \mathbf{a}_i + p_i}, \forall i. \quad (25)$$

If we group the real and imaginary parts of the same antenna's transmit signal into one block, the transmit signal vector will be decomposed into N_t blocks. Based on the structure of \mathbf{A} , we can find that $\mathbf{A}_i \in \mathbb{R}^{2K \times 2}$ is a matrix with orthogonal columns, which implies that the corresponding $\mathbf{A}_i^T \mathbf{A}_i$ is a 2×2 diagonal matrix with equal non-zero elements. Therefore, if we take \mathbf{P}_i as a diagonal matrix too, then the matrix inverse operation during the update of \mathbf{x}_i can be replaced by taking the reciprocals of the two entries in the main diagonal with reduced complexity, given by

$$\mathbf{x}_i^{t+1} = \left[\mathbf{P}_i\mathbf{x}_i^t + \rho\mathbf{A}_i^T \left(-\sum_{j \neq i}^{N_t} \mathbf{A}_j\mathbf{x}_j^t + \mathbf{b} + \mathbf{c}^{t+1} + \frac{\boldsymbol{\lambda}^t}{\rho} \right) \right] \oslash \mathbf{W}, \forall i, \quad (26)$$

where $\mathbf{W} \triangleq \text{diag} (2\mathbf{I} + \rho\mathbf{A}_i^T\mathbf{A}_i + \mathbf{P}_i)$.

B. Convergence Analysis

The global convergence of the PJ-ADMM for linear equality constraints is established in [37]. As shown in the preceding section, the linear inequality constraints of PM-SLP are reformulated into linear equality constraints with the aid of the slack variable \mathbf{c} , which is an exactly updated intermediate variable, thus not affecting convergence [38]. For the sake of completeness, the global convergence theorem of the PJ-ADMM PM-SLP is stated in the following.

Theorem 1: Let $\{\mathbf{u}^t\}$ be the sequence generated by (19) with arbitrary initialization. If there exists $\epsilon_i > 0$ such that

$$\mathbf{P}_i \succeq \rho \left(\frac{1}{\epsilon_i} - 1 \right) \mathbf{A}_i^T \mathbf{A}_i, \forall i, \sum_{i=1}^N \epsilon_i \leq 2 - \beta, \quad (27)$$

then $\{\mathbf{u}^t\}$ converges to a solution \mathbf{u}^* to the PM-SLP problem.

Proof: See Appendix A. ■

Furthermore, by choosing $\epsilon_i \leq \frac{2-\beta}{N}$, the sufficient condition can be rewritten as

$$\mathbf{P}_i \succeq \rho \left(\frac{N}{2-\beta} - 1 \right) \mathbf{A}_i^T \mathbf{A}_i, \forall i. \quad (28)$$

There are two special choices for \mathbf{P}_i as mentioned in [37]. The first one is termed the standard proximal, which takes the following form:

$$\mathbf{P}_i = \tau_i \mathbf{I}, \quad (29)$$

where $\tau_i \geq \rho \left(\frac{N}{2-\beta} - 1 \right) \|\mathbf{A}_i\|^2$. The other is termed the prox-linear proximal, which takes the following form:

$$\mathbf{P}_i = \tau_i \mathbf{I} - \rho \mathbf{A}_i^T \mathbf{A}_i, \quad (30)$$

where $\tau_i \geq \frac{\rho N}{2-\beta} \|\mathbf{A}_i\|^2$.

C. Adaptive Parameter Tuning Strategy

In the previous section, we have derived a sufficient condition to guarantee the convergence of the proposed algorithm, which provides a lower bound for the Hessian of the proximal term \mathbf{P}_i . However, the basic inequality (59) for bounding $\|\mathbf{u}\|_{\mathbf{Q}}^2$ is usually rather loose, so the sufficient condition may be fairly conservative in practical implementation [37].

In order to accelerate convergence, compared to adopting a constant proximal coefficient that satisfies the sufficient condition (28), it is more preferred to initialize \mathbf{P}_i with a relatively small value and increase it iteratively until $\|\mathbf{u}^t - \mathbf{u}^{t+1}\|_{\mathbf{Q}}^2 \geq 0$, i.e., adaptive tuning the proximal coefficient matrix \mathbf{P}_i . [37] proposed a heuristic scheme to tune the proximal coefficient matrix \mathbf{P}_i based on the exact value of $\|\mathbf{u}^t - \mathbf{u}^{t+1}\|_{\mathbf{Q}}^2$, specifically, the proximal coefficient is increased when \mathbf{Q} is not positive semi-definite, otherwise it will remain constant. The aforementioned analysis indicates that the resulting constant proximal coefficient does not necessarily satisfy the

sufficient condition (28). The details of the adaptive parameter tuning scheme for our proposed PJ-ADMM are summarized as follows:

$$\mathbf{P}_i^{t+1} = \begin{cases} \delta_i \mathbf{P}_i^t, & \text{if } \|\mathbf{u}^{t-1} - \mathbf{u}^t\|_{\mathbf{Q}}^2 < \eta \|\mathbf{u}^{t-1} - \mathbf{u}^t\|^2, \\ \mathbf{P}_i^t, & \text{otherwise,} \end{cases} \quad (31)$$

where $\delta_i > 1$, $\eta > 0$ is a sufficient small scalar.

D. Efficient and Inverse-Free Algorithms

For the standard proximal term (29), when $\mathbf{A}_i^T \mathbf{A}_i$ is a nondiagonal matrix, and the proximal parameter τ_i is tuned within each iteration, matrix inverse operation is required whenever τ_i is changed, thus the computational complexity of the algorithm is dominated by matrix inversion. The overall computational complexity can be further reduced by circumventing matrix inverse operation, which can be realized based on the fact that the eigenspace of the matrix to be inverted $\rho \mathbf{A}_i^T \mathbf{A}_i + (2 + \tau_i) \mathbf{I}$ is inherited from the eigenspace of $\mathbf{A}_i^T \mathbf{A}_i$ [39]. To be more specific, we firstly express the singular value decomposition (SVD) of $\mathbf{A}_i^T \mathbf{A}_i$ as

$$\mathbf{A}_i^T \mathbf{A}_i = \mathbf{U}_i \boldsymbol{\Sigma}_i \mathbf{V}_i^T, \quad (32)$$

where \mathbf{U}_i , \mathbf{V}_i are the right and left singular matrix, respectively, and \mathbf{U}_i is a unity matrix, i.e., $\mathbf{U}_i^{-1} = \mathbf{U}_i^T$. From the symmetry, we have $\mathbf{V}_i = \mathbf{U}_i$. $\boldsymbol{\Sigma}_i$ is a diagonal matrix of which the diagonal entries are singular values of $\mathbf{A}_i^T \mathbf{A}_i$. Based on the above, each \mathbf{x}_i can be efficiently updated by

$$\mathbf{x}_i^{t+1} = (2\mathbf{I} + \rho \mathbf{A}_i^T \mathbf{A}_i + \tau_i \mathbf{I})^{-1} \mathbf{r}_i = \mathbf{U}_i [\rho \boldsymbol{\Sigma}_i + (2 + \tau_i) \mathbf{I}]^{-1} \mathbf{U}_i^T \mathbf{r}_i = \mathbf{U}_i (\mathbf{U}_i^T \mathbf{r}_i \oslash \mathbf{q}_i), \forall i, \quad (33)$$

where $\mathbf{q}_i \triangleq \text{diag}\{\rho \boldsymbol{\Sigma}_i + (2 + \tau_i) \mathbf{I}\}$, $\mathbf{r}_i \triangleq \tau_i \mathbf{x}_i^t + \rho \mathbf{A}_i^T \left(-\sum_{j \neq i}^N \mathbf{A}_j \mathbf{x}_j^t + \mathbf{b} + \mathbf{c}^{t+1} + \frac{\boldsymbol{\lambda}^t}{\rho} \right)$.

With the above approach, the matrix inversion in each update of \mathbf{x}_i is replaced by one SVD in the first update and incremental matrix-vector multiplications in the remaining updates.

So far, the computational complexity induced by the adaptive parameter tuning strategy is alleviated by the preceding SVD-based efficient algorithm, thereby the constant proximal and the adaptive proximal only need one matrix inverse operation or SVD, respectively, both with $\mathcal{O}((2N_t)^3)$ complexity. For a massive MU-MIMO system equipped with hundreds of downlink transmit antennas, such complexity reduction is remarkable.

As a step further, we can observe that the matrix inversion is needed by the non-diagonal coefficient matrix $\rho \mathbf{A}_i^T \mathbf{A}_i$ of the quadratic penalty term in the augmented Lagrangian function

(12). Fortunately, flexible as the proximal term is, it can be used to subtract $\rho \mathbf{A}_i^T \mathbf{A}_i$, which means the matrix inverse operation can be fully eliminated. To devise such an inverse-free algorithm, we propose to construct a prox-linear proximal term as in (30), which linearizes the quadratic penalty term by approximating the Hessian $\rho \mathbf{A}_i^T \mathbf{A}_i$ of the quadratic penalty term with an identity proximal matrix $\tau_i \mathbf{I}$. Accordingly, the inverse-free closed-form solutions for the update of \mathbf{x}_i can be obtained by substituting (30) into (24), i.e.,

$$\mathbf{x}_i^{t+1} = \frac{1}{2 + \tau_i} \left[\tau_i \mathbf{x}_i^t + \rho \mathbf{A}_i^T \left(-\mathbf{A} \mathbf{x}^t + \mathbf{b} + \mathbf{c}^{t+1} + \frac{\boldsymbol{\lambda}^t}{\rho} \right) \right], \forall i. \quad (34)$$

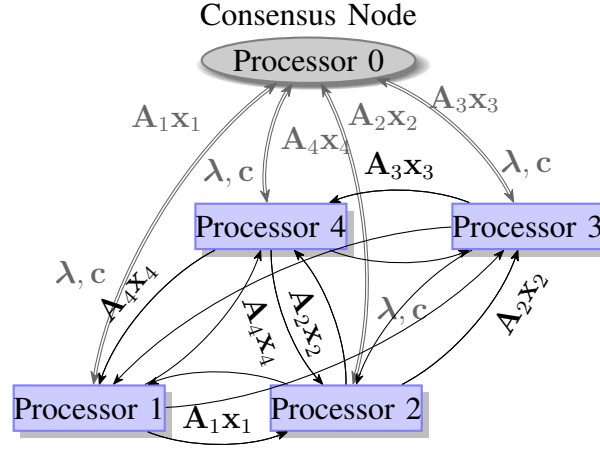
The computational complexity is mainly induced by the matrix-vector multiplications, any single matrix inversion or SVD is no longer demanded.

E. Decentralized Algorithm for Low-Coordination Overhead

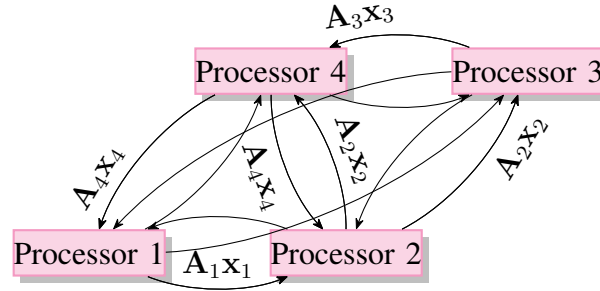
We can envision that the PIF-SLP algorithm can be implemented by a network of processor units connected by communication links [40], which will consume extra resources and cause time delay. In the preceding PJ-ADMM algorithm, iterations for the slack variable and Lagrangian multiplier vector need to be carried out at a central node, which requires extra information exchange with other processor units in the system to aggregate and propagate intermediate results. In such case, the time spent in exchanging information cannot be neglected. When real-time implementation and low-latency communication is required, we can reduce the coordination overhead by incorporating the slack variable iteration into the primal and dual variable iterations, and carrying out the multiplier iteration at the N blocks of transmit signal vector iterations. Another valuable feature is that the central node is no longer required, enabling us to further come up with a low-coordination overhead decentralized PIF-SLP algorithm. Without loss of generality, an example illustration of the centralized, as well as the decentralized system with $N = 4$ parallel processing units is shown in Fig. 2. For the centralized scheme shown in Fig. 2a, the processor unit 0 is working as a consensus node that collects the parallel processor units' results $\{\mathbf{A}_i \mathbf{x}_i\}$. As shown in Fig. 2b, consisting of the underlying 4 fully parallel processor units, the decentralized scheme does not need a central node and enjoys the lower coordination overhead.

For notation simplicity, we first denote $g(\mathbf{x}) \triangleq -\sum_{i=1}^N \mathbf{A}_i \mathbf{x}_i + \mathbf{b}$. The closed-form solution (21) for the slack variable \mathbf{c} can be rewritten as

$$\mathbf{c}^{t+1} = \max \left\{ -g(\mathbf{x}^t) - \frac{\boldsymbol{\lambda}^t}{\rho}, \mathbf{0} \right\}. \quad (35)$$



(a) Proposed parallel SLP approach with one centralized consensus node.



(b) Proposed parallel SLP approach with fully decentralized nodes.

Fig. 2. An example of parallel centralized and decentralized SLP systems.

Denoting $g^+(\mathbf{x}, \boldsymbol{\lambda}, \rho) \triangleq \max \left\{ g(\mathbf{x}), -\frac{\boldsymbol{\lambda}}{\rho} \right\}$, we have

$$g^+(\mathbf{x}, \boldsymbol{\lambda}, \rho) = g(\mathbf{x}) + \mathbf{c}. \quad (36)$$

Substituting (36) into (12), we can rewrite the augmented Lagrangian function as

$$\begin{aligned} \mathcal{L}_\rho(\mathbf{x}, \boldsymbol{\lambda}) &= \sum_{i=1}^N \|\mathbf{x}_i\|^2 + \boldsymbol{\lambda}^T \mathbf{g}^+(\mathbf{x}, \boldsymbol{\lambda}, \rho) + \frac{\rho}{2} \|\mathbf{g}^+(\mathbf{x}, \boldsymbol{\lambda}, \rho)\|^2 \\ &= \sum_{i=1}^N \|\mathbf{x}_i\|^2 + \frac{1}{2\rho} (\|\max \{\boldsymbol{\lambda} + \rho g(\mathbf{x}), \mathbf{0}\}\|^2 - \|\boldsymbol{\lambda}\|^2), \end{aligned} \quad (37)$$

where the penalty term $\frac{1}{2\rho} (\|\max \{\boldsymbol{\lambda} + \rho g(\mathbf{x}), \mathbf{0}\}\|^2 - \|\boldsymbol{\lambda}\|^2)$ corresponding to the inequality CI constraints is continuously differentiable with respect to \mathbf{x} as $g(\mathbf{x})$ is continuously differentiable [41]. Following the preceding procedure, we can obtain closed-form expressions similar to the

Algorithm 1 Proposed Low-Coordination Overhead Decentralized PIF-SLP Algorithm

Input: \mathbf{A} , \mathbf{b} , ρ , η , $\{\delta_i\}_{i=1}^N$
Output: \mathbf{x}

- 1: Initialize \mathbf{x}_i^0 ($i = 1, \dots, N$), $\boldsymbol{\lambda}^0$ and τ_i^0 ($i = 1, \dots, N$);
 - 2: **for** $t = 0, 1, \dots$ **do**
 - 3: Update \mathbf{x}_i^{t+1} for $i = 1, \dots, N$ in parallel by (39a);
 - 4: Share $\mathbf{A}_i \mathbf{x}_i^{t+1}$;
 - 5: Collect $\{\mathbf{A}_j \mathbf{x}_j^{t+1}\}_{j \neq i}$;
 - 6: Update $\boldsymbol{\lambda}^{t+1}$ for $i = 1, \dots, N$ in parallel by (39b);
 - 7: **if** $\|\mathbf{u}^t - \mathbf{u}^{t+1}\|_{\mathbf{Q}}^2 < \eta \|\mathbf{u}^t - \mathbf{u}^{t+1}\|^2$ **then**
 - 8: $\tau_i \leftarrow \delta_i \tau_i$;
 - 9: Backtrack $\mathbf{u}^{t+1} \leftarrow \mathbf{u}^t$;
 - 10: **end if**
 - 11: **end for**
-

preceding PJ-ADMM. By substituting (36) into the update for \mathbf{x}_i in (24) and $\boldsymbol{\lambda}$ in (19c), \mathbf{x}_i^{t+1} and $\boldsymbol{\lambda}^{t+1}$ can be further expressed as

$$\mathbf{x}_i^{t+1} = (2\mathbf{I} + \rho \mathbf{A}_i^T \mathbf{A}_i + \mathbf{P}_i)^{-1} \left[\mathbf{P}_i \mathbf{x}_i^t + \rho \mathbf{A}_i^T \left(\mathbf{g}^+(\mathbf{x}^t, \boldsymbol{\lambda}^t, \rho) + \mathbf{A}_i \mathbf{x}_i^t + \frac{\boldsymbol{\lambda}^t}{\rho} \right) \right], \forall i, \quad (38a)$$

$$\boldsymbol{\lambda}^{t+1} = \boldsymbol{\lambda}^t + \beta \rho \mathbf{g}^+(\mathbf{x}^{t+1}, \boldsymbol{\lambda}^t, \rho). \quad (38b)$$

A slight difference between PJ-ADMM and its decentralized counterpart during iteration lies in that the latter first uses the updated transmit signal vector to reformulate the slack variable \mathbf{c} , based on which the multiplier vector $\boldsymbol{\lambda}$ is updated.

For completeness, we further propose the decentralized PIF-SLP algorithm, which can be obtained by substituting the prox-linear proximal term (30) into (38), given by

$$\mathbf{x}_i^{t+1} = \frac{1}{2 + \tau_i} \left(\tau_i \mathbf{x}_i^t + \rho \mathbf{A}_i^T \left(\mathbf{g}^+(\mathbf{x}^t, \boldsymbol{\lambda}^t, \rho) + \frac{\boldsymbol{\lambda}^t}{\rho} \right) \right), \forall i, \quad (39a)$$

$$\boldsymbol{\lambda}^{t+1} = \boldsymbol{\lambda}^t + \beta \rho \mathbf{g}^+(\mathbf{x}^{t+1}, \boldsymbol{\lambda}^t, \rho). \quad (39b)$$

The corresponding algorithm is summarized in Algorithm 1.

V. COMPLEXITY AND OVERHEAD ANALYSIS

A. Computational Complexity

We evaluate the computational complexity of the proposed parallel SLP approach, including the matrix inversion-based plain implementation, the SVD-based efficient algorithm, and the PIF-SLP algorithm by counting float count operations in this section. Define the flop-count operator $\mathcal{F}(\mathbf{z}|\mathbf{y})$ as the number of flops to compute \mathbf{z} given \mathbf{y} . Assume that the dimensions of the parallel subproblems are equal, and let m, n be the row and column dimensions of \mathbf{A}_i respectively, i.e., $m = 2K, n = 2N_t/N$. When no matrix structure is exploited, matrix inversion is required in each iteration, then the straight and naive implementation of the proposed parallel PM-SLP approach costs

$$\begin{aligned} \mathcal{F}(\boldsymbol{\lambda}^{t+1}|\boldsymbol{\lambda}^t) &= \mathcal{F}(\mathbf{c}^{t+1} | (\mathbf{A}_i \mathbf{x}_i^t, \boldsymbol{\lambda}^t)) + \mathcal{F}(\mathbf{A}_i \mathbf{x}_i^{t+1} | (\mathbf{A}_i \mathbf{x}_i^t, \mathbf{c}^{t+1}, \boldsymbol{\lambda}^t)) \\ &\quad + \mathcal{F}(\boldsymbol{\lambda}^{t+1} | (\boldsymbol{\lambda}^t, \mathbf{A}_i \mathbf{x}_i^{t+1}, \mathbf{c}^{t+1})) \\ &= \mathcal{O}(m) + \mathcal{O}((m+n)n^2) + \mathcal{O}((m+n)n) + \mathcal{O}(m) \end{aligned} \quad (40)$$

flops per iteration. The dominant terms are caused by matrix inversion, matrix-matrix multiplication, and matrix-vector multiplication during the update of \mathbf{x}_i . As discussed in Section IV-D, when SVD is precomputed and used in the subsequent iterations, then the SVD-based efficient algorithm costs

$$\begin{aligned} \mathcal{F}(\boldsymbol{\lambda}^{t+1}|\boldsymbol{\lambda}^t) &= \mathcal{F}(\mathbf{c}^{t+1} | (\mathbf{A}_i \mathbf{x}_i^t, \boldsymbol{\lambda}^t)) + \mathcal{F}(\mathbf{A}_i \mathbf{x}_i^{t+1} | (\mathbf{A}_i \mathbf{x}_i^t, \mathbf{c}^{t+1}, \boldsymbol{\lambda}^t, \mathbf{U}_i, \boldsymbol{\Sigma}_i)) \\ &\quad + \mathcal{F}(\boldsymbol{\lambda}^{t+1} | (\boldsymbol{\lambda}^t, \mathbf{A}_i \mathbf{x}_i^{t+1}, \mathbf{c}^{t+1})) \\ &= \mathcal{O}(m) + \mathcal{O}((m+n)n) + \mathcal{O}(m) \end{aligned} \quad (41)$$

flops per iteration. The dominant terms of the efficient implementation turn to matrix-vector multiplication since matrix inversion and matrix-matrix multiplication are both eliminated. The PIF-SLP algorithm costs

$$\begin{aligned} \mathcal{F}(\boldsymbol{\lambda}^{t+1}|\boldsymbol{\lambda}^t) &= \mathcal{F}(\mathbf{c}^{t+1} | (\mathbf{A}_i \mathbf{x}_i^t, \boldsymbol{\lambda}^t)) + \mathcal{F}(\mathbf{A}_i \mathbf{x}_i^{t+1} | (\mathbf{A}_i \mathbf{x}_i^t, \mathbf{c}^{t+1}, \boldsymbol{\lambda}^t)) \\ &\quad + \mathcal{F}(\boldsymbol{\lambda}^{t+1} | (\boldsymbol{\lambda}^t, \mathbf{A}_i \mathbf{x}_i^{t+1}, \mathbf{c}^{t+1})) \\ &= \mathcal{O}(m) + \mathcal{O}((m+1)n) + \mathcal{O}(m) \end{aligned} \quad (42)$$

flops per iteration. It can be observed that the PIF-SLP algorithm is not only free of matrix inversion or SVD but also requires fewer matrix multiplications compared to the other two algorithms.

B. Coordination Overhead

For the parallel SLP scheme with a consensus node, the iteration needs $N + 1$ processor units, of which N for the update of \mathbf{x}_i , and the extra one for multiplier and slack variable update. Assume the CSI of transmit antennas is only accessed by the corresponding processors, while the data information is known by all processor units. The algorithm requires sharing CSI and interim results among processor units. We share $\{\mathbf{A}_i \mathbf{x}_i^t\}$ instead of sharing $\{\mathbf{A}_i\}$ and $\{\mathbf{x}_i^t\}$ separately, for reduced coordination overhead. Hence, the coordination overhead per iteration of the $N + 1$ processor units is $\mathcal{Q}N(N + 2)m$ bits, where \mathcal{Q} denotes the required bits for exchanging one real-valued scalar.

As for the low-coordination overhead decentralized counterpart, the processor dedicated to consensus variables, i.e., multiplier and slack variable, is eliminated. The multiplier is updated in each local processor unit, therefore the need for the exchange of consensus variables is removed. The exchanged information is the sole $\{\mathbf{A}_i \mathbf{x}_i^t\}$, thus the coordination overhead per iteration of the N processor units is $\mathcal{Q}N(N - 1)m$ bits.

VI. NUMERICAL RESULTS

This section evaluates and compares the performance of the proposed algorithms via Monte Carlo simulations. We assume each user has unit noise variance and equal instantaneous SINR threshold, i.e., $\sigma_k^2 = \sigma^2 = 1$, $\gamma_k = \gamma, \forall k$. QPSK modulation is employed throughout the simulations. A downlink massive MU-MISO system with 128 transmit antennas to serve 112 single-antenna users is considered unless otherwise specified. The transmit signal vector is partitioned into 64 blocks, namely $N = 64$, with 4 elements in each block.

In the sequel for clarity, we list the proposed algorithms as well as the benchmark schemes we have compared in our simulations:

- 1) ‘ZF’: The conventional ZF scheme with symbol-level power normalization. The corresponding precoded signal vector is given by

$$\tilde{\mathbf{x}}_{ZF} = \frac{1}{f_{ZF}} \tilde{\mathbf{H}}^H \left(\tilde{\mathbf{H}} \tilde{\mathbf{H}}^H \right)^{-1} \tilde{\mathbf{s}}, \quad (43)$$

where f_{ZF} is the symbol-level scaling factor. For the sake of comparison, the ZF transmit signal is normalized by the transmit power obtained by the IPM, thus we have

$$f_{ZF} = \frac{\left\| \tilde{\mathbf{H}}^H \left(\tilde{\mathbf{H}} \tilde{\mathbf{H}}^H \right)^{-1} \tilde{\mathbf{s}} \right\|}{\left\| \tilde{\mathbf{x}}_{IPM} \right\|}, \quad (44)$$

where $\tilde{\mathbf{x}}_{IPM}$ is the complex-valued precoded signal obtained by the IPM for PM-SLP.

- 2) ‘RZF’: The conventional RZF scheme with symbol-level power normalization. The corresponding precoded signal vector is given by

$$\tilde{\mathbf{x}}_{RZF} = \frac{1}{f_{RZF}} \tilde{\mathbf{H}}^H \left(\tilde{\mathbf{H}}\tilde{\mathbf{H}}^H + \varphi \mathbf{I} \right)^{-1} \tilde{\mathbf{s}}, \quad (45)$$

where f_{RZF} is the symbol-level scaling factor, φ is the regulation parameter, which is set to $\varphi = \sigma^2$. The RZF transmit signal is also normalized by the transmit power obtained by the IPM, thus we have

$$f_{RZF} = \frac{\left\| \tilde{\mathbf{H}}^H \left(\tilde{\mathbf{H}}\tilde{\mathbf{H}}^H + \varphi \mathbf{I} \right)^{-1} \tilde{\mathbf{s}} \right\|}{\|\tilde{\mathbf{x}}_{IPM}\|}. \quad (46)$$

- 3) ‘IPM’: The IPM for PM-SLP [42].
- 4) ‘EGPA’: The efficient gradient projection algorithm for PM-SLP [23].
- 5) ‘SCF’: The suboptimal closed-form solution for PM-SLP [24].
- 6) ‘ISCF’: The improved suboptimal closed-form solution for PM-SLP [25].
- 7) ‘PSLP-SA’: The proposed parallel and distributed approach for PM-SLP, with the standard proximal term and the adaptive parameter tuning strategy.
- 8) ‘PSLP-SC’: The proposed parallel and distributed approach for PM-SLP, with the standard proximal term and constant parameters.
- 9) ‘PSLP-LA’: The proposed parallel and distributed approach for PM-SLP, with the prox-linear proximal term and the adaptive parameter tuning strategy.
- 10) ‘PSLP-LC’: The proposed parallel and distributed approach for PM-SLP, with the prox-linear proximal term and constant parameters.

For PSLP-SA and PSLP-LA, the proximal parameters are initialized as $\tau_i = 0.1(N - 1)\rho$ and adaptively updated by the adaptive parameter tuning strategy in Section IV-C with $\delta_i = 2$. For PSLP-SC and PSLP-LC, we choose $\tau_i = 0.2\rho \left(\frac{N}{2-\beta} - 1 \right) \|\mathbf{A}_i\|^2$. The penalty parameter ρ is set to be 0.06; the damping parameter β is set to be 1.

A. Convergence Behavior

We first demonstrate the convergence behavior of the proposed approach. As the adaptive parameter tuning strategy needs precise information to evaluate whether the parameters need to be tuned, we adopt the more accurate low-coordination overhead decentralized formulation.

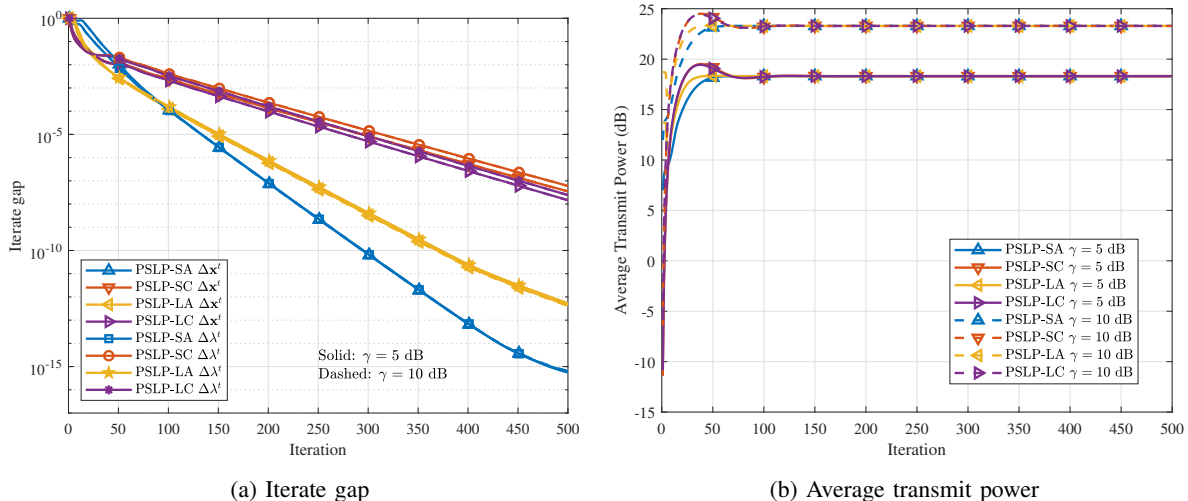


Fig. 3. Convergence behavior of the proposed approach in different SINR thresholds, QPSK, $N_t = 128$, $K = 112$, $N = 64$.

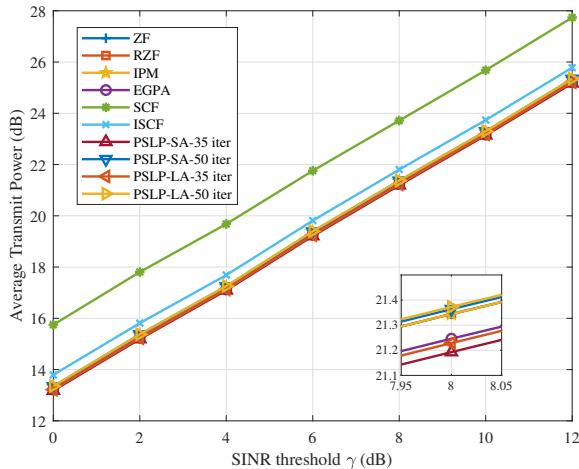
The iterate gap of the primal and dual variable is defined as $\Delta \mathbf{x}^t \triangleq \|\mathbf{x}^t - \mathbf{x}^{t-1}\| / \|\mathbf{x}^t\|$, $\Delta \lambda^t \triangleq \|\lambda^t - \lambda^{t-1}\| / \|\lambda^t\|$.

Fig. 3 illustrates the convergence behavior of the proposed algorithms in terms of iterate gap and average transmit power, which are both averaged over 2000 random channel realizations. The results in Fig. 3 show that the sequence generated by the proposed algorithm is convergent to a unique solution. The alternatively optimized primal and dual variable has an identical iterate gap. The algorithms with adaptive parameter tuning have a faster convergence rate compared to the algorithms with constant parameters. Meanwhile, using the same parameters, the inverse-free algorithms with the prox-linear proximal term have slightly slower convergence performance than the more complex algorithms with the standard proximal term.

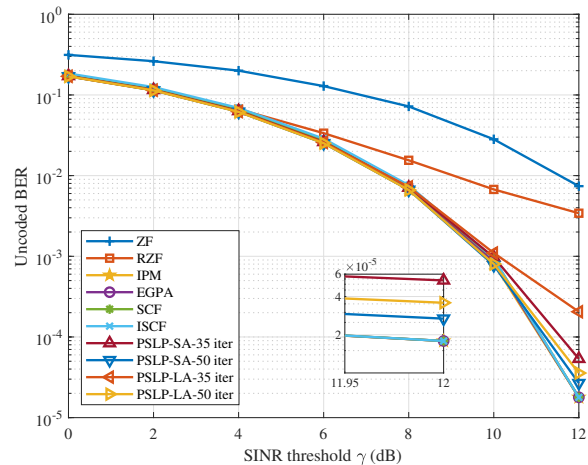
B. Transmit Power and Uncoded BER Performance

We compare the performance of the proposed approach and other schemes of interest in the view of transmit power and uncoded bit error rate (BER).

In Fig. 4a we depict the average transmit power with the SINR threshold for the same system setting. The transmit power of the proposed parallel SLP algorithms approach those of the IPM from low to high, since we initialize the transmit signal as a zero vector. Specifically, the early termination of the PSLP-SA and PSLP-LA algorithm at 35 iterates leads to a suboptimal solution of nearly 0.15 dB and 0.1 dB transmit power gap, respectively. When the number of iterations of PSLP reaches 50, all the proposed schemes have optimal transmit power.



(a) Average transmit power v.s. SINR threshold



(b) Uncoded BER v.s. SINR threshold

Fig. 4. Transmit power and uncoded BER performance of different schemes in different SINR thresholds, QPSK, $N_t = 128$, $K = 112$, $N = 64$.

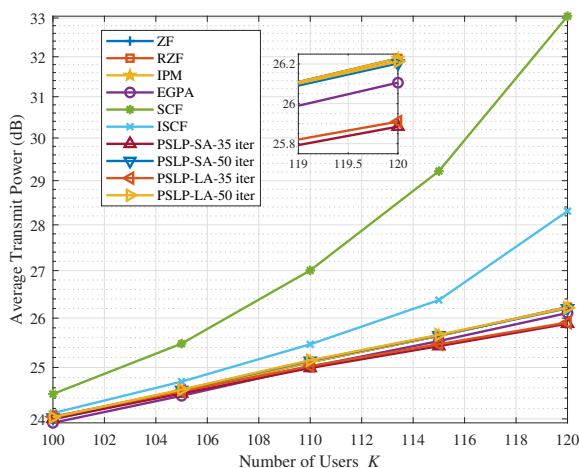


Fig. 5. Transmit power of different schemes in different number of users, QPSK, $N_t = 128$, $\gamma = 12$ dB, $N = 64$.

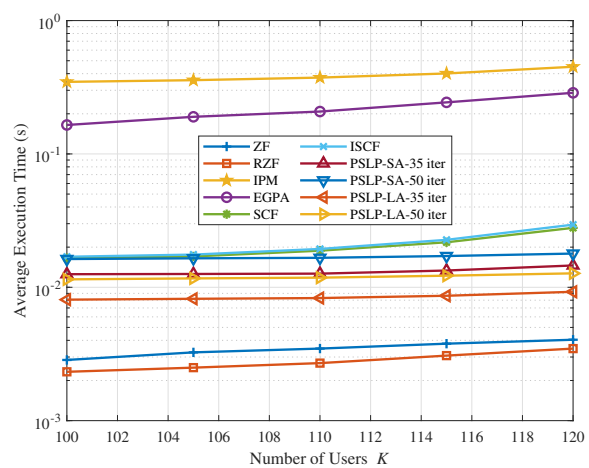


Fig. 6. Average execution time of different schemes in different number of users, QPSK, $N_t = 128$, $\gamma = 12$ dB, $N = 64$.

Fig. 4b shows the uncoded BER performance of the proposed parallel SLP approach compared to other schemes at various SINR thresholds for the same system setting. At the interference-limited medium-to-high SINR threshold region, the SLP schemes implemented by the proposed parallel approach, IPM, SCF, ISCF, and EGPA all achieve lower uncoded BER over the ZF and RZF scheme. The performance of the proposed approach increases stably with the number of iterations, providing a performance-complexity trade-off. With sufficient iterates, the uncoded BER performance of the proposed approach matches that of the IPM.

We then demonstrate the effectiveness of the proposed parallel SLP approach under a varying number of users in Fig. 5. The system has 128 transmit antennas, and the number of users varies from 100 to 120. The SINR threshold is 12 dB. The trend can be seen in Fig. 5 is that, unlike the ZF and RZF schemes as well as the SCF and ISCF schemes, whose transmit power performance severely deteriorates when the number of users increases, both the proposed PSLP-SA and PSLP-LA algorithms and EPGA show robustness among various load levels. Because the proposed parallel SLP approach and EPGA solve the PM-SLP problem successfully, regardless of the problem size. The performance of the early terminated parallel SLP approach at 35 iterations is degraded by the increasing system load, which validates that the larger problem size needs more iterations to converge.

C. Computational Complexity Comparison

Table I presents the average number of iterations and execution time per channel realization of the proposed schemes, where ‘MI’ and ‘SVD’ denote the plain implementation with matrix inversion in each iteration in Section IV-A and the SVD-based efficient implementation in Section IV-D, respectively. The stopping criterion is $\Delta \mathbf{x}^t < 10^{-3}$. We can observe that the proposed PIF-SLP algorithm with the adaptive tuning strategy (PSLP-LA) is the most efficient in reaching the given iterate gap.

TABLE I
AVERAGE NUMBER OF ITERATIONS AND EXECUTION TIME OVER 2000 RANDOM CHANNEL REALIZATIONS, QPSK,
 $N_t = 128, K = 112, N = 64$.

Schemes	$\gamma = 5$ dB		$\gamma = 10$ dB	
	Iterations	Time (s)	Iterations	Time (s)
PSLP-SA/MI	62.3255	0.0401	62.1500	0.0397
PSLP-SA/SVD	62.3255	0.0242	62.1500	0.0240
PSLP-LA	67.3320	0.0156	67.1945	0.0156
PSLP-SC/MI	133.0500	0.0742	133.3380	0.0741
PSLP-SC/SVD	133.0500	0.0426	133.3380	0.0425
PSLP-LC	126.1735	0.0293	126.2845	0.0292

Fig. 6 compares the average execution time required per channel realization of the concerned schemes. The system setting is the same as Fig. 5. We notice that implementing the parallel approach in physical parallel computing processors is beyond the range of this paper, thus the execution time for the proposed parallel SLP approach is the total time required for MATLAB simulation, which is an overestimate. It is observed that the proposed approach exhibits the lowest time complexity among other compared iterative schemes, i.e., IPM and EPGA. It also outperforms the closed-form solutions such as the SCF and the ISCF, excluding the heuristic ZF and RZF schemes.

VII. CONCLUSION

In this paper, parallel and decentralized processing for CI-based SLP is proposed for a massive MU-MISO downlink system based on ADMM. By reformulating the canonical PM-SLP optimization problem and introducing a slack variable vector, we transfer the original problem into separable equality constrained optimization, which is well-suited for the application of parallel processing. The augmented Lagrangian method is used to acquire an unconstrained problem formulation, which is further decomposed into several parallel subproblems via the PJ-ADMM framework. The sufficient condition for global convergence of the parallel CI-based SLP approach is derived, based on which a PIF-SLP algorithm is proposed to further alleviate the computational burden. Numerical results show the superiority of the proposed algorithms in terms of computational efficiency over state-of-the-art works, without compromising transmit power or BER performance.

APPENDIX A

PROOF OF THEOREM 1

Proof: The first-order optimal condition for \mathbf{x}_i is given by

$$\mathbf{A}_i^T \left(\boldsymbol{\lambda}^t - \rho \sum_{j=1}^N \mathbf{A}_j \mathbf{x}_j^t + \mathbf{b} + \mathbf{c}^{t+1} \right) + \mathbf{P}_i (\mathbf{x}_i^t - \mathbf{x}_i^{t+1}) + \rho \mathbf{A}_i^T \mathbf{A}_i (\mathbf{x}_i^t - \mathbf{x}_i^{t+1}) \in \partial f_i (\mathbf{x}_i^{t+1}), \quad (47)$$

where $f_i (\mathbf{x}_i) \triangleq \|\mathbf{x}_i\|^2$.

Denoting $\hat{\boldsymbol{\lambda}} = \boldsymbol{\lambda}^t - \rho (\mathbf{A} \mathbf{x}^{t+1} - \mathbf{b} - \mathbf{c}^{t+1})$, then the first-order optimal condition (47) turns to

$$\mathbf{A}_i^T \left[\hat{\boldsymbol{\lambda}} - \rho \sum_{j=1}^N \mathbf{A}_j (\mathbf{x}_j^t - \mathbf{x}_j^{t+1}) \right] + (\mathbf{P}_i + \rho \mathbf{A}_i^T \mathbf{A}_i) (\mathbf{x}_i^t - \mathbf{x}_i^{t+1}) \in \partial f_i (\mathbf{x}_i^{t+1}). \quad (48)$$

Assum there exist a saddle point $\mathbf{u}^* = (\mathbf{x}_1^*, \mathbf{x}_2^* \cdots, \mathbf{x}_N^*, \boldsymbol{\lambda}^*)$ for PM-SLP. From the convexity of $f_i(\mathbf{x}_i)$, we have

$$(\partial f_i(\mathbf{x}_i^{t+1}) - \partial f_i(\mathbf{x}_i^*))^T (\mathbf{x}_i^{t+1} - \mathbf{x}_i^*) \geq 0. \quad (49)$$

The stationarity condition of KKT conditions is given by

$$\mathbf{A}_i^T \boldsymbol{\lambda}^* \in \partial f_i(\mathbf{x}_i^*). \quad (50)$$

Thus (49) can be written as

$$\begin{aligned} & \left[\mathbf{A}_i^T \left(\hat{\boldsymbol{\lambda}} - \boldsymbol{\lambda}^* - \rho \sum_{j=1}^N \mathbf{A}_j (\mathbf{x}_j^t - \mathbf{x}_j^{t+1}) \right) \right]^T (\mathbf{x}_i^{t+1} - \mathbf{x}_i^*) \\ & + (\mathbf{x}_i^{t+1} - \mathbf{x}_i^*)^T (\mathbf{P}_i + \rho \mathbf{A}_i^T \mathbf{A}_i) (\mathbf{x}_i^t - \mathbf{x}_i^{t+1}) \geq 0. \end{aligned} \quad (51)$$

Summing the above inequality over all i , we obtain

$$\begin{aligned} & (\hat{\boldsymbol{\lambda}} - \boldsymbol{\lambda}^*)^T \mathbf{A} (\mathbf{x}^{t+1} - \mathbf{x}^*) + \sum_{i=1}^N (\mathbf{x}_i^{t+1} - \mathbf{x}_i^*)^T (\mathbf{P}_i + \rho \mathbf{A}_i^T \mathbf{A}_i) (\mathbf{x}_i^t - \mathbf{x}_i^{t+1}) \\ & \geq \rho (\mathbf{x}^t - \mathbf{x}^{t+1})^T \mathbf{A}^T \mathbf{A} (\mathbf{x}^{t+1} - \mathbf{x}^*). \end{aligned} \quad (52)$$

Note that

$$\mathbf{A} (\mathbf{x}^{t+1} - \mathbf{x}^*) = \frac{1}{\beta \rho} (\boldsymbol{\lambda}^t - \boldsymbol{\lambda}^{t+1}), \quad (53)$$

$$\hat{\boldsymbol{\lambda}} - \boldsymbol{\lambda}^* = \left(\hat{\boldsymbol{\lambda}} - \boldsymbol{\lambda}^{t+1} \right) + (\boldsymbol{\lambda}^{t+1} - \boldsymbol{\lambda}^*) = \frac{\beta - 1}{\beta} (\boldsymbol{\lambda}^t - \boldsymbol{\lambda}^{t+1}) + (\boldsymbol{\lambda}^{t+1} - \boldsymbol{\lambda}^*). \quad (54)$$

Substituting (53) and (54) into (52), we obtain

$$\begin{aligned} & (\boldsymbol{\lambda}^{t+1} - \boldsymbol{\lambda}^*)^T \frac{1}{\beta \rho} (\boldsymbol{\lambda}^t - \boldsymbol{\lambda}^{t+1}) + \sum_{i=1}^N (\mathbf{x}_i^{t+1} - \mathbf{x}_i^*)^T (\mathbf{P}_i + \rho \mathbf{A}_i^T \mathbf{A}_i) (\mathbf{x}_i^t - \mathbf{x}_i^{t+1}) \\ & \geq \frac{1 - \beta}{\beta^2 \rho} \|\boldsymbol{\lambda}^t - \boldsymbol{\lambda}^{t+1}\|^2 + \frac{1}{\beta} (\boldsymbol{\lambda}^t - \boldsymbol{\lambda}^{t+1})^T \mathbf{A} (\mathbf{x}^t - \mathbf{x}^{t+1}). \end{aligned} \quad (55)$$

For notation simplicity, denoting $\mathbf{G}_x \triangleq \begin{bmatrix} \mathbf{P}_1 + \rho \mathbf{A}_1^T \mathbf{A}_1 & & \\ & \ddots & \\ & & \mathbf{P}_N + \rho \mathbf{A}_N^T \mathbf{A}_N \end{bmatrix}$, $\mathbf{G} \triangleq \begin{bmatrix} \mathbf{G}_x & \\ & \frac{1}{\beta \rho} \mathbf{I} \end{bmatrix}$,

$$\mathbf{Q} \triangleq \begin{bmatrix} \mathbf{P}_1 + \rho \mathbf{A}_1^T \mathbf{A}_1 & & & \frac{1}{\beta} \mathbf{A}_1^T \\ & \ddots & & \vdots \\ & & \mathbf{P}_N + \rho \mathbf{A}_N^T \mathbf{A}_N & \frac{1}{\beta} \mathbf{A}_N^T \\ \frac{1}{\beta} \mathbf{A}_1^T & \cdots & \frac{1}{\beta} \mathbf{A}_N^T & \frac{2 - \beta}{\rho \beta^2} \mathbf{I} \end{bmatrix}. \text{ Essentially, from (55) we have}$$

$$(\mathbf{u}^t - \mathbf{u}^{t+1})^T \mathbf{G} (\mathbf{u}^{t+1} - \mathbf{u}^*) \geq \frac{1 - \beta}{\beta^2 \rho} \|\boldsymbol{\lambda}^t - \boldsymbol{\lambda}^{t+1}\|^2 + \frac{1}{\beta} (\boldsymbol{\lambda}^t - \boldsymbol{\lambda}^{t+1})^T \mathbf{A} (\mathbf{x}^t - \mathbf{x}^{t+1}). \quad (56)$$

Thus we have the relationship

$$\|\mathbf{u}^t - \mathbf{u}^*\|_{\mathbf{G}}^2 - \|\mathbf{u}^{t+1} - \mathbf{u}^*\|_{\mathbf{G}}^2 = 2(\mathbf{u}^t - \mathbf{u}^{t+1})^T \mathbf{G} (\mathbf{u}^{t+1} - \mathbf{u}^*) + \|\mathbf{u}^t - \mathbf{u}^{t+1}\|_{\mathbf{G}}^2 \quad (57a)$$

$$\begin{aligned} &\geq \frac{2-2\beta}{\beta^2\rho} \|\boldsymbol{\lambda}^t - \boldsymbol{\lambda}^{t+1}\|^2 + \frac{2}{\beta} (\boldsymbol{\lambda}^t - \boldsymbol{\lambda}^{t+1})^T \mathbf{A} (\mathbf{x}^t - \mathbf{x}^{t+1}) \\ &\quad + \|\mathbf{u}^t - \mathbf{u}^{t+1}\|_{\mathbf{G}}^2 \end{aligned} \quad (57b)$$

$$\begin{aligned} &= \frac{2-2\beta}{\beta^2\rho} \|\boldsymbol{\lambda}^t - \boldsymbol{\lambda}^{t+1}\|^2 + \frac{2}{\beta} (\boldsymbol{\lambda}^t - \boldsymbol{\lambda}^{t+1})^T \mathbf{A} (\mathbf{x}^t - \mathbf{x}^{t+1}) \\ &\quad + \|\mathbf{x}^t - \mathbf{x}^{t+1}\|_{\mathbf{G}_x}^2 + \frac{1}{\beta\rho} \|\boldsymbol{\lambda}^t - \boldsymbol{\lambda}^{t+1}\|^2 \end{aligned} \quad (57c)$$

$$\begin{aligned} &= \frac{2-\beta}{\beta^2\rho} \|\boldsymbol{\lambda}^t - \boldsymbol{\lambda}^{t+1}\|^2 + \frac{2}{\beta} (\boldsymbol{\lambda}^t - \boldsymbol{\lambda}^{t+1})^T \mathbf{A} (\mathbf{x}^t - \mathbf{x}^{t+1}) \\ &\quad + \|\mathbf{x}^t - \mathbf{x}^{t+1}\|_{\mathbf{G}_x}^2 \end{aligned} \quad (57d)$$

$$= \|\mathbf{u}^t - \mathbf{u}^{t+1}\|_{\mathbf{Q}}^2. \quad (57e)$$

To prove the convergence of the PJ-ADMM for PM-SLP is reduced to ensure that \mathbf{Q} is positive semi-definite. For any $\mathbf{u} \in \mathbb{R}^{2N_t+2K}$, we have

$$\|\mathbf{u}\|_{\mathbf{Q}}^2 = \|\mathbf{x}\|_{\mathbf{G}_x}^2 + \frac{2-\beta}{\beta^2\rho} \|\boldsymbol{\lambda}\|^2 + \frac{2}{\beta} \boldsymbol{\lambda}^T \mathbf{A} \mathbf{x}. \quad (58)$$

Using the basic inequality

$$\frac{2}{\beta} \boldsymbol{\lambda}^T \mathbf{A} \mathbf{x} = \sum_{i=1}^N \frac{2}{\beta} \boldsymbol{\lambda}^T \mathbf{A}_i \mathbf{x}_i \geq - \sum_{i=1}^N \left(\frac{\epsilon_i}{\rho\beta^2} \|\boldsymbol{\lambda}\|^2 + \frac{\rho}{\epsilon_i} \|\mathbf{A}_i \mathbf{x}_i\|^2 \right), \quad (59)$$

for any $\epsilon_i > 0$, we have

$$\|\mathbf{u}\|_{\mathbf{Q}}^2 \geq \sum_{i=1}^N \|\mathbf{x}_i\|_{\mathbf{P}_i + \rho \mathbf{A}_i^T \mathbf{A}_i - \frac{\rho}{\epsilon_i} \mathbf{A}_i^T \mathbf{A}_i}^2 + \frac{2-\beta - \sum_{i=1}^N \epsilon_i}{\beta^2\rho} \|\boldsymbol{\lambda}\|^2. \quad (60)$$

Therefore, \mathbf{Q} is positive semi-definite if

$$\mathbf{P}_i \succeq \rho \left(\frac{1}{\epsilon_i} - 1 \right) \mathbf{A}_i^T \mathbf{A}_i, \forall i, \sum_{i=1}^N \epsilon_i \leq 2 - \beta, \quad (28)$$

where $\epsilon_i > 0$.

If the sufficient condition is satisfied, then the error metric $\|\mathbf{u}^t - \mathbf{u}^*\|_{\mathbf{G}}^2$ is monotonically non-decreasing, and the sequence $\{\mathbf{u}^t\}$ generated by the PJ-ADMM is contractive. The global convergence of the algorithm follows immediately from the analysis of the contraction method [43]. ■

REFERENCES

- [1] F. Rusek, D. Persson, B. K. Lau, E. G. Larsson, T. L. Marzetta, O. Edfors, and F. Tufvesson, "Scaling up mimo: Opportunities and challenges with very large arrays," *IEEE Signal Process. Mag.*, vol. 30, no. 1, pp. 40–60, 2013.
- [2] L. Sanguinetti, E. Björnson, and J. Hoydis, "Toward massive mimo 2.0: Understanding spatial correlation, interference suppression, and pilot contamination," *IEEE Trans. Commun.*, vol. 68, no. 1, pp. 232–257, 2020.
- [3] L. Liu and W. Yu, "Massive connectivity with massive mimo—part ii: Achievable rate characterization," *IEEE Trans. Signal Process.*, vol. 66, no. 11, pp. 2947–2959, 2018.
- [4] T. L. Marzetta, "Noncooperative cellular wireless with unlimited numbers of base station antennas," *IEEE Trans. Wireless Commun.*, vol. 9, no. 11, pp. 3590–3600, 2010.
- [5] H. Q. Ngo, E. G. Larsson, and T. L. Marzetta, "Energy and spectral efficiency of very large multiuser mimo systems," *IEEE Trans. Commun.*, vol. 61, no. 4, pp. 1436–1449, 2013.
- [6] L. Zheng and D. Tse, "Diversity and multiplexing: a fundamental tradeoff in multiple-antenna channels," *IEEE Trans. Inf. Theory*, vol. 49, no. 5, pp. 1073–1096, 2003.
- [7] T. Lo, "Maximum ratio transmission," *IEEE Trans. Commun.*, vol. 47, no. 10, pp. 1458–1461, 1999.
- [8] G. Caire and S. Shamai, "On the achievable throughput of a multiantenna gaussian broadcast channel," *IEEE Trans. Inf. Theory*, vol. 49, no. 7, pp. 1691–1706, 2003.
- [9] C. Peel, B. Hochwald, and A. Swindlehurst, "A vector-perturbation technique for near-capacity multiantenna multiuser communication—part i: channel inversion and regularization," *IEEE Trans. Commun.*, vol. 53, no. 1, pp. 195–202, 2005.
- [10] H. Yang and T. L. Marzetta, "Performance of conjugate and zero-forcing beamforming in large-scale antenna systems," *IEEE J. Sel. Areas Commun.*, vol. 31, no. 2, pp. 172–179, 2013.
- [11] M. Costa, "Writing on dirty paper (corresp.)," *IEEE Trans. Inf. Theory*, vol. 29, no. 3, pp. 439–441, 1983.
- [12] C. Windpassinger, R. Fischer, T. Vencel, and J. Huber, "Precoding in multiantenna and multiuser communications," *IEEE Trans. Wireless Commun.*, vol. 3, no. 4, pp. 1305–1316, 2004.
- [13] B. Hochwald, C. Peel, and A. Swindlehurst, "A vector-perturbation technique for near-capacity multiantenna multiuser communication—part ii: perturbation," *IEEE Trans. Commun.*, vol. 53, no. 3, pp. 537–544, 2005.
- [14] E. Visotsky and U. Madhow, "Optimum beamforming using transmit antenna arrays," in *1999 IEEE 49th Vehicular Technology Conference (Cat. No.99CH36363)*, vol. 1, 1999, pp. 851–856 vol.1.
- [15] A. Wiesel, Y. Eldar, and S. Shamai, "Linear precoding via conic optimization for fixed mimo receivers," *IEEE Trans. Signal Process.*, vol. 54, no. 1, pp. 161–176, 2006.
- [16] M. Bengtsson and B. Ottersten, "Optimal downlink beamforming using semidefinite optimization," in *37th Annual Allerton Conference on Communication, Control, and Computing*, 1999, pp. 987–996.
- [17] D. Palomar, J. Cioffi, and M. Lagunas, "Joint tx-rx beamforming design for multicarrier mimo channels: a unified framework for convex optimization," *IEEE Trans. Signal Process.*, vol. 51, no. 9, pp. 2381–2401, 2003.
- [18] M. Schubert and H. Boche, "Solution of the multiuser downlink beamforming problem with individual sinr constraints," *IEEE Trans. Veh. Technol.*, vol. 53, no. 1, pp. 18–28, 2004.
- [19] E. Karipidis, N. D. Sidiropoulos, and Z.-Q. Luo, "Quality of service and max-min fair transmit beamforming to multiple cochannel multicast groups," *IEEE Trans. Signal Process.*, vol. 56, no. 3, pp. 1268–1279, 2008.
- [20] C. Masouros and E. Alsusa, "A novel transmitter-based selective-precoding technique for ds/cdma systems," vol. 14, no. 9, 2007, pp. 637–640.
- [21] ———, "Dynamic linear precoding for the exploitation of known interference in mimo broadcast systems," *IEEE Trans. Wireless Commun.*, vol. 8, no. 3, pp. 1396–1404, 2009.

- [22] C. Masouros, “Correlation rotation linear precoding for mimo broadcast communications,” *IEEE Trans. Signal Process.*, vol. 59, no. 1, pp. 252–262, 2011.
- [23] C. Masouros and G. Zheng, “Exploiting known interference as green signal power for downlink beamforming optimization,” *IEEE Trans. Signal Process.*, vol. 63, no. 14, pp. 3628–3640, 2015.
- [24] A. Haqiqatnejad, F. Kayhan, and B. Ottersten, “Power minimizer symbol-level precoding: A closed-form suboptimal solution,” *IEEE Signal Process. Lett.*, vol. 25, no. 11, pp. 1730–1734, 2018.
- [25] —, “An approximate solution for symbol-level multiuser precoding using support recovery,” in *2019 IEEE 20th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, 2019, pp. 1–5.
- [26] A. Li and C. Masouros, “Interference exploitation precoding made practical: Optimal closed-form solutions for psk modulations,” *IEEE Trans. Wireless Commun.*, vol. 17, no. 11, pp. 7661–7676, 2018.
- [27] A. Li, C. Masouros, B. Vucetic, Y. Li, and A. L. Swindlehurst, “Interference exploitation precoding for multi-level modulations: Closed-form solutions,” *IEEE Trans. Commun.*, vol. 69, no. 1, pp. 291–308, 2021.
- [28] Z. Lei, X. Liao, Z. Gao, and A. Li, “Ci-nn: A model-driven deep learning-based constructive interference precoding scheme,” *IEEE Commun. Lett.*, vol. 25, no. 6, pp. 1896–1900, 2021.
- [29] A. Mohammad, C. Masouros, and Y. Andreopoulos, “An unsupervised deep unfolding framework for robust symbol level precoding,” *arXiv preprint arXiv:2111.08129*, 2021.
- [30] A. Li, D. Spano, J. Krivochiza, S. Domouchtsidis, C. G. Tsinos, C. Masouros, S. Chatzinotas, Y. Li, B. Vucetic, and B. Ottersten, “A tutorial on interference exploitation via symbol-level precoding: Overview, state-of-the-art and future directions,” *IEEE Communications Surveys & Tutorials*, vol. 22, no. 2, pp. 796–839, 2020.
- [31] A. Haqiqatnejad, F. Kayhan, and B. Ottersten, “Constructive interference for generic constellations,” *IEEE Signal Process. Lett.*, vol. 25, no. 4, pp. 586–590, 2018.
- [32] C. Masouros, M. Sellathurai, and T. Ratnarajah, “Vector perturbation based on symbol scaling for limited feedback mimo downlinks,” *IEEE Trans. Signal Process.*, vol. 62, no. 3, pp. 562–571, 2014.
- [33] D. P. Bertsekas, *Constrained Optimization and Lagrange Multiplier Methods*. Academic Press, 2014.
- [34] C. Chen, B. He, Y. Ye, and X. Yuan, “The direct extension of admm for multi-block convex minimization problems is not necessarily convergent,” *Math. Program.*, vol. 155, no. 1, pp. 57–79, 2016.
- [35] B. He, L. Hou, and X. Yuan, “On full jacobian decomposition of the augmented lagrangian method for separable convex programming,” *SIAM J. Optim.*, vol. 25, no. 4, pp. 2274–2312, 2015.
- [36] B. He, H.-K. Xu, and X. Yuan, “On the proximal jacobian decomposition of alm for multiple-block separable convex minimization problems and its relationship to admm,” *J. Sci. Comput.*, vol. 66, no. 3, pp. 1204–1217, 2016.
- [37] W. Deng, M.-J. Lai, Z. Peng, and W. Yin, “Parallel multi-block admm with $\mathcal{O}(1/k)$ convergence,” *J. Sci. Comput.*, vol. 71, no. 2, pp. 712–736, 2017.
- [38] D. Gabay and B. Mercier, “A dual algorithm for the solution of nonlinear variational problems via finite element approximation,” *Computers & Mathematics with Applications*, vol. 2, no. 1, pp. 17–40, 1976.
- [39] L. Chu, F. Wen, L. Li, and R. Qiu, “Efficient nonlinear precoding for massive mimo downlink systems with 1-bit dacs,” *IEEE Trans. Wireless Commun.*, vol. 18, no. 9, pp. 4213–4224, 2019.
- [40] D. Bertsekas and J. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*. Athena Scientific, 2015.
- [41] D. Bertsekas, *Nonlinear Programming*. Athena Scientific, 2016, vol. 4.
- [42] M. Grant and S. Boyd, “CVX: Matlab software for disciplined convex programming, version 2.1,” <http://cvxr.com/cvx>, Mar. 2014.
- [43] B. He, “A class of projection and contraction methods for monotone variational inequalities,” *Appl. Math. Optim.*, vol. 35, no. 1, pp. 69–76, 1997.