Noname manuscript No.
(will be inserted by the editor)

A Two-Level Distributed Algorithm for Nonconvex Constrained Optimization

Kaizhao Sun · X. Andy Sun

Received: date / Accepted: date

Abstract This paper aims to develop distributed algorithms for nonconvex optimization problems with complicated constraints associated with a network. The network can be a physical one, such as an electric power network, where the constraints are nonlinear power flow equations, or an abstract one that represents constraint couplings between decision variables of different agents. Despite the recent development of distributed algorithms for nonconvex programs, highly complicated constraints still pose a significant challenge in theory and practice. We first identify some difficulties with the existing algorithms based on the alternating direction method of multipliers (ADMM) for dealing with such problems. We then propose a reformulation that enables us to design a two-level algorithm, which embeds a specially structured three-block ADMM at the inner level in an augmented Lagrangian method (ALM) framework. Furthermore, we prove the global and local convergence as well as iteration complexity of this new scheme for general nonconvex constrained programs, and show that our analysis can be extended to handle more complicated multi-block inner-level problems. Finally, we demonstrate with computation that the new scheme provides convergent and parallelizable algorithms for various nonconvex applications, and is able to complement the performance of the state-of-the-art distributed algorithms in practice by achieving either faster convergence in optimality gap or in feasibility or both.

Keywords Distributed Optimization · Augmented Lagrangian Method · Alternating Direction Method of Multipliers

Kaizhao Sun
H. Milton Stewart School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA
E-mail: ksun46@gatech.edu

X. Andy Sun
Sloan School of Management, Massachusetts Institute of Technology, Cambridge, MA
E-mail: sunx@mit.edu

Mathematics Subject Classification (2010) 90C06 · 90C26 · 90C30 · 90C35

1 Introduction

This paper develops a new two-level distributed algorithm with global and local convergence guarantees for solving general smooth and nonsmooth constrained nonconvex optimization problems. We will start with a general constrained optimization model that is motivated by nonlinear network flow problems, and then explore reformulations for distributed computation. This process of reformulation leads us to observe two structural properties that a distributed reformulation should possess, which in fact pose some challenges to existing distributed algorithms in terms of convergence and practical performance. This observation inspired us to develop the two-level distributed algorithm. We will summarize our contributions in the end of this section.

1.1 Constrained Nonconvex Optimization over a Network

Consider a connected, undirected graph[1] $G(\mathcal{V}, \mathcal{E})$ with a set of nodes $\mathcal{V}$ and a set of edges $\mathcal{E}$. A centralized constrained optimization problem on G is given as

$$\begin{aligned} \min \quad & \sum_{i\in\mathcal{V}} f_i(x_i) && \text{(1a)}\\ \text{s.t.} \quad & h_i(x_i, \{x_j\}_{j\in\delta(i)}) = 0, \quad \forall i \in \mathcal{V}, && \text{(1b)}\\ & g_i(x_i, \{x_j\}_{j\in\delta(i)}) \le 0, \quad \forall i \in \mathcal{V}, && \text{(1c)}\\ & x_i \in \mathcal{X}_i, \quad \forall i \in \mathcal{V}, && \text{(1d)} \end{aligned}$$

where each node $i \in \mathcal{V}$ of the graph G is associated with a decision variable x_i and a cost function $f_i(x_i)$ as in (1a). Variable x_i and variables x_j of i's adjacent nodes $j \in \delta(i)$ are coupled through constraints (1b)-(1c), and $\mathcal{X}_i$ in (1d) represents some constraints only for x_i. The functions f_i, h_i, g_i, and the set $\mathcal{X}_i$ may be nonconvex.

Any constrained optimization problem can be reformulated as (1) after proper transformation. An especially interesting motivation for us is the nonlinear network flow problems. In this case, the graph G represents a physical network such as an electric power network, a natural gas pipeline network, or a water transport network, where the variables x_i in (1) are nodal potentials such as electric voltages, gas pressures, or water pressures, and the constraints h_i and g_i are usually nonconvex functions that describe the physical relations between nodal potentials and flows on the edges, flow balance at nodes, and flow capacity constraints. Notice that a node i in the graph can also represent a sub-network of the entire physical network, and then the constraints could

[1] In this paper, we use "networks" and "graphs" interchangeably.

involve variables in adjacent sub-networks. There has been much recent interest in solving nonlinear network flow problems, e.g. applications in the optimal power flow problem in electric power network [34], the natural gas nomination problem [52], and the water network scheduling problem [14].

In many situations, it is desirable to solve problem (1) in a distributed manner, where each node i represents an individual agent that solves a localized problem, while agents coordinate with their neighbors to solve the overall problem. Each agent need to handle its own set of local constraints h_i, g_i, and $\mathcal{X}_i$. For example, agents may be geographically dispersed with local constraints representing the physics of the subsystems, which cannot be controlled by other agents; or agents may have private data in their constraints, which cannot be shared with other agents; or the sheer amount of data needed to describe constraints or objective could be too large to be stored or transmitted in distributed computation between agents. These practical considerations pose restrictions that each agent in a distributed algorithm has to deal with a set of complicated, potentially nonconvex, constraints.

1.2 Necessary Structures of Distributed Formulations

In order to do distributed computation, the centralized formulation (1) first needs to be transformed into a formulation to which a distributed algorithm could be applied. We call such a formulation a *distributed formulation*, whose form may depend on specific distributed algorithms as well as on the structure of the distributed computation, e.g. which variables and constraints are controlled by which agents and in what order computation and communication can be carried out. Despite the great variety of distributed formulations, we want to identify some desirable and necessary features for a distributed formulation.

One desirable feature is the capability of *parallel decomposition* so that all agents can solve their local problems in parallel, rather than in sequence. To realize this, each agent needs a local copy of its neighboring agents' variables. For problem (1), we may introduce a local copy x_j^i of the original variable x_j and a global copy $\bar{x}_j$, and enforce consensus as

$$x_j = \bar{x}_j,\ x_j^i = \bar{x}_j, \quad \forall j \in \mathcal{V},\ i \in \delta(j). \tag{2}$$

Using this duplication scheme, a distributed formulation of (1) can be written as

$$\min_{x,\bar{x}} \quad f(x) = \sum_{i\in\mathcal{V}} f_i(x^i) \tag{3a}$$

$$\text{s.t.} \quad Ax + B\bar{x} = 0, \tag{3b}$$

$$x^i \in \mathcal{X}_i, \quad \forall i \in \mathcal{V}, \quad \bar{x} \in \bar{\mathcal{X}}. \tag{3c}$$

In problem (3), the optimization variables are $x = [\{x^i\}_{i\in\mathcal{V}}] \in \mathbb{R}^{n_1}$ and $\bar{x} = [\{\bar{x}_j\}_{j\in\mathcal{V}}] \in \mathbb{R}^{n_2}$. Each subvector $x^i = [x_i, \{x_j^i\}_{j\in\delta(i)}] \in \mathbb{R}^{n_{1i}}$ of x denotes

all the local variables controlled by agent i including the original variable x_i and the local copies x_j^i; each subvector $\bar{x}_j$ of $\bar{x}$ denotes a global copy of x_j. The set $\mathcal{X}_i \subseteq \mathbb{R}^{n_{1i}}$ is defined as $\mathcal{X}_i := \{v : \ h_i(v) = 0, \ g_i(v) \leq 0\}$, so the original constraints (1b)-(1c) are decoupled into each agent's local constraints $\mathcal{X}_i$, which also absorb the constraints (1d). Additionally, the global copy $\bar{x}$ is constrained in some simple convex set $\bar{\mathcal{X}} \subseteq \mathbb{R}^{n_2}$. The only coupling among agents are (3b), which formulate the consensus constraint (2) with $A \in \mathbb{R}^{m \times n_1}$ and $B \in \mathbb{R}^{m \times n_2}$. An alternating optimization scheme is then natural, as all the agents can solve their subproblems over x^i's in parallel once $\bar{x}$ is fixed; and once x^i's are updated and fixed, the subproblems over $\bar{x}$ can also be solved in parallel.

In fact, for any constrained optimization problem, not necessarily a network flow type problem, if distributed computation is considered, variables of the centralized problem need to be grouped into variables x^i in a distributed formulation for agents i according to the decision structure, and duplicate variables $\bar{x}$ need to be introduced to decouple the constraints from agents. In this way, problem (3) provides a general formulation for distributed computation of constrained optimization problems. Conversely, due to the necessity of duplicating variables, any distributed formulation of a constrained program *necessarily* shares some key structures of (3). In particular, problem (3) has two simple but crucial properties. Namely,

- Property 1: As the matrices A and B are defined in (2), the image of A strictly contains the image of B, i.e. $\mathrm{Im}(A) \supsetneq \mathrm{Im}(B)$.
- Property 2: Each agent i may face local nonconvex constraints $\mathcal{X}_i$.

Property 1 follows from the fact that, for any given value of $\bar{x}_j$ in (2), there is always a feasible solution (x_j, x_j^i) that satisfies the equalities in (2), but if $x_j \neq x_j^i$, then there does not exist an $\bar{x}_j$ that satisfies both equalities in (2). Property 2 follows from our desire to decompose the computation for different agents.

In this paper, we will show that the above two properties of distributed constrained optimization pose a significant challenge to the theory and practice of existing distributed optimization algorithms. In particular, existing distributed algorithms based on the alternating direction method of multipliers (ADMM) may fail to converge for the general nonconvex constrained problem (3) without further reformulation or relaxation. Before proceeding, we summarize our contributions.

1.3 Summary of Contributions

The contributions of the paper can be summarized below.

Firstly, we propose a new reformulation and a two-level distributed algorithm for solving nonconvex constrained optimization problem (1)-(2), which embeds a specially structured three-block ADMM at the inner level in an augmented Lagrangian method (ALM) framework. The proposed algorithm maintains the flexibility of ADMM in achieving distributed computation.

Secondly, we prove global and local convergence as well as iteration complexity results for the proposed two-level algorithm, and illustrate that the underlying algorithmic framework can be extended to more complicated nonconvex multi-block problems. For the convergence of ADMM, we allow each nonconvex subproblem to be solved to a stationary point with certain improvement in the objective function compared to the previous iterate, which mildly relaxes the global optimality of nonconvex subproblems commonly assumed in the ADMM literature. Our convergence analysis builds on the classical and recent works on ADMM and ALM, and our results are derived by relating these two methods in an analytical way.

Thirdly, we provide extensive computational tests of our two-level algorithm on nonconvex network flow problems, parallel minimization of nonconvex functions over compact manifolds, and a robust tensor PCA problem from machine learning. Numerical results demonstrate the advantages of the proposed algorithm over existing ones, including randomized updates, modified ADMM, and centralized solver, either in the convergence speed to close optimality gap, or to close feasibility gap, or both. Moreover, our test result on the multi-block robust tensor PCA problem suggests that the proposed two-level algorithm not only ensures convergence for a wider range of applications where ADMM may fail, but also tends to accelerate ADMM on problems where convergence of ADMM is already guaranteed.

1.4 Notation

Throughout this paper, we use $\mathbb{Z}_+$ (resp. $\mathbb{Z}_{++}$) to denote the set of nonnegative (resp. positive) integers, and $\mathbb{R}^n$ to denote the n-dimensional real Euclidean space. For $x, y \in \mathbb{R}^n$, the inner product is denoted by $x^\top y$ or $\langle x, y\rangle$; the Euclidean norm is denoted by $\|x\| = \sqrt{\langle x, x\rangle}$. A vector x may consist of J subvectors $x_j \in \mathbb{R}^{n_j}$ with $\sum_{j=1}^J n_j = n$; in this case, we will write $x = [\{x_j\}_{j\in[J]}]$, where $[J] = \{1, \cdots, J\}$. Occasionally, we use x_i to denote the i-th component of x if there is no confusion to do so. For a matrix A, denote its largest singular value by $\|A\|$ and image space by $\mathrm{Im}(A)$. We use $B_r(x)$ to denote the Euclidean ball centered at x with radius $r > 0$. For a closed set $C \subset \mathbb{R}^n$, the interior of C is denoted by $\mathbf{Int}\ C$, the projection operator onto C is denoted by $\mathrm{Proj}_C(x)$, and the indicator function of C is denoted by $\mathbb{I}_C(x)$, which takes value 0 if $x \in C$ and $+\infty$ otherwise.

The rest of this paper is organized as follows. In Section 2, we review the literature and summarize two conditions that are crucial to the convergence of ADMM, which are essentially contradicting to Properties 1 and 2. In Section 3, we propose our new reformulation and a two-level algorithm for solving problem (3) in a distributed way. In Section 4, we provide the global convergence as well as the iteration complexity result, and show our scheme can be applied to more complicated multi-block problems. Then in Section 5, we show the local convergence result under standard second-order assumptions. Finally, we present computational results in Section 6 and conclude in Section 7.

2 Related Literature

In this section, we review the literature on ADMM and other distributed algorithms, and identify some limitations of the standard ADMM approach in solving problem (3).

2.1 Earlier Works and ADMM for Convex Problems

ALM and the method of multipliers (MoM) were proposed in the late 1960s by Hestenes [28] and Powell [53]. ALM enjoys more robust convergence properties than dual decomposition [4,55], and convergence for partial elimination of constraints has been studied [5]. ADMM was proposed by Glowinski and Marrocco [19] and Gabay and Mercier [18] in the mid-1970s, and has deep roots in maximal monotone operator theory and numerical algorithms for solving partial differential equations [15,12,51]. ADMM solves the subproblems in ALM by alternately optimizing through blocks of variables and in this way achieves distributed computation. The convergence of ADMM with two block variables is proved for convex optimization problems [19,18,17,15] and the $\mathcal{O}(1/k)$ convergence rate is established [26,50,27]. Some applications in distributed consensus problems include [7,68,59,46,2,47]. More recent convergence results on multi-block convex ADMM can be found in [24,25,23,9,42, 38–40,8,30,10,41].

2.2 ADMM for Nonconvex Problems

The convergence of ADMM has been observed for many nonconvex problems with various applications in matrix completion and factorization [72,57,74, 73], optimal power flow [61,16,45], asset allocation [69], and polynomial optimization [33], among others. For convergence theory, several conditions have been proposed to guarantee convergence on structured nonconvex problems that can be abstracted in the following form

$$\begin{aligned} \min_{x_1,\cdots,x_p,z} \ & \sum_{i=1}^{p} f_i(x_i) + h(z) + g(x_1,\cdots,x_p,z) \\ \text{s.t.} \ & \sum_{i=1}^{p} A_i x_i + Bz = b,\ x_i \in \mathcal{X}_i \ \forall i \in [p]. \end{aligned} \tag{4}$$

We summarize some convergence conditions in Table 1. For instance, Hong et al. [31] studied ADMM for nonconvex consensus and sharing problems under cyclic or randomized update order. Li and Pong [37] and Guo et al. [22] studied two-block ADMM, where one of the blocks is the identity matrix. One of the most general frameworks for proving convergence of multi-block ADMM is proposed by Wang et al. [67], where the authors showed a global subsequential convergence with a rate of $o(1/\sqrt{k})$. A more recent work by Themelis

and Patrinos [62] established a primal equivalence of nonconvex ADMM and Douglas-Rachford splitting.

Another line of research explores some variants of ADMM. Wang et al. [66,65] studied the nonconvex Bregman-ADMM, where a Bregman divergence term is added to the augmented Lagrangian function during each block update to facilitate the descent of certain potential function. Gonçalves, Melo, and Monteiro [20] provided an alternative convergence rate proof of proximal ADMM applied to convex problems, which was shown to be an instance of a more general non-Euclidean hybrid proximal extragradient framework. The two-block, multi-block, and Jacobi-type extensions of this framework to nonconvex problems can be found in [21,49,48], where an iteration complexity of $\mathcal{O}(1/\sqrt{k})$ was also established. Jiang et al. [32] proposed two variants of proximal ADMM. Some proximal terms are added to the first p block updates; for the last block, either a gradient step is performed, or a quadratic approximation of the augmented Lagrangian is minimized.

Table 1: Comparisons of the Nonconvex ADMM Literature

	p	f_i's	$\mathcal{X}_i$'s	h	g	A_i's	B
[31]	1	convex	convex	smooth	-	-	I
	≥ 2	convex smooth nonconvex	convex	smooth	-	full col.	I
[37]	1	l.s.c		$\nabla^2 h$ bounded	-	I	full row
[22]	1	l.s.c		smooth	-	full col.	I
[66]	1	l.s.c & $f_1 + h$ subanalytic		smooth	-	full col.	full row
[65]	2	l.s.c & $f_1 + f_2 + h$ subanalytic		smooth	-	-	full row
[67]	≥ 2	l.s.c & restricted prox-regular ∂f_1 bounded & $f_{>1}$'s p.w. linear		smooth		$\mathrm{Im}([A,b]) \subseteq \mathrm{Im}(B)$ Lip. sub-min path	
[21,49]	1,2	l.s.c		$\approx$smooth	-	$\mathrm{Im}([A,b]) \subseteq \mathrm{Im}(B)$	
[48]	≥ 2	l.s.c		smooth	-	$\mathrm{Im}([A,b]) \subseteq \mathrm{Im}(B)$	
[32]	≥ 2	Lipschitz continuous l.s.c	compact	smooth		-	I **or** full row

l.s.c: lower semi-continuous; smooth: Lipschitz differentiable; full col./row: full column/row rank

For general nonconvex and nonsmooth problems, we note that the convergence of ADMM relies on the following two conditions.

- Condition 1: Denote $A := [A_0, \cdots, A_p]$, then $\mathrm{Im}([A,b]) \subseteq \mathrm{Im}(B)$.
- Condition 2: The last block objective function $h(z)$ is Lipschitz differentiable.

Due to the sequential update order of ADMM, z^k is obtained after x^k is calculated. If Condition 1 on the images of A and B is not satisfied, then it is possible that x^k converges to some x^* such that there is no z^* satisfying $Ax^* + Bz^* = b$. In addition, Condition 2 provides a way to control dual iterates by primal iterates via the optimality condition of the z-subproblem. This relation requires unconstrained optimality condition of z-update, so the

last block variable z cannot be constrained elsewhere. See also [67] for some relevant discussions. As indicated from Table 1, these two conditions (and their variants) are almost necessary for ADMM to converge in the absence of convexity. We also note that, even for convex problems, these two conditions are used to relax the strong convexity assumption in the objective [40] or accelerate ADMM with $\mathcal{O}(1/k^2)$ iteration complexity [63].

It turns out that the two conditions and the two properties we mentioned in Section 1.2 may conflict each other. By Property 1, the image of A strictly constrains the image of B, so by Condition 1, we should update local variables after the global variable in each ADMM iteration to ensure feasibility. However, by Property 2, each local variable is subject to some local constraints, so Condition 2 cannot be satisfied; technically speaking, we cannot utilize the unconstrained optimality condition of the last block to link primal and dual variables, which again makes it difficult to ensure primal feasibility of the solution. When ADMM is directly applied to nonconvex problems, divergence is indeed observed [61,45,67]. As a result, for many applications in the form of (3) where the above two conditions are not available, the ADMM framework cannot guarantee convergence.

After completing a draft of this paper, we were informed of a ADMM-based approach in [32], where the authors proposed to solve the *relaxed* problem of (3)

$$\min_{x\in\mathcal{X},\bar{x}\in\tilde{\mathcal{X}},z} \quad f(x)+\frac{\beta(\epsilon)}{2}\|z\|^2 \quad \text{s.t.} \quad Ax+B\bar{x}+z=0. \tag{5}$$

Notice first that, as proved in [32], in order to achieve a desired feasibility with $\|Ax+B\bar{x}\|=\mathcal{O}(\epsilon)$, the coefficient $\beta(\epsilon)$ and ADMM penalty need to be as large as $\mathcal{O}(1/\epsilon^2)$. Such large parameters may lead to slow convergence and large optimality gaps. Also notice that, applying ADMM to (5) may produce an approximate stationary solution to (3), even when the problem is infeasible to begin with. As we will show in Section 4, our proposed two-level algorithm is able to achieve the same order of iteration complexity as the reformulation (5) and the one-level ADMM approach proposed in [32], and meanwhile the proposed two-level algorithm provides information on ill conditions and infeasibility; in Section 6, we empirically demonstrate with computation that the proposed algorithm robustly converges on large-scale constrained nonconvex programs with a faster speed and obtains solutions with higher qualities.

2.3 Other Distributed Algorithms

Some other distributed algorithms not based on ADMM are also studied in the literature. Hong [29] introduced a proximal primal-dual algorithm for distributed optimization problems, where a proximal term is added to cancel out cross-product terms in the augmented Lagrangian function. Lan and Zhou [36] proposed a randomized incremental gradient algorithm for a class of convex problems over a multi-agent network. Lan and Yang [35] proposed accelerated

stochastic algorithms for nonconvex finite-sum and multi-block problems; interestingly, the analysis for the multi-block problem also requires the last block variable to be unconstrained with an invertible coefficient matrix and a Lipschitz differentiable objective, which further confirms the necessity of Conditions 1 and 2. We end this subsection with a recent work by Shi et al. [58]. They studied the problem

$$\min_{\mathbf{x},\mathbf{y}} \quad f(\mathbf{x},\mathbf{y}) + \sum_{j=1}^{m} \tilde{\phi}_j(\mathbf{y}_j) \quad \text{s.t.} \quad h(\mathbf{x},\mathbf{y}) = 0, \quad g_i(\mathbf{x}_i) \leq 0, \quad \mathbf{x}_i \in \mathcal{X}_i \quad \forall i \in [n]. \tag{6}$$

The variables $\mathbf{x}$ and $\mathbf{y}$ are divided into n and m subvectors, respectively. $f(\mathbf{x},\mathbf{y})$, $h(\mathbf{x},\mathbf{y})$, $g_i(\mathbf{x}_i)$ are continuously differentiable, $\tilde{\phi}_j(\mathbf{y}_j)$ is a composite function, and $\mathcal{X}_i$'s are convex. The authors proposed a doubly-looped penalty dual decomposition method (PDD). The overall algorithm used the ALM framework, where the coupling constraint $h(\mathbf{x},\mathbf{y}) = 0$ is relaxed and each ALM subproblem is solved by a randomized block update scheme. We note that randomization is crucial in their convergence analysis, and a deterministic implementation of the inner-level algorithm for solving the ALM subproblem may not converge when nonconvex functional constraints are present.

3 A Key Reformulation and A Two-level Algorithm

We say $(x^*, \bar{x}^*, y^*) \in \mathbb{R}^{n_1} \times \mathbb{R}^{n_2} \times \mathbb{R}^m$ is a *stationary point* of problem (3) if it satisfies the following condition

$$0 \in \nabla f(x^*) + A^\top y^* + N_{\mathcal{X}}(x^*), \tag{7a}$$

$$0 \in B^\top y^* + N_{\bar{\mathcal{X}}}(\bar{x}^*), \tag{7b}$$

$$0 = Ax^* + B\bar{x}^*; \tag{7c}$$

or equivalently, $0 \in \partial L(x^*, \bar{x}^*, y^*)$, where

$$L(x, \bar{x}, y) := f(x) + \mathbb{I}_{\mathcal{X}}(x) + \mathbb{I}_{\bar{\mathcal{X}}}(\bar{x}) + \langle y, Ax + B\bar{x}\rangle. \tag{8}$$

In equations (7) and (8), the notation $N_{\mathcal{X}}(x)$ denotes the general normal cone of $\mathcal{X}$ at $x \in \mathcal{X}$ [56, Def 6.3], and $\partial L(\cdot)$ denotes the general subdifferential of $L(\cdot)$ [56, Def 8.3]. Some properties and calculus rules of normal cones and the general subdifferential can be found in [56, Chap 6, 8, 10].

It can be shown that if $(x^*, \bar{x}^*)$ is a local minimum of (3) and satisfies some mild regularity condition, then condition (7) is satisfied [56, Thm 8.15]. If $\mathcal{X}$ and $\bar{\mathcal{X}}$ are defined by finitely many continuously differentiable constraints, then condition (7) is equivalent to the well-known KKT condition of problem (3) under some constraint qualification. Therefore, condition (7) can be viewed as a generalized first-order necessary optimality condition for nonsmooth constrained problems. Our goal is to find such a stationary point $(x^*, \bar{x}^*, y^*)$ for problem (3).

3.1 A Key Reformulation

As analyzed in the previous section, since directly applying ADMM to a distributed formulation of the general constrained nonconvex problem (3) cannot guarantee convergence without using the relaxation scheme in [32], we want to go beyond the standard ADMM framework. We propose two steps for achieving this. The first step is taken in this subsection to propose a new reformulation, and the second step is taken in the next subsection to propose a new two-level algorithm for the new reformulation.

We consider the following reformulation of (3)

$$\min_{x\in\mathcal{X},\bar{x}\in\bar{\mathcal{X}},z} \quad f(x) \quad \text{s.t.} \quad Ax+B\bar{x}+z=0,\ z=0. \tag{9}$$

The idea of adding a slack variable $z\in\mathbb{R}^m$ has two consequences. The first consequence is that the linear coupling constraint $Ax+B\bar{x}+z=0$ now has three blocks, and the last block is an identity matrix I_m, whose image is the whole space. Given any x and $\bar{x}$, we can always let $z=-Ax-B\bar{x}$ to make the constraint satisfied. The second consequence is that the artificial constraint $z=0$ can be treated separately from the coupling constraint. Notice that a direct application of ADMM to problem (9) still does not guarantee convergence since Conditions 1 and 2 are not satisfied yet. So it is necessary to separate the linear constraints into two levels. If we ignore $z=0$ for the moment, existing techniques in ADMM analysis can be applied to the rest of the problem. Since we want to utilize the unconstrained optimality condition of the last block, we can relax $z=0$. This observation motivates us to choose ALM. To be more specific, consider the problem

$$\min_{x\in\mathcal{X},\bar{x}\in\bar{\mathcal{X}},z} \quad f(x)+\langle\lambda^k,z\rangle+\frac{\beta^k}{2}\|z\|^2 \quad \text{s.t.} \quad Ax+B\bar{x}+z=0, \tag{10}$$

which is obtained by dualizing constraint $z=0$ with $\lambda^k\in\mathbb{R}^m$ and adding a quadratic penalty $\frac{\beta^k}{2}\|z\|^2$ with $\beta^k>0$. The augmented Lagrangian term $\langle\lambda^k,z\rangle+\frac{\beta^k}{2}\|z\|^2$ can be viewed as an objective function in variable z, which is not only Lipschitz differentiable but also strongly convex. Problem (10) can be solved by a three-block ADMM in a distributed fashion when a separable structure is available. Notice that the first-order optimality condition of problem (10) at a stationary solution $(x^k,\bar{x}^k,z^k,y^k)$ is

$$0\in\nabla f(x^k)+A^\top y^k+N_{\mathcal{X}}(x^k), \tag{11a}$$
$$0\in B^\top y^k+N_{\bar{\mathcal{X}}}(\bar{x}^k), \tag{11b}$$
$$0=\lambda^k+\beta^k z^k+y^k, \tag{11c}$$
$$0=Ax^k+B\bar{x}^k+z^k. \tag{11d}$$

However, such a solution may not satisfy primal feasibility $Ax+B\bar{x}=0$, which is the only difference from the optimality condition (7) (note that (11c) is analogous to the dual feasibility in variable z in the KKT condition). Fortunately,

the ALM offers a scheme to drive the slack variable z to zero by updating λ and we can expect iterates to converge to a stationary point of the original problem (3). In summary, reformulation (9) separates the complication of the original problem into two levels, where the inner level (10) provides a formulation that simultaneously satisfies Conditions 1 and 2, and the outer level drives z to zero. We propose a two-level algorithmic architecture in the next subsection to realize this.

3.2 A Two-level Algorithm

The proposed algorithm consists of two levels, both of which are based on the augmented Lagrangian framework. The inner-level algorithm is described in Algorithm 1, which uses a three-block ADMM to solve problem (10) and its iterates are indexed by t. The outer-level algorithm is described in Algorithm 2 with iterates indexed by k.

Given $\lambda^k \in \mathbb{R}^m$ and $\beta^k > 0$, the augmented Lagrangian function associated with the k-th inner-level problem (10) is defined as

$$\begin{aligned} L_{\rho^k}(x, \bar{x}, z, y) :=& f(x) + \mathbb{I}_{\mathcal{X}}(x) + \mathbb{I}_{\bar{\mathcal{X}}}(\bar{x}) + \langle \lambda^k, z\rangle + \frac{\beta^k}{2}\|z\|^2 \\ &+ \langle y, Ax + B\bar{x} + z\rangle + \frac{\rho^k}{2}\|Ax + B\bar{x} + z\|^2, \end{aligned} \tag{12}$$

where $y \in \mathbb{R}^m$ is the dual variable for constraint $Ax + B\bar{x} + z = 0$ and ρ^k is a penalty parameter for ADMM. In view of (11), the k-th inner-level ADMM aims to find an approximate stationary solution $(x^k, \bar{x}^k, z^k, y^k)$ of (10) in the sense that there exist d_1^k, d_2^k, and d_3^k such that

$$d_1^k \in \nabla f(x^k) + A^\top y^k + N_{\mathcal{X}}(x^k), \tag{13a}$$
$$d_2^k \in B^\top y^k + N_{\bar{\mathcal{X}}}(\bar{x}^k), \tag{13b}$$
$$0 = \lambda^k + \beta^k z^k + y^k, \tag{13c}$$
$$d_3^k = Ax^k + B\bar{x}^k + z^k, \tag{13d}$$
$$\|d_i^k\| \le \epsilon_i^k,\ \forall i \in [3], \tag{13e}$$

where ϵ_i^k's are positive tolerances. The optimality conditions of x^t in Line 5 and $\bar{x}^t$ in Line 7 of Algorithm 1 read:

$$\begin{aligned} 0 \in& \nabla f(x^t) + A^\top y^{t-1} + \rho^k A^\top (Ax^t + B\bar{x}^{t-1} + z^{t-1}) + N_{\mathcal{X}}(x^t), \\ 0 \in& B^\top y^{t-1} + \rho^k B^\top (Ax^t + B\bar{x}^t + z^{t-1}) + N_{\bar{\mathcal{X}}}(\bar{x}^t). \end{aligned}$$

With the dual update in Line 11, we can see that

$$\begin{aligned} -\rho^k A^\top (B\bar{x}^{t-1} + z^{t-1} - B\bar{x}^t - z^t) \in& \nabla f(x^t) + A^\top y^t + N_{\mathcal{X}}(x^t), \\ -\rho^k B^\top (z^{t-1} - z^t) \in& B^\top y^t + N_{\bar{\mathcal{X}}}(\bar{x}^t). \end{aligned}$$

As a result, Algorithm 1 can be terminated if it finds $(x^t, \bar{x}^t, z^t)$ such that

$$\|\rho^k A^\top (B\bar{x}^{t-1} + z^{t-1} - B\bar{x}^t - z^t)\| \le \epsilon_1^k, \tag{14a}$$

$$\|\rho^k B^\top (z^{t-1} - z^t)\| \le \epsilon_2^k, \tag{14b}$$

$$\|Ax^t + B\bar{x}^t + z^t\| \le \epsilon_3^k. \tag{14c}$$

Notice that ρ^k does not appear in (14c), so we can use different tolerances for the above three measures. Since (13c) is always maintained by ADMM with $(y^k, z^k) = (y^t, z^t)$, a solution satisfying (14) is an approximate stationary solution to problem (10) by assigning $(x^k, \bar{x}^k, z^k, y^k) := (x^t, \bar{x}^t, z^t, y^t)$.

Algorithm 1 : The k-th inner-level ADMM

1: **Input** $(\lambda^k, \beta^k, \epsilon_1^k, \epsilon_2^k, \epsilon_3^k) \in \mathbb{R}^m \times \mathbb{R}^4_{++}$;
2: **initialize** $(x^0, \bar{x}^0, z^0, y^0) \in \mathcal{X} \times \bar{\mathcal{X}} \times \mathbb{R}^m \times \mathbb{R}^m$ with $\lambda^k + \beta^k z^0 + y^0 = 0$, $\rho^k = 2\beta^k$;
3: **for** $t = 1, 2, 3, \cdots$ **do**
4: /* First block update (parallelize over subvectors of x) */
5: obtain a stationary x^t such that $0 \in \partial_x L_{\rho^k}(x^t, \bar{x}^{t-1}, z^{t-1}, y^{t-1})$;
6: /* Second block update (parallelize over components of $\bar{x}$) */
7: $\bar{x}^t \leftarrow \operatorname{argmin}_{\bar{x}} L_{\rho^k}(x^t, \bar{x}, z^{t-1}, y^{t-1})$;
8: /* Third block update (parallelize over subvectors of z) */
9: $z^t \leftarrow \operatorname{argmin}_z L_{\rho^k}(x^t, \bar{x}^t, z, y^{t-1})$;
10: /* Inner dual update (parallelize over subvectors of y) */
11: $y^t \leftarrow y^{t-1} + \rho^k(Ax^t + B\bar{x}^t + z^t)$;
12: **if** stopping criteria (14) is satisfied **then**
13: **return** $(x^t, \bar{x}^t, z^t, y^t)$;
14: **break**.
15: **end if**
16: **end for**

The first block update in Algorithm 1 reads as

$$\min_{x \in \mathcal{X}} \quad f(x) + \langle y^{t-1}, Ax + B\bar{x}^{t-1} + z^{t-1} \rangle + \frac{\rho^k}{2} \|Ax + B\bar{x}^{t-1} + z^{t-1}\|^2, \tag{15}$$

so line 5 of Algorithm 1 searches for a stationary solution x^t of the constrained problem (15). The second and third block updates in lines 7 and 9 admit closed form solutions, so in view of the network flow problem (1), the proposed reformulation (9) does not introduce additional computational burden. All primal and dual updates in Algorithm 1 can be implemented in parallel as f and $\mathcal{X}$ admit separable structures. In each ADMM iteration, agents solve their own local problems independently and only need to communicate with their immediate neighbors. We resolve this by updating λ and β, which is referred as outer-level iterations indexed by k in Algorithm 2.

In Algorithm 2, we choose some predetermined bounds $[\underline{\lambda}, \overline{\lambda}]$ and explicitly project the "true" dual variable $\lambda^k + \beta^k z^k$ onto this hyper-cube to obtain λ^{k+1} used in the next outer iteration. Such safeguarding technique is essential to establish the global convergence of ALM [1,43]. We increase the outer-level penalty β^k if there is no significant improvement in reducing $\|z^k\|$.

Algorithm 2 : Outer-level ALM

Initialize $\lambda^1 \in [\underline{\lambda}, \overline{\lambda}]$ where $\underline{\lambda}, \overline{\lambda} \in \mathbb{R}^m$ and $\overline{\lambda} - \underline{\lambda} \in \mathbb{R}^m_{++}$, $\beta^1 = \beta^0\gamma$ for some $\beta^0 \geq \frac{1}{4}$ and $\gamma > 1$, $\omega \in [0, 1)$, $\{\epsilon_i^k\} \subset \mathbb{R}_+$ with $\epsilon_i^k \to 0$ for $i \in [3]$;
for $k = 1, 2, 3, \cdots$ **do**
  obtain $(x^k, \bar{x}^k, z^k, y^k)$ from Algorithm 1 with input $(\lambda^k, \beta^k, \epsilon_1^k, \epsilon_2^k, \epsilon_3^k)$;
  $\lambda^{k+1} \leftarrow \text{Proj}_{[\underline{\lambda}, \overline{\lambda}]}(\lambda^k + \beta^k z^k)$;
  if $\|z^k\| \leq \omega \|z^{k-1}\|$ **then**
    $\beta^{k+1} \leftarrow \beta^k$,
  else
    $\beta^{k+1} \leftarrow \gamma\beta^k$;
  end if
end for

Before proceeding to the next section, we note that the key reformulation (9) is inspired by the hope to reconcile the conflict between the two properties and the two condition so that ADMM can be applied. The introduction of additional variable z is not necessary in the sense that any method that achieves distributed computation for the subproblem

$$\min_{x \in \mathcal{X}, \bar{x} \in \bar{\mathcal{X}}} f(x) + \langle \lambda^k, Ax + B\bar{x} \rangle + \frac{\beta^k}{2} \|Ax + B\bar{x}\|^2 \tag{16}$$

can be embedded inside the ALM framework. The aforementioned PDD method [58] is such an approach. There are some other update schemes [6, 71] that can handle functional constraints in (16), assuming that the (Euclidean) projection oracle onto the nonconvex set $\mathcal{X}$ is available. It would be interesting to compare their performances with ADMM when used in the inner level, and we leave this to future work. Meanwhile, as we will demonstrate in Section 6, the proposed two-level algorithm preserves the desirable properties of ADMM in practice, such as fast convergence in early stages and scalability to handle large-scale problems.

4 Global Convergence

In this section, we prove global convergence and convergence rate of the proposed two-level algorithm. Starting from any initial point, iterates generated by the proposed algorithm have a limit point; every limit point is a stationary solution to the original problem under some mild condition. In particular, we make the following assumptions.

Assumption 1 *Problem* (9) *is feasible and the set of stationary points satisfying* (7) *is nonempty.*

Assumption 2 *The objective function $f : \mathbb{R}^n \to \mathbb{R}$ is continuously differentiable, $\mathcal{X} \subseteq \mathbb{R}^n$ is a compact set, and $\bar{\mathcal{X}}$ is convex and compact.*

Assumption 3 *Given λ^k, β^k, and ρ^k, the first block update can find a stationary solution x^t such that $0 \in \partial_x L_{\rho^k}(x^t, \bar{x}^{t-1}, z^{t-1}, y^{t-1})$ and*

$$L_{\rho^k}(x^t, \bar{x}^{t-1}, z^{t-1}, y^{t-1}) \leq L_{\rho^k}(x^{t-1}, \bar{x}^{t-1}, z^{t-1}, y^{t-1}) < +\infty$$

for all $t \in \mathbb{Z}_{++}$.

We give some comments below. Assumption 1 ensures the feasibility of problem (9), which is standard. Though it is desirable to design an algorithm that can guarantee feasibility of the limit point, usually this is too much to ask: the powerful ALM may converge to an infeasible limit point even if the original problem is feasible. If this situation happens, or problem (9) is infeasible in the first place, our algorithm will converge to a limit point that is stationary to some problem, as stated in Theorem 1. The compactness required in Assumption 2 ensures that the sequence generated by our algorithm stays bounded, and can be dropped if the existence of a limit point is directly assumed or derived from elsewhere. We do not make any explicit assumptions on matrices A and B in this section, and our analysis does not rely on any convenient structures that A and B may process, such as full row or column rank.

For Assumption 3, we note that finding a stationary point usually can be achieved at the successful termination of some nonlinear solvers. In addition, the state-of-the-art nonlinear solver IPOPT [64] will accept a trial point if either the objective or the constraint violation is decreased in each iteration. In step 1 of Algorithm 1, since x^{t-1} is already a feasible solution, if we start from x^{t-1}, it is reasonable to expect a new stationary point x^t is reached with an improved objective value. Assumption 3 is slightly weaker and more realistic than assuming that the nonconvex subproblem can be solved globally, which is commonly adopted in the nonconvex ADMM literature.

In Section 4.1, we show that each inner-level ADMM converges to a solution that approximately satisfies the stationary condition (11) of problem (10). This sequence of solutions that we obtain at termination of the inner ADMM is referred as outer-level iterates. Then in Section 4.2, we firstly characterize limit points of outer-level iterates, whose existence is guaranteed. Then we show that a limit point is stationary to problem (3) if some mild constraint qualification is satisfied.

4.1 Convergence of Inner-level Iterations

In this subsection, we show that, by applying the three-block ADMM to problem (10), we will get an approximate stationary point $(x^k, \bar{x}^k, z^k, y^k)$ satisfying the approximate stationary condition (13). The convergence of the inner-level ADMM in this subsection uses some techniques from the literature, e.g., [67]. We present a self-contained proof in the appendix and demonstrate that the descent oracle assumed in Assumption 3 relaxes the global optimality of subproblems without affecting the overall convergence.

Proposition 1 *Suppose Assumptions 2-3 hold. The k-th inner-level ADMM of Algorithm 1 terminates, i.e., the stopping criteria* (14) *is satisfied, in at most*

$$T_k := \left\lceil \frac{8\max\{\|A\|^2, \|B\|^2, 1\}\beta^k(\overline{L}_k - \underline{L})}{\min\{\epsilon_1^k, \epsilon_2^k, \epsilon_3^k\}^2} \right\rceil \tag{17}$$

iterations, where $\overline{L}_k := L_{\rho^k}(x^0, \bar{x}^0, z^0, y^0)$ and $\underline{L} \in \mathbb{R}$ is a finite constant independent of outer-level index k.

Proof See Appendix A.1. □

In particular, the approximate stationary condition (13) is satisfied with the solution returned by ADMM.

4.2 Convergence of Outer-level Iterations

In this subsection, we prove the convergence of outer-level iterations. In general, when the method of multipliers is used as a global method, there is no guarantee that the constraint being relaxed can be satisfied at the limit. Due to the special structure of our reformulation, we are able to give a characterization of limit points of outer-level iterates.

Theorem 1 *Suppose Assumptions 2-3 hold. Let $\{(x^k, \bar{x}^k, z^k, y^k)\}_{k\in\mathbb{Z}_{++}}$ be the sequence of outer-level iterates of Algorithm 2 satisfying condition* (13)*. Then the sequence of the primal solutions $\{(x^k, \bar{x}^k, z^k)\}_{k\in\mathbb{Z}_{++}}$ are bounded, and every limit point $(x^*, \bar{x}^*, z^*)$ of this sequence satisfies one of the following:*

1. *$(x^*, \bar{x}^*)$ is feasible for problem* (3)*, i.e., $z^* = 0$;*
2. *$(x^*, \bar{x}^*)$ is a stationary point of the problem*

$$\min_{x\in\mathcal{X}, \bar{x}\in\bar{\mathcal{X}}} \quad \frac{1}{2}\|Ax + B\bar{x}\|^2. \tag{18}$$

Proof See Appendix A.2. □

Theorem 1 gives a complete characterization of limit points of outer-level iterates. If the limit point is infeasible, i.e. $z^* \neq 0$, then $(x^*, \bar{x}^*)$ is a stationary point of the problem (18). This is also the case if problem (3) is infeasible, i.e. the feasible region defined by $\mathcal{X}$ and $\bar{\mathcal{X}}$ does not intersect the affine plane $Ax+B\bar{x} = 0$, since each inner-level problem (10) is always feasible and the first case in Theorem 1 cannot happen. We also note that even if $(x^*, \bar{x}^*)$ falls into the second case of Theorem 1, it is still possible that the associated $z^* = 0$, but then $(x^*, \bar{x}^*)$ will be some irregular feasible solution. In both cases, we believe $(x^*, \bar{x}^*)$ generated by the two-level algorithm has its own significance and may provide some useful information regarding the problem structure. Since stationarity and optimality are maintained in all subproblems, we should expect that any feasible limit point of the outer-level iterates is stationary for the original problem. As we will prove in the next theorem, this is indeed the case if some mild constraint qualification is satisfied.

Theorem 2 *Suppose Assumptions 1-3 hold. Let* $(x^*, \bar{x}^*, z^*)$ *be a limit point of the outer-level iterates* $\{(x^k, \bar{x}^k, z^k)\}_{k\in\mathbb{Z}_{++}}$ *of Algorithm 2. If* $\{y^k\}_{k\in\mathbb{Z}_{++}}$ *has a limit point* y^* *along a subsequence converging to* $(x^*, \bar{x}^*, z^*)$*, then* $(x^*, \bar{x}^*, y^*)$ *is a stationary point of problem* (3) *satisfying stationary condition* (7)*.*

Proof See Appendix A.3. □

In Theorem 2, we assume the dual variable $\{y^k\}$ has a limit point y^*. Since by (38) we have $\lambda^k + \beta^k z^k + y^k = 0$, the "true" multiplier $\tilde{\lambda}^{k+1} := \lambda^k + \beta^k z^k$ also has a limit point. We note that the existence of a limit point can be ensured by the existence of a bounded dual subsequence, which is known as the sequentially bounded constraint qualification (SBCQ) [44]. More specifically in the context of smooth nonlinear problems, the constant positive linear dependence (CPLD) condition proposed by Qi and Wei [54] also guarantees that the sequence of dual variables has a bounded subsequence. Therefore, we think our assumption of y^* is analogous to some constraint qualification in the KKT condition for smooth problems, and does not restrict the field where our algorithm is applicable.

We also give some comments regarding the predetermined bound $[\underline{\lambda}, \overline{\lambda}]$ on outer-level dual variable λ. In principle, the bound should be chosen large enough at the beginning of the algorithm. Otherwise λ^k will probably stay at $\underline{\lambda}$ or $\overline{\lambda}$ all the time; in this case, the outer-level ALM automatically converts to the penalty method, which usually requires β^k to go to infinity, because, in general, exact penalization does not hold for a quadratic penalty function. In contrast, a proper choice of the dual variable can compensate asymptotic exactness even when the penalty function is not sharp at the origin. In terms of convergence analysis, one may notice that the choice of λ is actually not that important: if we set $\lambda^k = 0$ for all k, the analysis can still go through. This is because in the framework of ALM, the dual variable λ is closely related to local optimal solutions. While we study global convergence, it is not clear which local solution the algorithm will converge to, so the role of λ is not significant. It seems difficult to establish the uniform boundedness of dual variables without the projection step, especially when there are nonconvex constraints

In Section 5, we will show our algorithm inherits some nice local convergence properties of ALM, where λ does play an important role, and in Section 6, we will demonstrate that keeping λ indeed enables the algorithm to converge faster than the penalty method.

4.3 Iteration Complexity

In this subsection, we provide an iteration complexity analysis of the proposed algorithm. In view of (7), our goal is to give a complexity bound on the number of ADMM iterations for finding an ϵ-stationary solution $(x^K, \bar{x}^K, y^K)$ in the

sense that there exist d_1, d_2, d_3 such that

$$d_1 \in \nabla f(x^K) + A^\top y^K + N_{\mathcal{X}}(x^K), \tag{19a}$$
$$d_2 \in B^\top y^K + N_{\bar{\mathcal{X}}}(\bar{x}^K), \tag{19b}$$
$$d_3 = Ax^K + B\bar{x}^K, \tag{19c}$$
$$\max\{\|d_1\|, \|d_2\|, \|d_3\|\} \le \epsilon. \tag{19d}$$

In order to illustrate the main result in a concise and clear way, we slightly modify the outer-level Algorithm 2 as follows.

Algorithm 3 : Modified Outer-level ALM

1: **Initialize** $\lambda^1 \in [\underline{\lambda}, \overline{\lambda}]$ where $\underline{\lambda}, \overline{\lambda} \in \mathbb{R}^m$ and $\overline{\lambda} - \underline{\lambda} \in \mathbb{R}^m_{++}$, $\beta^1 = \beta^0\gamma$ for some $\beta^0 \ge \frac{1}{4}$ and $\gamma > 1$, $\epsilon > 0$;
2: **for** $k = 1, 2, 3, \cdots$ **do**
3: obtain $(x^k, \bar{x}^k, z^k, y^k)$ from Algorithm 1 with input $(\lambda^k, \beta^k, \epsilon, \epsilon, \epsilon/2)$;
4: $\lambda^{k+1} \leftarrow \mathrm{Proj}_{[\underline{\lambda}, \overline{\lambda}]}(\lambda^k + \beta^k z^k)$, $\beta^{k+1} \leftarrow \gamma\beta^k$;
5: **end for**

In Algorithm 3, we choose some tolerance $\epsilon > 0$ and apply the stopping criteria (14) with $\epsilon_1^k = \epsilon_2^k = 2\epsilon_3^k = \epsilon$ for the k-th inner-level ADMM. For the ease of the analysis, we multiply the outer-level penalty β^k by some $\gamma > 1$ in each outer-iteration, instead of checking the improvement in primal feasibility. Moreover, we add the following technical assumption.

Assumption 4 *There exists some $\overline{L} \in \mathbb{R}$ such that $L_{\rho^k}(x^0, \bar{x}^0, z^0, y^0) \le \overline{L}$ for all $k \in \mathbb{Z}_{++}$.*

Remark 1 This assumption can be satisfied if ADMM can make significant progress in reducing $\|z^k\|$ or equivalently $\|Ax^k + B\bar{x}^k\|$. Another naive implementation can be seen as follows: suppose a feasible point $(x, \bar{x})$ is known a priori, i.e., $(x, \bar{x}) \in \mathcal{X} \times \bar{\mathcal{X}}$, and $Ax + B\bar{x} = 0$, then the initialization of the k-ADMM with $(x^0, \bar{x}^0, z^0, y^0) = (x, \bar{x}, 0, -\lambda^k)$ guarantees that $L_{\rho^k}(x^0, \bar{x}^0, z^0, y^0) \le \overline{L}$, where $\overline{L} = \max_{x \in \mathcal{X}} f(x)$.

Theorem 3 *Under Assumptions 1-4, Algorithm 3 finds an ϵ-stationary solution $(x^K, \bar{x}^K, y^K)$ of (3) in the sense of (19) in no more than $\mathcal{O}\left(1/\epsilon^4\right)$ inner ADMM iterations. Furthermore, if $\hat{\lambda}^k := \lambda^k + \beta^k z^k$ is bounded, then the iteration complexity can be improved to $\mathcal{O}\left(1/\epsilon^3\right)$.*

Proof See Appendix A.4. □

We acknowledge that $\{\hat{\lambda}^k\}_k$ may not be bounded for some applications. The second part of Theorem 3 (as well as Theorem 4 to be presented next) aims to reasonably justify the performance of the proposed algorithm under the boundedness condition.

4.4 Extension to Multi-block Problems

In this section, we will discuss the extension of the two-level framework to the more general class of multi-block problems (4). In particular, we are interested in the case where Conditions 1 and 2 are not satisfied. As we mentioned earlier, Jiang et al. [32] proposed to solve the following perturbed problem of (4):

$$\begin{aligned} \min_{x_1,\cdots,x_p,z} \ & \sum_{i=1}^{p} f_p(x_i) + g(x_1,\cdots,x_p) + \lambda^\top z + \frac{\beta}{2}\|z\|^2 \\ \text{s.t.} \ & \sum_{i=1}^{p} A_i x_i + z = b, \ x_i \in \mathcal{X}_i \ \forall i \in [p]. \end{aligned} \tag{20}$$

for $\lambda = 0$, where f_i's are lower semi-continuous, and f_p and g are Lipschitz differentiable. Notice that we change h and B in (4) to f_p and A_p for ease of presentation. The iteration complexity for this one-level workaround is $\mathcal{O}\left(1/\epsilon^4\right)$ when the dual variable is bounded, and $\mathcal{O}\left(1/\epsilon^6\right)$ otherwise. In contrast, we can apply our two-level framework to the multi-block problem (4) as well: with some initial guess λ and moderate β, we solve (20) approximately using ADMM, and then we update λ and β. We define dual residual similarly as in (14a)-(14b) for each block variable, and ϵ-stationary solution as a pair of primal-dual points where the primal residual ($\|\sum_{i=1}^{p} A_i x_i - b\|$) and dual residuals (with respect to each primal block) are less than some $\epsilon > 0$. An extension of the two-level framework is presented in Algorithm 4 below.

Algorithm 4 : Extension to Multi-block Problems

1: **Initialize** $\lambda^1 \in [\underline{\lambda}, \overline{\lambda}]$ where $\underline{\lambda}, \overline{\lambda} \in \mathbb{R}^m$ and $\overline{\lambda} - \underline{\lambda} \in \mathbb{R}^m_{++}$, $\beta^1 = \beta^0\gamma$ for some $\beta^0 > 0$ and $\gamma > 1$, $\epsilon > 0$;
2: **for** $k = 1, 2, 3, \cdots$ **do**
3: obtain an $(\epsilon/2)$-stationary solution $(x_1^k, \cdots, x_p^k, z^k, y^k)$ of (20) with $(\lambda, \beta) = (\lambda^k, \beta^k)$ by proximal ADMM-m or ADMM-g [32];
4: $\lambda^{k+1} \leftarrow \mathrm{Proj}_{[\underline{\lambda}, \overline{\lambda}]}(\lambda^k + \beta^k z^k)$, $\beta^{k+1} \leftarrow \gamma\beta^k$;
5: **end for**

Theorem 4 *Under Assumption 4, Algorithm 4 finds an ϵ-stationary solution of (4) in no more than $\mathcal{O}(1/\epsilon^6)$ ADMM iterations. Furthermore, if $\hat{\lambda}^k := \lambda^k + \beta^k z^k$ is bounded, then the iteration complexity can be improved to $\mathcal{O}\left(1/\epsilon^4\right)$.*

Proof See Appendix A.5. □

Although the proposed algorithm invokes a series of ADMM with varying outer-level dual variables and penalties, Theorem 4 suggests that its iteration complexity for finding a stationary solution is no worse than that of the single-looped ADMM variant proposed in [32]. In Section 5, local convergence results are presented as an alternative perspective to help us understand the behavior of the proposed algorithm.

5 Local Convergence

We show in this section that the proposed algorithm inherits some nice local convergence properties of the augmented Lagrangian method. The analysis builds on the classic local convergence of ALM [5], and our purpose is to provide some quantitative justification for the fast convergence of the two-level algorithm, which will be presented in Section 6.

To begin with, we note that the inner-level problem (10) solved by ADMM is closely related to the problem

$$\min_{x\in\mathcal{X},\bar{x}\in\bar{\mathcal{X}}} f(x) - \langle\lambda^k, Ax + B\bar{x}\rangle + \frac{\beta^k}{2}\|Ax + B\bar{x}\|^2. \tag{21}$$

It is straightforward to verify that $(x^k, \bar{x}^k)$ is a stationary point of (21) in the sense that

$$0 \in \nabla f(x^k) + A^\top(-\lambda^k + \beta^k(Ax^k + B\bar{x}^k)) + N_{\mathcal{X}}(x^k), \tag{22a}$$
$$0 \in B^\top(-\lambda^k + \beta^k(Ax^k + B\bar{x}^k)) + N_{\bar{\mathcal{X}}}(\bar{x}^k), \tag{22b}$$

if and only if $(x^k, \bar{x}^k, z^k, y^k)$ is a stationary point of (10) satisfying (11) with $z^k = -Ax^k - B\bar{x}^k$ and $y^k = -\lambda^k + \beta^k(Ax^k + B\bar{x}^k)$. In addition, an approximate stationary solution of (10) can be mapped to an approximate solution of (21).

Lemma 1 *Let $(x^k, \bar{x}^k, z^k, y^k)$ be a (d_1^k, d_2^k, d_3^k)-stationary point of (10) in the sense of (13). Then $(x^k, \bar{x}^k)$ is a $(\tilde{d}_i^k, \tilde{d}_2^k)$-stationary point of (21), i.e.,*

$$\tilde{d}_1^k \in \nabla f(x^k) + A^\top(-\lambda^k + \beta^k(Ax^k + B\bar{x}^k)) + N_{\mathcal{X}}(x^k), \tag{23a}$$
$$\tilde{d}_2^k \in B^\top(-\lambda^k + \beta^k(Ax^k + B\bar{x}^k)) + N_{\bar{\mathcal{X}}}(\bar{x}^k), \tag{23b}$$

where $\tilde{d}_1^k = d_1^k + \beta^k A^\top d_3^k$, and $\tilde{d}_2^k = d_2^k + \beta^k B^\top d_3^k$.

Proof By (13c) and (13d), we have $y^k = -\lambda^k + \beta^k(Ax^k + Bz^k - d_3^k)$; plugging this equality into (13a)-(13b) yields the result. □

Thus we will mainly focus on problem (21) and its approximate stationarity system (23) in this section. We add following assumptions on problem (3).

Assumption 5 *The set $\mathcal{X} = \{x \in \mathbb{R}^{n_1} : h(x) = 0\}$ is compact with $h : \mathbb{R}^{n_1} \to \mathbb{R}^p$ being second-order continuously differentiable, the objective f is second-order continuously differentiable over some open set containing $\mathcal{X}$, and $\bar{\mathcal{X}}$ is a convex set with nonempty interior in R^{n_2}. The matrix B has full column rank.*

Remark 2 Any inequality constraint in $\mathcal{X}$ can be converted to the form $h(x) = 0$ by adding the squares of additional slack variables. The second-order continuous differentiability of f and h are standard to establish local convergence of the augmented Lagrangian method. In addition, we explicitly require B to have full column rank, which can be justified by the reformulation (2).

Definition 1 Let $x^* \in \mathcal{X} = \{x | h(x) = 0\}$ and $\nabla h(x^*) = [\nabla h_1(x^*), \cdots, \nabla h_p(x^*)] \in \mathbb{R}^{n_1 \times p}$.

1. The tangent cone of $\mathcal{X}$ at x^*:

$$T_{\mathcal{X}}(x^*) = \left\{ d \in \mathbb{R}^{n_1} \mid \exists x^k \in \mathcal{X}, x^k \to x^*, \frac{x^k - x^*}{\|x^k - x^*\|} \to \frac{d}{\|d\|} \right\}.$$

2. The cone of the first-order feasible variation of $\mathcal{X}$ at x^*:

$$V_{\mathcal{X}}(x^*) = \{d \in \mathbb{R}^{n_1} : \nabla h(x^*)^\top d = 0\}.$$

3. We say that x^* is quasiregular if $T_{\mathcal{X}}(x^*) = V_{\mathcal{X}}(x^*)$.

Assumption 6 *Problem* (3) *has a feasible solution* $(x^*, \bar{x}^*)$*, where* $\bar{x}^* \in \operatorname{Int} \bar{\mathcal{X}}$ *and all equality constraints have linearly independent gradient vectors. In addition,* $(x^*, \bar{x}^*)$*, together with some dual multipliers* $\lambda^* \in (\underline{\lambda}, \overline{\lambda})$ *and* $\mu^* \in \mathbb{R}^p$*, satisfy*

$$\nabla f(x^*) - A^\top \lambda^* + \nabla h(x^*)\mu^* = 0, \quad B^\top \lambda^* = 0, \tag{24a}$$

$$u^\top \left(\nabla^2 f(x^*) + \sum_{i=1}^{p} \mu_i^* \nabla^2 h_i(x^*) \right) u > 0,$$
$$\forall (u, v) \neq 0 \text{ s.t. } Au + Bv = 0,\ \nabla h(x^*)^\top u = 0. \tag{24b}$$

Moreover, there exists $R > 0$ *such that* x *is quasiregular for all* $x \in B_R(x^*) \cap \mathcal{X}$.

Remark 3 Assumption 6 can be regarded as a second-order sufficient condition at a local minimizer $(x^*, \bar{x}^*)$ of problem (3), and B having full column rank is necessary for (24b) to hold. The quasiregularity assumption can be satisfied by a wide range of constraint qualifications.

The quasiregularity condition bridges the normal cone stationarity condition to the well-known KKT condition.

Proposition 2 *If* $x^k \in \mathcal{X}$ *is quasiregular,* $\bar{x}^k \in \operatorname{Int} \bar{\mathcal{X}}$*, and* $(x^k, \bar{x}^k)$ *satisfies condition* (23) *with some* $\tilde{d}_1^k$ *and* $\tilde{d}_2^k$*, then there exists some* $\mu^k \in \mathbb{R}^p$ *such that* $(x^k, \bar{x}^k)$ *satisfies the approximate KKT condition of problem* (21)*, i.e.,* $h(x^k) = 0$*,*

$$\tilde{d}_1^k = \nabla f(x^k) + A^\top(-\lambda^k + \beta^k(Ax^k + B\bar{x}^k)) + \nabla h(x^k)\mu^k, \tag{25a}$$

$$\tilde{d}_2^k = B^\top(-\lambda^k + \beta^k(Ax^k + B\bar{x}^k)). \tag{25b}$$

Proof The claim uses the fact that the normal cone $N_{\mathcal{X}}(x)$ is the polar cone of the tangent cone $T_{\mathcal{X}}(x)$, and $N_{\bar{\mathcal{X}}}(\bar{x}) = \{0\}$ for $\bar{x} \in \operatorname{Int} \bar{X}$. The existence of μ^k follows from the Farkas' Lemma [3, Prop 4.3.12]. □

Proposition 3 *Suppose Assumption 5 holds, and let* $(x^*, \bar{x}^*, \mu^*, \lambda^*)$ *be defined as in Assumption 6. There exist positive* $\underline{\beta}$ *and* δ *such that for all* $s = (\lambda, \beta, \tilde{d}_1, \tilde{d}_2)$ *belonging to the set*

$$S := \left\{ s = (\lambda, \beta, \tilde{d}_1, \tilde{d}_2) \mid \left(\frac{\|\lambda - \lambda^*\|^2}{\beta^2} + \|\tilde{d}_1\|^2 + \|\tilde{d}_2\|^2 \right)^{1/2} \leq \delta, \beta \geq \underline{\beta} \right\},$$

there exist unique continuously differentiable mappings $x(s)$, $\bar{x}(s)$, $\mu(s)$, *and* $\tilde{\lambda}(s) = \lambda - \beta[Ax(s) + B\bar{x}(s)]$ *defined in the interior of* S *satisfying*

$$\nabla f[x(s)] - A^\top \tilde{\lambda}(s) + \nabla h[x(s)]\mu(s) = \tilde{d}_1,\ B^\top \tilde{\lambda}(s) = \tilde{d}_2,\ h[x(s)] = 0; \tag{26}$$

$$\Big(x(\lambda^*, \beta, 0, 0), \bar{x}(\lambda^*, \beta, 0, 0), \mu(\lambda^*, \beta, 0, 0), \tilde{\lambda}(\lambda^*, \beta, 0, 0)\Big) = (x^*, \bar{x}^*, \mu^*, \lambda^*); \tag{27}$$

$$\bar{x}(s) \in \text{Int } \bar{\mathcal{X}}, \|x(s) - x^*\| \leq R. \tag{28}$$

Moreover, there exists $M > 0$ *such that for any* $s \in S$, *we have*

$$\begin{aligned} &\max\{\|x(s) - x^*\|, \|\bar{x}(s) - \bar{x}^*\|, \|\tilde{\lambda}(s) - \lambda^*\|\} \\ \leq &M(\|\lambda - \lambda^*\|^2/\beta^2 + \|\tilde{d}_1\|^2 + \|\tilde{d}_2\|^2)^{1/2}. \end{aligned} \tag{29}$$

Proof See Appendix B.1. □

Proposition 4 *Suppose Assumptions 5 and 6 hold. Let* M *and* S *be defined as in Proposition 3. Suppose for some* (β^k, λ^k) *with* $\beta^k \geq M$, *ADMM finds a* (d_1^k, d_2^k, d_3^k)*-stationary solution* $(x^k, \bar{x}^k, z^k, y^k)$ *satisfying* (13) *such that*

1. $s^k = (\lambda^k, \beta^k, \tilde{d}_1^k, \tilde{d}_2^k) \in S$, *where* $\tilde{d}_1^k = d_1^k + \beta^k A^\top d_3^k$, *and* $\tilde{d}_2^k = d_2^k + \beta^k B^\top d_3^k$;
2. $(x^k, \bar{x}^k) = (x(s^k), \bar{x}(s^k))$;
3. *there exists a positive constant* $\eta < \beta^k/M$ *such that*

$$\left(\|A\| + \|B\| + \frac{1}{M} \right) (\|d_1^k\| + \|d_2^k\| + \|d_3^k\|) \leq \frac{\eta}{\beta^k} \|Ax^k + B\bar{x}^k\|. \tag{30}$$

Denote $\hat{\lambda}^k := \lambda^k + \beta^k z^k$. *Then we have*

$$\|\hat{\lambda}^k - \lambda^*\| \leq \left(\frac{M}{\beta^k} + \frac{M\eta(M + \beta^k)}{\beta^k(\beta^k - M\eta)} \right) \|\lambda^k - \lambda^*\|. \tag{31}$$

Proof See Appendix B.2. □

Theorem 5 *Suppose Assumptions 5 and 6 hold. Let* $\underline{\beta}$, δ, M, *and* S *be defined as in Proposition 3. Suppose the three conditions in Proposition 4 are satisfied for all iterates* $k \in \mathbb{Z}_+$, *and the initial penalty* $\beta^0 > \frac{M}{\varrho}(1 + \eta + \varrho\eta)$ *for some* $\varrho \in (0, 1)$. *Then the following results hold:*

1. *the sequence* $\{\lambda^k\}_{k \in \mathbb{Z}_{++}}$ *stays inside the interior of* $[\underline{\lambda}, \overline{\lambda}]$, *i.e.,* $\lambda^{k+1} = \hat{\lambda}^k = \lambda^k + \beta^k z^k$;

2. the dual variable λ^k converges to λ^* with at least a linear rate i.e.,

$$\lim_{k\to+\infty}\frac{\|\lambda^{k+1}-\lambda^*\|}{\|\lambda^k-\lambda^*\|}\le\varrho<1,\ \textit{and}\ \lim_{k\to+\infty}\frac{\|\lambda^{k+1}-\lambda^*\|}{\|\lambda^k-\lambda^*\|}=0\ \textit{if}\ \beta^k\to+\infty;$$

3. $\max\{\|x^k-x^*\|,\|\bar{x}^k-\bar{x}^*\|\}\le\varrho\|\lambda^k-\lambda^*\|\le\varrho^{k+1}\|\lambda^0-\lambda^*\|$.

Proof The coefficient in the right-hand side of (31) is less than ϱ if $\beta^k>\frac{M}{\varrho}(1+\eta+\varrho\eta)$, and converges to 0 if $\beta^k\to+\infty$; thus the first two parts of the theorem are proved. Part 3 is due to (29) and the same derivation as in Proposition 4. □

Theorem 5 suggests that if we have a good initial point (inside the set S defined in Proposition 3) and each inner ADMM locates the approximate stationary solution specified by the implicit function theorem (as in Proposition 4), then the two-level algorithm exhibits local linear or super-linear convergence in its outer level. The results are consistent with our empirical observations to be presented in Section 6, where usually only a few outer-level updates are required upon convergence.

6 Examples

We present some applications of the two-level algorithm. All programs are coded using the Julia programming language 1.1.0 with JuMP package 0.18 [13] and implemented on a 64-bit laptop with one 2.6 GHz Intel Core i7 processor, 6 cores, and 16GB RAM. All nonlinear constrained problems are solved by the interior point solver IPOPT (version 3.12.8) [64] with linear solver MA27.

6.1 Nonlinear Network Flow Problem

We consider a specific class of network flow problems, which is covered by the motivating formulation (1). Suppose a connected graph $G(\mathcal{V},\mathcal{E})$ is given, where some nodes have demands of certain commodity and such demands need to be satisfied by some supply nodes. Each node i keeps local variables $[p_i;x_i;\{x_{ij}\}_{j\in\delta(i)};\{y_{ij}\}_{j\in\delta(i)}]\in\mathbb{R}^{2|\delta(i)|+2}$. Variable p_i is the production variable at node i, and (x_i,x_{ij},y_{ij}) determine the flow from node i to node j: $p_{ij}=g_{ij}(x_i,x_{ij},y_{ij})$ where $g_{ij}:\mathbb{R}^3\to\mathbb{R}$. For example, in an electric power network or a natural gas network, variables (x_i,x_{ij},y_{ij}) are usually related to electric voltages or gas pressures of local utilities. Moreover, for each $(i,j)\in\mathcal{E}$, nodal variables (x_i,x_j,x_{ij},y_{ij}) are coupled together in a nonlinear fashion: $h_{ij}(x_i,x_j,x_{ij},y_{ij})=0$ where $h_{ij}:\mathbb{R}^4\to\mathbb{R}$. As an analogy, this coupling

represents some physical laws on nodal potentials. We consider the problem

$$\min \quad \sum_{i\in\mathcal{V}} f_i(p_i) \tag{32a}$$

$$\text{s.t.} \quad p_i - d_i = \sum_{j\in\delta(i)} p_{ij} \quad \forall i \in \mathcal{V}, \tag{32b}$$

$$p_{ij} = g_{ij}(x_i, x_{ij}, y_{ij}) \quad \forall (i,j) \in \mathcal{E}, \tag{32c}$$

$$h_{ij}(x_i, x_j, x_{ij}, y_{ij}) = 0 \quad \forall (i,j) \in \mathcal{E}, \tag{32d}$$

$$x_i \in [\underline{x}_i, \overline{x}_i] \quad \forall i \in \mathcal{V}. \tag{32e}$$

In (32), the generation cost of each node, denoted by $f_i(\cdot)$, is a function of its production level p_i. The goal is to minimize total generation cost over the network. Each node is associated with a demand d_i and has to satisfy the injection balance constraint (32b); nodal variable x_i is bounded in $[\underline{x}_i, \overline{x}_i]$. Formulation (32) covers a wide range of problems and can be categorized into the GNF problem studied in [60]. Suppose the network is partitioned into a few subregions, and (i,j) is an edge crossing two subregions with i (resp. j) in region 1 (resp. 2). In order to facilitate parallel implementation, we replace constraint (32d) by the following constraints with additional variables:

$$h_{ij}(x_i^1, x_j^1, x_{ij}, y_{ij}) = 0,\ h_{ji}(x_j^2, x_i^2, x_{ji}, y_{ji}) = 0, \tag{33a}$$

$$x_i^1 = \bar{x}_i,\ x_i^2 = \bar{x}_i,\ x_j^1 = \bar{x}_j,\ x_j^2 = \bar{x}_j; \tag{33b}$$

similarly, we replace p_{ij} and p_{ji} in (32c) by

$$p_{ij} = g_{ij}(x_i^1, x_{ij}, y_{ij}),\ p_{ji} = g_{ji}(x_j^2, x_{ji}, y_{ji}). \tag{34}$$

Notice that $(x_i^1, x_j^1, x_{ij}, y_{ij})$ are controlled by region 1 and $(x_i^2, x_j^2, x_{ji}, y_{ji})$ are controlled by region 2. After incorporating constraints (33)-(34) for all crossing edges (i,j) into problem (32), the resulting problem is in the form of (3) and ready for our two-level algorithm. We consider the case where coupling constraints are given by $p_{ij} = \frac{a_i}{|\delta(i)|}x_i + b_{ij}x_{ij} + c_{ij}y_{ij}$ and $h_{ij}(x_i, x_j, x_{ij}, y_{ij}) = x_{ij}^2 + y_{ij}^2 - x_i x_j$. Constraint (32c) is linear with parameters (a_i, b_{ij}, c_{ij}), while the nonconvex constraint (32d) restricts $(x_i, x_j, x_{ij}, y_{ij})$ on the surface of a rotated second-order cone.

We use the underlying network topology from [75] to generate our testing networks. Each network is partitioned into two, three, or four subregions. The graph information and centralized objectives from IPOPT are recorded in the first three columns of Table 2. The column "LB" records the objective value by relaxing the constraint (32d) to $h_{ij}(x_i, x_j, x_{ij}, y_{ij}) \le 0$. It is clear that this relaxation makes problem (32) convex and provides a lower bound to the global optimal value. Partition information are given in the last two columns. We compare our algorithm with PDD in [58] as well as the proximal ADMM-g proposed in [32] (which solves problem (5) instead). We set an absolute tolerance $\epsilon = 1.0e-5$, and initialize $(x_i, x_j, x_{ij}, y_{ij})$ with $(1,1,1,0)$ and p_i with the initial value provided in [75]. For our two-level algorithm, we choose

Table 2: Network information

$\|\mathcal{V}\|$	$\|\mathcal{E}\|$	Central Obj.	LB	Idx	Partition Size	# cross edges
14	20	53.67	53.67	14-2	5+9	3
				14-3	4+5+5	5
				14-4	2+4+4+4	7
118	179	862.09	862.03	118-2	47+71	4
				118-3	35+35+48	7
				118-4	20+28+34+36	12
300	409	4751.31	4751.20	300-2	111+189	4
				300-3	80+87+133	7
				300-4	58+64+88+90	11
1354	1710	740.09	740.02	1354-2	455+899	11
				1345-3	340+455+559	18
				1354-4	236+303+386+429	25

$\omega = 0.75$, $\gamma = 1.5$, and $\beta^1 = 1000$. Each component of λ is restricted between $\pm 10^6$. The stopping criteria (14) suggests that ϵ_1^k and ϵ_2^k should be of the order $\mathcal{O}(\rho^k \epsilon_3^k)$. Motivated by this observation, we terminate the inner-level ADMM when $\|Ax^t + B\bar{x}^t + z^t\| \leq \max\{\epsilon, \sqrt{m}/(k \cdot \rho^k)\}$, where m is the dimension of the vector, and ρ^k is the inner ADMM penalty at outer iteration k. For PDD, as suggested in [58, Section V.B], we terminate the inner-level of PDD when the relative gap of two consecutive augmented Lagrangian values is less than $\max\{\epsilon, 100\epsilon \times (2/3)^k\}$; at the end of each inner-level rBSUM, the primal feasibility is checked and penalty is updated with the same ω and γ. Notice that the parameters used in the proposed algorithm and PDD are matched in our experiments. For proximal ADMM-g, we choose $\beta = 1/\epsilon^2$ and $\rho = 3/\epsilon^2$; additional proximal terms $\frac{1}{2}\|x - x^t\|_H^2$ and $\frac{1}{2}\|\bar{x} - \bar{x}^t\|_H^2$ are added to the subproblem update, where $H = \frac{0.01}{\epsilon} I$. All three algorithms terminate if $\|Ax^k + B\bar{x}^k\| \leq \sqrt{m} \times \epsilon$. Test results are presented in Table 3.

The number of outer-level updates (ALM multiplier updates for PDD and the two-level algorithm) and the total number of inner-level updates (rBSUM iterations for PDD and ADMM iterations for the two-level algorithm) are reported in columns "Outer" and "Inner", respectively. We see that both the proposed algorithm and PDD converge in all test cases, and both of them take around 10-30 outer-level iterations to drive the constraint violation "$\|Ax + B\bar{x}\|$" close to zero. PDD converges fast for three cases of network 300; however, for most cases it requires more total inner and outer iterations for convergence than the proposed algorithm. Such performance is consistent with the analysis in [58], where the inner-level rBSUM algorithm needs to run long enough to guarantee each block variable achieves stationarity. The objective values and duality gaps of solutions generated by the three algorithms are recorded in "Obj" and "Gap (%)". We can see both the proposed algorithm and PDD are able to achieve near global optimality, while the proposed algorithm finds solutions with even higher quality than PDD at termination. The algorithm running time (model building time excluded) is recorded in the last

Table 3: Comparison with PDD [58], proximal ADMM-g [32]

Idx	Method	Outer	Inner	$\|Ax + B\bar{x}\|$	Obj	Gap (%)	Time (s)
	ADMM-g	-	42	3.35e-05	93.06	42.33	8.30
14-2	PDD	21	94	3.65e-05	53.96	0.53	2.01
	Proposed	10	54	3.77e-05	53.98	0.58	1.25
	ADMM-g	-	247	5.27e-05	72.86	26.34	6.54
14-3	PDD	22	188	3.88e-05	53.98	0.57	1.82
	Proposed	20	140	1.11e-05	53.99	0.60	1.40
	ADMM-g	-	259	5.90e-05	81.67	34.28	7.58
14-4	PDD	24	896	5.29e-05	54.72	1.91	9.41
	Proposed	19	250	7.69e-05	54.42	1.37	2.43
	ADMM-g	-	40	4.43e-05	1283.48	32.84	6.20
118-2	PDD	24	85	3.34e-05	870.20	0.94	3.75
	Proposed	15	100	3.16e-05	864.71	0.31	3.94
	ADMM-g	-	67	6.26e-05	1200.01	28.16	1.91
118-3	PDD	25	141	5.80e-05	867.44	0.62	2.95
	Proposed	11	86	5.16e-05	866.17	0.48	1.85
	ADMM-g	-	59	8.11e-05	1201.82	28.27	4.48
118-4	PDD	25	178	6.64e-05	868.68	0.77	3.59
	Proposed	14	137	6.50e-05	867.16	0.59	2.86
	ADMM-g	-	227	4.80e-05	5054.52	6.00	15.34
300-2	PDD	28	93	3.87e-05	4757.06	0.12	5.54
	Proposed	20	304	1.74e-05	4751.71	0.01	18.04
	ADMM-g	-	400	6.43e-05	5049.16	5.90	20.46
300-3	PDD	28	127	6.27e-05	4757.63	0.13	6.27
	Proposed	25	517	4.80e-05	4752.52	0.03	23.83
	ADMM-g	-	1000	1.67e-04	5041.50	5.76	46.41
300-4	PDD	28	243	7.37e-05	4765.06	0.29	10.28
	Proposed	20	512	7.26e-05	4752.56	0.03	19.53
	ADMM-g	-	901	7.86e-05	767.44	3.57	672.51
1354-2	PDD	25	299	6.56e-05	745.50	0.73	212.91
	Proposed	19	126	6.39e-05	743.32	0.44	84.61
	ADMM-g	-	1000	1.66e-04	771.52	4.08	342.77
1354-3	PDD	26	422	8.86e-05	747.90	1.05	174.78
	Proposed	18	137	7.04e-05	744.91	0.66	50.77
	ADMM-g	-	1000	4.90e-04	769.55	3.84	265.59
1354-4	PDD	27	838	1.10e-04	749.61	1.28	523.78
	Proposed	18	170	8.15e-05	744.98	0.67	115.71

column "Time (s)". We would like to emphasize that, under similar algorithmic settings, the proposed two-level algorithm in general converges faster and shows better scalability than the other two algorithms.

Even with sufficiently large penalty on the slack variable z, the proximal ADMM-g does not achieve the desired primal feasibility for cases 300-4, 1354-3, and 1354-4 in 1000 iterations; for other cases, it usually takes more time than the proposed algorithm. We point out that ADMM-g usually finds suboptimal solutions, and the duality gap can be as large as 42%. We believe this happens because problem (5) requires the introduction of large $\beta(\epsilon)$ and $\rho(\epsilon)$, which affect the structure of the original problem (3) and result in solutions with poor quality. Moreover, such large parameters also cause numerical issues

for the IPOPT solver and slow down the overall convergence, and this is the reason why ADMM-g takes a long time even when the number of iterations is relatively small for the first four test cases. We also tried a smaller penalty $\mathcal{O}(1/\epsilon)$, in which case the ADMM-g cannot achieve the desired feasibility level.

6.2 Minimization over Compact Manifold

We consider the following problem

$$\min \quad \sum_{i=1}^{n_p-1} \sum_{j=i+1}^{n_p} \left((x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2\right)^{-\frac{1}{2}} \tag{35a}$$

$$\text{s.t.} \quad x_i^2 + y_i^2 + z_i^2 = 1, \quad \forall i \in [n_p]. \tag{35b}$$

Problem (35) is obtained from the benchmark set COPS 3.0 [11] of nonlinear optimization problems. The same problem is used in [70] to test algorithms that preserve spherical constraints through curvilinear search. We compare solutions and computation time of our distributed algorithm with those obtained from the centralized IPOPT solver. Each test problem is firstly solved in a centralized way; objective value and total running time are recorded in the second and third column of Table 4. Using additional variables to break couplings in the objective (35a), we divide each test problem into three subproblems. Subproblems have the same number of variables, constraints, and objective terms (as in (35a)). For our two-level algorithm, we choose $\gamma = 2$, $\omega = 0.5$; initial value of penalty β^1 is set to 100 for $n_p \in \{60, 90\}$, 200 for $n_p \in \{120, 180\}$, and 500 for $n_p \in \{240, 300\}$. The initial point is set to $(x_i, y_i, z_i) = (0.2, 0.3, 0.1)$ for all $i \in [n_p]$ for IPOPT. We set bounds on each component of λ to be $\pm 10^6$. The inner-level ADMM terminates when $\|Ax^t + B\bar{x}^t + z^t\| \leq \sqrt{3n_p}/(2500k)$, where k is the current outer-level index; the outer level terminates when $\|Ax^k + B\bar{x}^k\| \leq \sqrt{3n_p} \times 1.0e-6$.

The quality of the centralized solution is slightly better than distributed solutions, while our proposed algorithm is able to reduce the running time significantly except for one case ($n_p = 90$) while ensuring feasibility. In addition, as indicated in Table 4, numbers of iterations for both inner and outer levels stay stable across all test cases, which suggests that the proposed algorithm scales well with the size of the problem. In view of the discussion in Section 4.2, we compare with the penalty method, where $\lambda^k = 0$ for all k, to demonstrate the effect of the outer-level dual variable. Without updating λ, the penalty method requires more inner/outer updates and substantially longer time.

6.3 A Multi-block Problem: Robust Tensor PCA

In this section, we use the robust tensor PCA problem considered in [32] to illustrate that the two-level framework can be generalized to multi-block problem (4), and when Conditions 1 and 2 are satisfied, the resulting two-level

Table 4: Comparison of centralized and distributed solutions

	Centralized Ipopt		Proposed two-level algorithm and penalty method					
n_p	Obj.	Time (s)	Method	Outer	Inner	$\|Ax + B\bar{x}\|$	Gap (%)	Time (s)
60	1543.83	9.55	Proposed	11	62	1.17e-05	0.79	4.17
			Penalty	18	102	1.32e-05	0.54	7.82
90	3579.18	17.34	Proposed	12	98	1.01e-05	0.14	20.42
			Penalty	18	136	9.62e-06	0.13	26.97
120	6474.77	56.64	Proposed	12	79	8.77e-06	0.30	45.28
			Penalty	17	113	1.75e-05	0.21	60.42
180	14867.41	212.95	Proposed	12	82	1.71e-05	0.10	173.81
			Penalty	18	121	1.69e-05	0.09	233.75
240	26747.84	710.68	Proposed	12	79	1.25e-05	0.44	417.62
			Penalty	17	111	2.02e-05	0.28	534.59
300	42131.88	1568.64	Proposed	12	80	1.51e-05	0.17	852.19
			Penalty	18	115	2.94e-05	0.12	1094.91

algorithm can potentially accelerate one-level ADMM. In particular, given an estimate R of the CP-rank, the problem of interest is casted as

$$\min_{A,B,C,\mathcal{Z},\mathcal{E},\mathcal{B}} \|\mathcal{Z} - [\![A, B, C]\!]\|^2 + \alpha\|\mathcal{E}\|_1 + \alpha_N\|\mathcal{B}\|_F^2 \quad \text{s.t.} \;\; \mathcal{E} + \mathcal{Z} + \mathcal{B} = \mathcal{T}, \tag{36}$$

where $A \in \mathbb{R}^{I_1 \times R}, B \in \mathbb{R}^{I_2 \times R}, C \in \mathbb{R}^{I_3 \times R}$, and $[\![A, B, C]\!]$ denotes the sum of column-wise outer product of A, B, and C. We denote the mode-i unfolding of tensor $\mathcal{Z}$ by $Z_{(i)}$, the Khatri-Rao product of matrices by $\odot$, the Hadamard product by $\circ$, and the soft shrinkage operator by $\mathbf{S}$. We implement the two-level framework as in Algorithm 5. We firstly perform ADMM-g in steps 3-10. We note that there are some modifications to the ADMM-g described in [32]: since our two-level framework requires the introduction of an additional slack variable $\mathcal{S}$, steps 6-8 have an additional term $S_{(1)}^k$, and $S_{(1)}^{k+1}$ is then updated in step 9 via a gradient step as in ADMM-g; moreover, during the update of $\mathcal{B}$, we also add a proximal term with coefficients $\delta_6/2$. When the residual $\|\mathcal{Z}^{k+1} + \mathcal{E}^{k+1} + \mathcal{B}^{k+1} + \mathcal{S}^{k+1} - \mathcal{T}\|_F$ is small enough, which can serve as an indicator of the convergence of ADMM-g, we multiply the penalty β by some γ as long as $\beta < 1.0e+6$, and update the outer-level dual variable Λ as in step 12, where the projection step is omitted.

We experiment on tensors with dimensions $I_1 = 30$, $I_2 = 50$, and $I_3 = 70$, which match the largest instances tested by [32]; the initial estimation R is given by $R_{CP} + \lceil 0.2 * R_{CP} \rceil$. In our implementation, we set $\gamma = 1.5$, $c = 3$, and the initial β is set to 2; the inner-level ADMM-g terminates if the residual $\|\mathcal{Z}^{k+1} + \mathcal{E}^{k+1} + \mathcal{B}^{k+1} + \mathcal{S}^{k+1} - \mathcal{T}\|_F$ is less than $\max\{1e-5, 1e-3/K_{\text{out}}\}$, where K_{out} is the current outer-level iteration count. All other parameters, generation of problem data, and initialization follow the description in [32, Section 5]. For each value of the CP rank, we generate 10 cases and let ADMM-g and the proposed two-level Algorithm perform 2000 (inner) iterations. We calculate the

Algorithm 5 : Two-level Algorithm for Robust Tensor PCA

Initialize primal variables $A^0, B^0, C^0, \mathcal{E}^0, \mathcal{Z}^0, \mathcal{B}^0$, $\mathcal{S}^0$, dual variables Y^0, Λ^0, penalty parameters β, $\rho = c\beta$, stepsize $\tau = \frac{1}{\rho}$, constants $\delta_i > 0$ for $i \in [6]$, $\gamma > 1$;
for $k = 0, 1, 2, \cdots$ **do**
  $A^{k+1} = [(Z)^k_{(1)}(C^k \odot B^k) + \frac{\delta_1}{2}A^k][((C^k)^\top C^k) \circ ((B^k)^\top B^k) + \frac{\delta_1}{2}I_R]^{-1}$;
  $B^{k+1} = [(Z)^k_{(2)}(C^k \odot A^k) + \frac{\delta_2}{2}B^k][((C^k)^\top C^k) \circ ((A^k)^\top A^k) + \frac{\delta_2}{2}I_R]^{-1}$;
  $C^{k+1} = [(Z)^k_{(3)}(B^k \odot A^k) + \frac{\delta_2}{2}C^k][((B^k)^\top B^k) \circ ((C^k)^\top C^k) + \frac{\delta_3}{2}I_R]^{-1}$;
  $E^{k+1}_{(1)} = \mathbf{S}\left(\frac{\rho}{\rho+\delta_4}\left(T_{(1)} + \frac{1}{\rho}Y^k_{(1)} - B^k_{(1)} - Z^k_{(1)} - S^k_{(1)}\right) + \frac{\delta_4}{\rho+\delta_4}E^k_{(1)}, \frac{\alpha}{\rho+\delta_4}\right)$;
  $Z^{k+1}_{(1)} = \frac{1}{2+2\delta_5+\rho}\left(2A^{k+1}\left(C^{k+1} \odot B^{k+1}\right)^\top + 2\delta_5 Z^k_{(1)} + \Lambda^k_{(1)} - \rho\left(E^{k+1}_{(1)} + B^k_{(1)} + S^k_{(1)} - T_{(1)}\right)\right)$;
  $B^{k+1}_{(1)} = \frac{1}{2\alpha_N+\delta_6+\rho}(Y^k_{(1)} + \delta_6 B^k_{(1)} - \rho\left(E^{k+1}_{(1)} + Z^{k+1}_{(1)} + S^k_{(1)} - T_{(1)})\right)$;
  $S^{k+1}_{(1)} = S^k_{(1)} - \tau(-Y^k_{(1)} + \Lambda_{(1)} + \rho(Z^{k+1}_{(1)} + E^{k+1}_{(1)} + B^{k+1}_{(1)} + S^k_{(1)} - T_{(1)}))$;
  $Y^{k+1}_{(1)} = Y^k_{(1)} - \rho\left(Z^{k+1}_{(1)} + E^{k+1}_{(1)} + B^{k+1}_{(1)} + S^{k+1}_{(1)} - T_{(1)}\right)$;
  if $\|\mathcal{Z}^{k+1} + \mathcal{E}^{k+1} + \mathcal{B}^{k+1} + \mathcal{S}^{k+1} - \mathcal{T}\|_F$ is smaller than some threshold **then**
    $\Lambda \leftarrow \Lambda + \beta\mathcal{S}^{k+1}$, $\beta \leftarrow \gamma\beta$, $\rho \leftarrow c\beta$, $\tau \leftarrow 1/\rho$;
  end if
end for

geometric mean r_k^{Geo} of the primal residuals $r_k = \|\mathcal{Z}^k + \mathcal{E}^k + \mathcal{B}^k - \mathcal{T}\|_F$ over 10 cases, and plot $\lg r_k^{\text{Geo}}$ as a function of iteration count k in Figure 1. We also calculate the geometric mean e_k^{Geo} of relative errors $\|\mathcal{Z}^k - \mathcal{Z}_{\text{true}}\|_F / \|\mathcal{Z}_{\text{true}}\|_F$ over 10 cases, where $\mathcal{Z}_{\text{true}}$ is the generated true low-rank tensor, and plot e_k^{Geo} in Figure 2. For our two-level algorithm, the primal residual decreases relatively slow during the first few inner ADMM-g; however, as we update the outer-level dual variable Λ and penalty β, r_k^{Geo} drops significantly faster than that of ADMM-g, and achieves feasibility with high precision in around 500 inner iterations. The relative error r_k^{Geo} of the two-level algorithm converges slightly slower than ADMM-g, while it is able to catch up and obtain the same level of optimality. The result suggests that our proposed two-level algorithm not only ensures convergence for a wider range of applications where ADMM may fail, but also accelerates ADMM on problems where convergence is already guaranteed.

7 Conclusion

This paper proposes a two-level distributed algorithm to solve the nonconvex constrained optimization problem (3). We identify some limitation of the standard ADMM algorithm, which in general cannot guarantee convergence when parallelization of constrained subproblems is considered. In order to overcome such difficulties, we propose a novel while concise distributed reformulation, which enables us to separate the underlying complication into two levels. The inner level utilizes multi-block ADMM to facilitate parallel implementation while the outer level uses the classic ALM to guarantee convergence to feasible solutions. Global convergence, local convergence, and iteration complexity of the proposed two-level algorithm are established, and we certify the possibil-

Fig. 1: Comparison of Infeasibility $\lg r_k^{\mathrm{Geo}}$

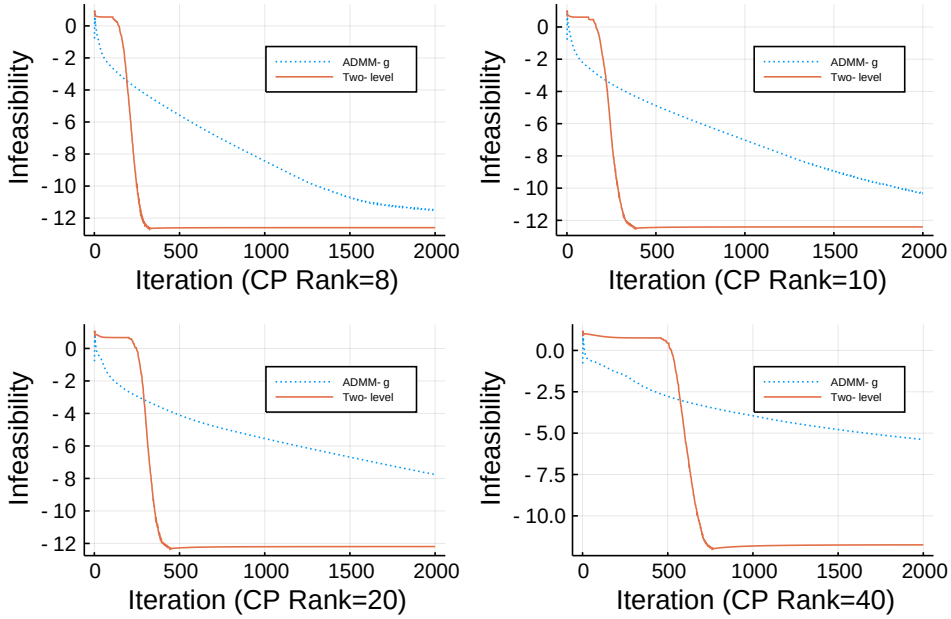
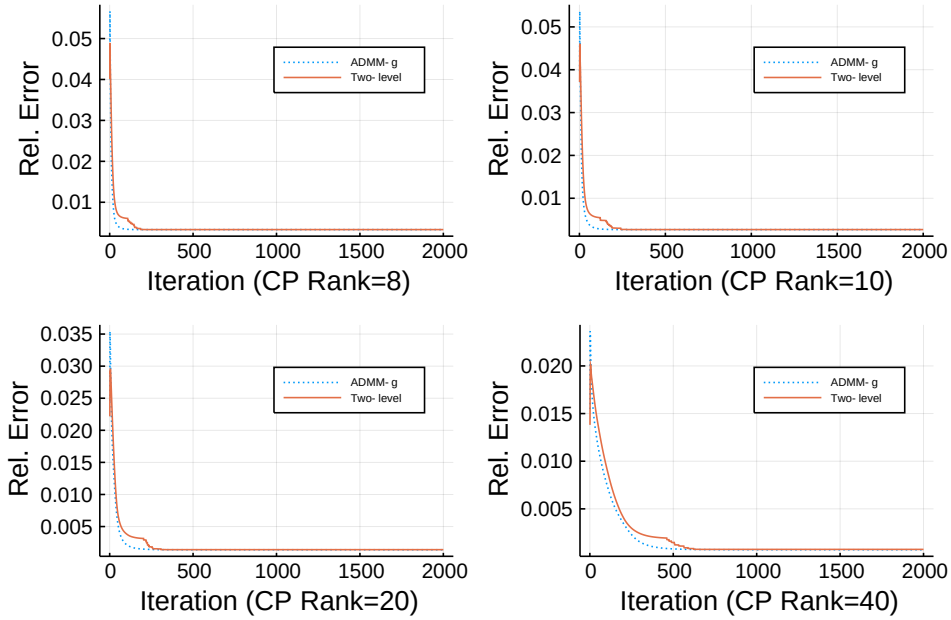


Fig. 2: Comparison of Relative Errors e_k^{Geo}



ity to extend the underlying algorithmic framework to solve more complicated nonconvex multi-block problems (4). In comparison to the other existing algorithms that are capable of solving the same class of nonconvex constrained programs, the proposed algorithm exhibits its advantages in terms of speed, scalability, and robustness. Thus for general nonconvex constrained multi-block problems, the two-level algorithm can serve an alternative to the workaround proposed in [32] when Condition 1 or 2 fails, and potentially accelerate ADMM on problems where slow convergence is frequently encountered.

References

1. Andreani, R., Birgin, E.G., Martínez, J.M., Schuverdt, M.L.: On augmented lagrangian methods with general lower-level constraints. SIAM Journal on Optimization **18**(4), 1286–1309 (2007)
2. Aus Ozdaglar, A. Makhdoumi: Distributed Multiagent Optimization: Linear Convergence Rate of ADMM (2015)
3. Bertsekas, D.P.: Nonlinear programming
4. Bertsekas, D.P.: Convergence rate of penalty and multiplier methods. In: Decision and Control including the 12th Symposium on Adaptive Processes, 1973 IEEE Conference on, vol. 12, pp. 260–264. IEEE (1973)
5. Bertsekas, D.P.: Constrained optimization and Lagrange multiplier methods. Academic press (2014)
6. Bolte, J., Sabach, S., Teboulle, M.: Proximal alternating linearized minimization for nonconvex and nonsmooth problems. Mathematical Programming **146**(1-2), 459–494 (2014)
7. Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J., et al.: Distributed optimization and statistical learning via the alternating direction method of multipliers. Foundations and Trends® in Machine Learning **3**(1), 1–122 (2011)
8. Chen, C., He, B., Ye, Y., Yuan, X.: The direct extension of admm for multi-block convex minimization problems is not necessarily convergent. Mathematical Programming **155**(1-2), 57–79 (2016)
9. Chen, C., Shen, Y., You, Y.: On the convergence analysis of the alternating direction method of multipliers with three blocks. In: Abstract and Applied Analysis, vol. 2013. Hindawi (2013)
10. Davis, D., Yin, W.: A three-operator splitting scheme and its optimization applications. Set-valued and variational analysis **25**(4), 829–858 (2017)
11. Dolan, E.D., Moré, J.J., Munson, T.S.: Benchmarking optimization software with cops 3.0. Tech. rep., Argonne National Lab., Argonne, IL (US) (2004)
12. Douglas, J., Rachford, H.H.: On the numerical solution of heat conduction problems in two and three space variables. Transactions of the American mathematical Society **82**(2), 421–439 (1956)
13. Dunning, I., Huchette, J., Lubin, M.: Jump: A modeling language for mathematical optimization. SIAM Review **59**(2), 295–320 (2017). DOI 10.1137/15M1020575
14. D'Ambrosio, C., Lodi, A., Wiese, S., Bragalli, C.: Mathematical programming techniques in water network optimization. European Journal of Operational Research **243**(3), 774–788 (2015)
15. Eckstein, J., Bertsekas, D.P.: On the douglas—rachford splitting method and the proximal point algorithm for maximal monotone operators. Mathematical Programming **55**(1-3), 293–318 (1992)
16. Erseghe, T.: Distributed optimal power flow using admm. IEEE Transactions on Power Systems **29**(5), 2370–2380 (2014)
17. Gabay, D.: Applications of the method of multipliers to variational inequalities, in,(1983), 299. doi: 10.1016. S0168-2024 (08) pp. 70034–1

18. Gabay, D., Mercier, B.: A dual algorithm for the solution of nonlinear variational problems via finite element approximation. Computers & Mathematics with Applications **2**(1), 17–40 (1976)
19. Glowinski, R., Marroco, A.: Sur l'approximation, par éléments finis d'ordre un, et la résolution, par pénalisation-dualité d'une classe de problèmes de dirichlet non linéaires. Revue française d'automatique, informatique, recherche opérationnelle. Analyse numérique **9**(R2), 41–76 (1975)
20. Gonçalves, M.L., Melo, J.G., Monteiro, R.D.: Extending the ergodic convergence rate of the proximal admm. arXiv preprint arXiv:1611.02903 (2016)
21. Gonçalves, M.L., Melo, J.G., Monteiro, R.D.: Convergence rate bounds for a proximal admm with over-relaxation stepsize parameter for solving nonconvex linearly constrained problems. arXiv preprint arXiv:1702.01850 (2017)
22. Guo, K., Han, D., Wu, T.T.: Convergence of alternating direction method for minimizing sum of two nonconvex functions with linear constraints. International Journal of Computer Mathematics **94**(8), 1653–1669 (2017)
23. Han, D., Yuan, X.: A note on the alternating direction method of multipliers. Journal of Optimization Theory and Applications **155**(1), 227–238 (2012)
24. He, B., Tao, M., Yuan, X.: Alternating Direction Method with Gaussian Back Substitution for Separable Convex Programming. SIAM Journal on Optimization **22**(2), 313–340 (2012). DOI 10.1137/110822347. URL http://epubs.siam.org/doi/10.1137/110822347
25. He, B., Tao, M., Yuan, X.: Convergence rate and iteration complexity on the alternating direction method of multipliers with a substitution procedure for separable convex programming. Math. Oper. Res., under revision **2**, 000–000 (2012)
26. He, B., Yuan, X.: On the o(1/n) convergence rate of the douglas–rachford alternating direction method. SIAM Journal on Numerical Analysis **50**(2), 700–709 (2012)
27. He, B., Yuan, X.: On non-ergodic convergence rate of douglas–rachford alternating direction method of multipliers. Numerische Mathematik **130**(3), 567–577 (2015)
28. Hestenes, M.R.: Multiplier and gradient methods. Journal of optimization theory and applications **4**(5), 303–320 (1969)
29. Hong, M.: Decomposing linearly constrained nonconvex problems by a proximal primal dual approach: Algorithms, convergence, and applications. arXiv preprint arXiv:1604.00543 (2016)
30. Hong, M., Luo, Z.Q.: On the linear convergence of the alternating direction method of multipliers. Mathematical Programming **162**(1-2), 165–199 (2017)
31. Hong, M., Luo, Z.Q., Razaviyayn, M.: Convergence analysis of alternating direction method of multipliers for a family of nonconvex problems. SIAM Journal on Optimization **26**(1), 337–364 (2016)
32. Jiang, B., Lin, T., Ma, S., Zhang, S.: Structured nonconvex and nonsmooth optimization: algorithms and iteration complexity analysis. Computational Optimization and Applications **72**(1), 115–157 (2019)
33. Jiang, B., Ma, S., Zhang, S.: Alternating direction method of multipliers for real and complex polynomial optimization models. Optimization **63**(6), 883–898 (2014)
34. Kocuk, B., Dey, S.S., Sun, X.A.: Strong socp relaxations for the optimal power flow problem. Operations Research **64**(6), 1177–1196 (2016)
35. Lan, G., Yang, Y.: Accelerated stochastic algorithms for nonconvex finite-sum and multiblock optimization. arXiv preprint arXiv:1805.05411 (2018)
36. Lan, G., Zhou, Y.: Random gradient extrapolation for distributed and stochastic optimization. SIAM Journal on Optimization **28**(4), 2753–2782 (2018)
37. Li, G., Pong, T.K.: Global convergence of splitting methods for nonconvex composite optimization. SIAM Journal on Optimization **25**(4), 2434–2460 (2015)
38. Li, M., Sun, D., Toh, K.C.: A convergent 3-block semi-proximal admm for convex minimization problems with one strongly convex block. Asia-Pacific Journal of Operational Research **32**(04), 1550024 (2015)
39. Lin, T., Ma, S., Zhang, S.: On the global linear convergence of the admm with multiblock variables. SIAM Journal on Optimization **25**(3), 1478–1497 (2015)
40. Lin, T., Ma, S., Zhang, S.: Iteration complexity analysis of multi-block admm for a family of convex minimization without strong convexity. Journal of Scientific Computing **69**(1), 52–81 (2016)

41. Lin, T., Ma, S., Zhang, S.: Global convergence of unmodified 3-block admm for a class of convex minimization problems. Journal of Scientific Computing **76**(1), 69–88 (2018)
42. Lin, T.Y., Ma, S.Q., Zhang, S.Z.: On the sublinear convergence rate of multi-block admm. Journal of the Operations Research Society of China **3**(3), 251–274 (2015)
43. Luo, H., Sun, X., Wu, H.: Convergence properties of augmented lagrangian methods for constrained global optimization. Optimisation Methods & Software **23**(5), 763–778 (2008)
44. Luo, Z.Q., Pang, J.S., Ralph, D., Wu, S.Q.: Exact penalization and stationarity conditions of mathematical programs with equilibrium constraints. Mathematical Programming **75**(1), 19–76 (1996)
45. Magnússon, S., Weeraddana, P.C., Fischione, C.: A distributed approach for the optimal power-flow problem based on admm and sequential convex approximations. IEEE Transactions on Control of Network Systems **2**(3), 238–253 (2015)
46. Makhdoumi, A., Ozdaglar, A.: Broadcast-based distributed alternating direction method of multipliers. In: 2014 52nd Annual Allerton Conference on Communication, Control, and Computing (Allerton), pp. 270–277. IEEE, Monticello, IL, USA (2014). DOI 10.1109/ALLERTON.2014.7028466. URL http://ieeexplore.ieee.org/document/7028466/
47. Makhdoumi, A., Ozdaglar, A.: Convergence Rate of Distributed ADMM over Networks. arXiv:1601.00194 [math] (2016). URL http://arxiv.org/abs/1601.00194. ArXiv: 1601.00194
48. Melo, J.G., Monteiro, R.D.: Iteration-complexity of a jacobi-type non-euclidean admm for multi-block linearly constrained nonconvex programs. arXiv preprint arXiv:1705.07229 (2017)
49. Melo, J.G., Monteiro, R.D.: Iteration-complexity of a linearized proximal multiblock admm class for linearly constrained nonconvex optimization problems. Available on: http://www. optimization-online. org (2017)
50. Monteiro, R.D., Svaiter, B.F.: Iteration-complexity of block-decomposition algorithms and the alternating direction method of multipliers. SIAM Journal on Optimization **23**(1), 475–507 (2013)
51. Peaceman, D.W., Rachford Jr, H.H.: The numerical solution of parabolic and elliptic differential equations. Journal of the Society for industrial and Applied Mathematics **3**(1), 28–41 (1955)
52. Pfetsch, M.E., Fügenschuh, A., Geißler, B., Geißler, N., Gollmer, R., Hiller, B., Humpola, J., Koch, T., Lehmann, T., Martin, A., et al.: Validation of nominations in gas network optimization: models, methods, and solutions. Optimization Methods and Software **30**(1), 15–53 (2015)
53. Powell, M.J.: " A method for non-linear constraints in minimization problems". UKAEA (1967)
54. Qi, L., Wei, Z.: On the constant positive linear dependence condition and its application to sqp methods. SIAM Journal on Optimization **10**(4), 963–981 (2000)
55. Rockafellar, R.T.: The multiplier method of hestenes and powell applied to convex programming. Journal of Optimization Theory and applications **12**(6), 555–562 (1973)
56. Rockafellar, R.T., Wets, R.J.B.: Variational analysis, vol. 317. Springer Science & Business Media (2009)
57. Shen, Y., Wen, Z., Zhang, Y.: Augmented lagrangian alternating direction method for matrix separation based on low-rank factorization. Optimization Methods and Software **29**(2), 239–263 (2014)
58. Shi, Q., Hong, M., Fu, X., Chang, T.H.: Penalty dual decomposition method for nonsmooth nonconvex optimization. arXiv preprint arXiv:1712.04767 (2017)
59. Shi, W., Ling, Q., Yuan, K., Wu, G., Yin, W.: On the Linear Convergence of the ADMM in Decentralized Consensus Optimization. IEEE Transactions on Signal Processing **62**(7), 1750–1761 (2014). DOI 10.1109/TSP.2014.2304432. URL http://ieeexplore.ieee.org/document/6731604/
60. Sojoudi, S., Fattahi, S., Lavaei, J.: Convexification of generalized network flow problem. Mathematical Programming pp. 1–39
61. Sun, A.X., Phan, D.T., Ghosh, S.: Fully decentralized ac optimal power flow algorithms. In: Power and Energy Society General Meeting (PES), 2013 IEEE, pp. 1–5. IEEE (2013)

62. Themelis, A., Patrinos, P.: Douglas-rachford splitting and admm for nonconvex optimization: tight convergence results (2018)
63. Tian, W., Yuan, X.: An alternating direction method of multipliers with a worst-case $o(1/n^2)$ convergence rate. Mathematics of Computation **88**(318), 1685–1713 (2019)
64. Wächter, A., Biegler, L.T.: On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. Mathematical programming **106**(1), 25–57 (2006)
65. Wang, F., Cao, W., Xu, Z.: Convergence of multi-block bregman admm for nonconvex composite problems. arXiv preprint arXiv:1505.03063 (2015)
66. Wang, F., Xu, Z., Xu, H.K.: Convergence of bregman alternating direction method with multipliers for nonconvex composite problems. arXiv preprint arXiv:1410.8625 (2014)
67. Wang, Y., Yin, W., Zeng, J.: Global convergence of admm in nonconvex nonsmooth optimization. Journal of Scientific Computing pp. 1–35 (2015)
68. Wei, E., Ozdaglar, A.: Distributed Alternating Direction Method of Multipliers. In: 2012 IEEE 51st IEEE Conference on Decision and Control (CDC), pp. 5445–5450. IEEE, Maui, HI, USA (2012). DOI 10.1109/CDC.2012.6425904. URL http://ieeexplore.ieee.org/document/6425904/
69. Wen, Z., Peng, X., Liu, X., Sun, X., Bai, X.: Asset allocation under the basel accord risk measures. arXiv preprint arXiv:1308.1321 (2013)
70. Wen, Z., Yin, W.: A feasible method for optimization with orthogonality constraints. Mathematical Programming **142**(1-2), 397–434 (2013)
71. Xu, Y., Yin, W.: A globally convergent algorithm for nonconvex optimization based on block coordinate update. Journal of Scientific Computing **72**(2), 700–734 (2017)
72. Xu, Y., Yin, W., Wen, Z., Zhang, Y.: An alternating direction algorithm for matrix completion with nonnegative factors. Frontiers of Mathematics in China **7**(2), 365–384 (2012)
73. Yang, L., Pong, T., Chen, X.: Alternating direction method of multipliers for nonconvex background/foreground extraction. arXiv preprint arXiv:1506.07029 (2015)
74. Zhang, R., Kwok, J.: Asynchronous distributed admm for consensus optimization. In: International Conference on Machine Learning, pp. 1701–1709 (2014)
75. Zimmerman, R.D., Murillo-Sánchez, C.E., Thomas, R.J., et al.: Matpower: Steady-state operations, planning, and analysis tools for power systems research and education. IEEE Transactions on power systems **26**(1), 12–19 (2011)

A Additional Proofs in Section 4

A.1 Proof of Proposition 1

We omit the index k in $(\rho^k, \beta^k, \lambda^k, T_k)$ occasionally. We first prove two lemmas.

Lemma 2 *For all $t \in \mathbb{Z}_{++}$, we have*

$$\langle B^\top y^{t-1} + \rho B^\top (Ax^t + B\bar{x}^t + z^{t-1}), \hat{x} - \bar{x}^t \rangle \geq 0 \quad \forall \hat{x} \in \bar{\mathcal{X}}, \tag{37}$$

$$\lambda + \beta z^t + y^t = 0. \tag{38}$$

Proof The claim follows from the optimality conditions of the $\bar{x}$ and z updates. □

Lemma 3 *Suppose Assumptions 2-3 hold, and we set $\rho = 2\beta$, then*

$$L_\rho(x^{t-1}, \bar{x}^{t-1}, z^{t-1}, y^{t-1}) - L_\rho(x^t, \bar{x}^t, z^t, y^t) \geq \beta \|B\bar{x}^{t-1} - B\bar{x}^t\|^2 + \beta \|z^{t-1} - z^t\|^2 \tag{39}$$

for all $t \in \mathbb{Z}_{++}$; in addition, there exists $\underline{L} \in \mathbb{R}$ independent of k such that for all $t \in \mathbb{Z}_+$,

$$L_\rho(x^t, \bar{x}^t, z^t, y^t) \geq \underline{L} > -\infty. \tag{40}$$

Proof We firstly show descent over x and $\bar{x}$ updates. By Assumption 3, we have

$$L_\rho(x^{t-1}, \bar{x}^{t-1}, z^{t-1}, y^{t-1}) \geq L_\rho(x^t, \bar{x}^{t-1}, z^{t-1}, y^{t-1}). \tag{41}$$

In addition, notice that

$$\begin{aligned}
&L_\rho(x^t, \bar{x}^{t-1}, z^{t-1}, y^{t-1}) - L_\rho(x^t, \bar{x}^t, z^{t-1}, y^{t-1}) \\
=&\langle y^{t-1}, B\bar{x}^{t-1} - B\bar{x}^t\rangle + \frac{\rho}{2}\|Ax^t + B\bar{x}^{t-1} + z^{t-1}\|^2 - \frac{\rho}{2}\|Ax^t + B\bar{x}^t + z^{t-1}\|^2 \\
=&\langle B^\top y^{t-1} + \rho B^\top(Ax^t + B\bar{x}^t + z^{t-1}), \bar{x}^{t-1} - \bar{x}^t\rangle + \frac{\rho}{2}\|B\bar{x}^{t-1} - B\bar{x}^t\|^2 \\
\geq&\frac{\rho}{2}\|B\bar{x}^{t-1} - B\bar{x}^t\|^2,
\end{aligned} \tag{42}$$

the second equality is due to $\|a+b\|^2 - \|a+c\|^2 = 2(a+c)^\top(b-c) + \|b-c\|^2$ with $a = Ax^t + z^{t-1}$, $b = B\bar{x}^{t-1}$, and $c = B\bar{x}^t$, and the last inequality is due to (37) of Lemma 2. Now we will show descent over z and y updates. Notice that if we define $h(z) = \lambda^\top z + \frac{\beta}{2}\|z\|^2$, then by Lemma 2, we have $\nabla h(z^t) = \lambda + \beta z^t = -y^t$; since $h(\cdot)$ is convex, it follows $h(z^{t-1}) - h(z^t) + (y^t)^\top(z^{t-1} - z^t) \geq 0$. Notice that

$$\begin{aligned}
&L_\rho(x^t, \bar{x}^t, z^{t-1}, y^{t-1}) - L_\rho(x^t, \bar{x}^t, z^t, y^t) \\
=&h(z^{t-1}) - h(z^t) + (y^t)^\top(z^{t-1} - z^t) + \frac{\rho}{2}\|z^{t-1} - z^t\|^2 - \rho\|Ax^t + B\bar{x}^t + z^t\|^2 \\
\geq&(\frac{\rho+\beta}{2} - \frac{\beta^2}{\rho})\|z^{t-1} - z^t\|^2.
\end{aligned} \tag{43}$$

The equality is due to the update of dual variable in Algorithm 1, the optimality condition (38), and the fact that $-\rho(a+b)^\top(a+c) + \frac{\rho}{2}\|a+c\|^2 - \frac{\rho}{2}\|a+b\|^2 = \frac{\rho}{2}\|c-b\|^2 - \rho\|a+b\|^2$ with $a = Ax^t + B\bar{x}^t$, $b = z^t$, and $c = z^{t-1}$; the inequality is due to $h(z)$ being β-strongly convex and (38) of Lemma 2. Since $\rho = 2\beta$, adding (41)-(43) proves (39).

To see $L_\rho(x^t, \bar{x}^t, z^t, y^t)$ is bounded from below, we note that the function $h(z)$ defined above is also Lipschitz differentiable with constant β, so define $s^t := -(Ax^t + B\bar{x}^t)$, we have $h(z^t) - (y^t)^\top(s^t - z^t) \geq h(s^t) - \frac{\beta}{2}\|s^t - z^t\|^2$. As a result, for all $t \in \mathbb{Z}_+$,

$$\begin{aligned}
L_\rho(x^t, \bar{x}^t, z^t, y^t) =&f(x^t) + h(z^t) + (y^t)^\top(Ax^t + B\bar{x}^t + z^t) + \frac{\rho}{2}\|Ax^t + B\bar{x}^t + z^t\|^2 \\
\geq&f(x^t) + h(s^t) - \frac{\beta}{2}\|s^t - z^t\|^2 + \frac{\rho}{2}\|Ax^t + B\bar{x}^t + z^t\|^2 \\
\geq&f(x^t) + h(s^t) \geq f(x^t) - \frac{\|\lambda\|^2}{2\beta},
\end{aligned} \tag{44}$$

where the last inequality is due to $h(s^t) = \frac{\beta}{2}\|s^t + \frac{\lambda}{\beta}\|^2 - \frac{\|\lambda\|^2}{2\beta}$. Since λ is bounded, there exists $M \in \mathbb{R}$ such that $\|\lambda\|^2 \leq M$; since the outer-level penalty β^k is nondecreasing, we can define $\underline{L} := f^* - M/\beta^1$, where $f^* = \min_{x\in\mathcal{X}} f(x)$. The minimum is achievable due to Assumption 2. □

Now we are ready to prove Proposition 1.

Proof By Lemma 3, for any $T \in \mathbb{Z}_{++}$ we have

$$\beta\sum_{t=1}^{T}\|B\bar{x}^{t-1} - B\bar{x}^t\|^2 + \|z^{t-1} - z^t\|^2 \leq \overline{L}_k - \underline{L},$$

which implies the existence of a particular index $t \in [T]$ such that

$$\|B\bar{x}^{t-1} - B\bar{x}^t\|^2 + \|z^{t-1} - z^t\|^2 \leq \frac{\overline{L}_k - \underline{L}}{\beta T}. \tag{45}$$

Using the fact that $\|Ax^t + B\bar{x}^t + z^t\| = \frac{\beta}{\rho}\|z^{t-1} - z^t\| = \frac{1}{2}\|z^{t-1} - z^t\|$, the KKT errors can be bounded by

$$\begin{aligned}
&\max\{\|\rho A^\top(B\bar{x}^{t-1} + z^{t-1} - B\bar{x}^t - z^t)\|, \|\rho B^\top(z^{t-1} - z^t)\|, \|Ax^t + B\bar{x}^t + z^t\|\} \\
\le&\rho \max\{\|A\|, \|B\|, 1/(2\rho)\}\left(\|B\bar{x}^{t-1} - B\bar{x}^t\| + \|z^{t-1} - z^t\|\right) \\
\le&2\sqrt{2}\beta \max\{\|A\|, \|B\|, 1\}\left(\|B\bar{x}^{t-1} - B\bar{x}^t\|^2 + \|z^{t-1} - z^t\|^2\right)^{1/2} \\
\le&2\sqrt{2}\beta \max\{\|A\|, \|B\|, 1\}\left(\frac{\overline{L}_k - \underline{L}}{\beta T}\right)^{1/2} \le \min\{\epsilon_1^k, \epsilon_2^k, \epsilon_3^k\},
\end{aligned}$$

where the first inequality is due to the triangle inequality, the second inequality is due to the Cauchy–Schwarz inequality and $\rho = \rho^k = 2\beta^k \ge 2\beta^0 \ge 1/2$, the third inequality is due to (45), and the last inequality is due to the claimed upper bound on T. □

A.2 Proof of Theorem 1

Proof Since $x^k \in \mathcal{X}$, $\bar{x}^k \in \bar{\mathcal{X}}$ and $\mathcal{X}$, $\bar{\mathcal{X}}$ are bounded, we know $\|Ax^k + B\bar{x}^k\|$ is bounded; since $\|Ax^k + B\bar{x}^k + z^k\| \le \epsilon_3^k$ and $\epsilon_3^k \to 0$, $\{z^k\}$ is also bounded. We conclude that $\{(x^k, \bar{x}^k, z^k)\}$ is bounded and therefore has at least one limit point, denoted by $(x^*, \bar{x}^*, z^*)$. We use k_r to denote a subsequence converging to $(x^*, \bar{x}^*, z^*)$. Since $\mathcal{X}$, $\bar{\mathcal{X}}$ are also closed, we have $x^* \in \mathcal{X}$ and $\bar{x}^* \in \bar{\mathcal{X}}$. Moreover, $Ax^* + B\bar{x}^* + z^* = \lim_{r\to\infty} Ax^{k_r} + B\bar{x}^{k_r} + z^{k_r} = 0$. Therefore $(x^*, \bar{x}^*)$ is feasible for problem (3) if and only if $z^* = 0$. If β^k is bounded, then according to the update scheme, we have $z^k \to 0$, so $z^* = 0$. Now suppose β^k is unbounded. Since β^k is nondecreasing, any subsequence is also unbounded. By (13c), we have

$$\frac{\lambda^{k_r}}{\beta^{k_r}} + z^{k_r} + \frac{y^{k_r}}{\beta^{k_r}} = 0. \tag{46}$$

Since $\{\lambda^{k_r}\}$ is bounded, we may assume $\lambda^{k_r} \to \lambda^*$. Again we consider two cases. In the first case, suppose $\{y^{k_r}\}$ has a bounded subsequence, and therefore has a limit point y^*. Then taking limit on both sides of (46) along the subsequence converging to y^*, we have $z^* = 0$, so $(x^*, \bar{x}^*)$ is feasible. Otherwise in the second case, $\lim_{r\to\infty} \|y^{k_r}\| = +\infty$. Denote $\tilde{y}^{k_r} := \frac{y^{k_r}}{\beta^{k_r}}$. We know the sequence $\{\tilde{y}^{k_r}\}$ converges to $-z^*$, because

$$\lim_{r\to\infty} \tilde{y}^{k_r} = \lim_{r\to\infty} \frac{y^{k_r}}{\beta^{k_r}} = \lim_{r\to\infty} -z^{k_r} - \frac{\lambda^{k_r}}{\beta^{k_r}} = -z^*. \tag{47}$$

By (13a) and (13b), we have

$$d_1^{k_r} - \nabla f(x^{k_r}) - A^\top y^{k_r} \in N_{\mathcal{X}}(x^{k_r}), \quad d_2^{k_r} - B^\top y^{k_r} \in N_{\bar{\mathcal{X}}}(\bar{x}^{k_r}).$$

Since $N_{\mathcal{X}}(x^{k_r})$ and $N_{\bar{\mathcal{X}}}(\bar{x}^{k_r})$ are cones and $\beta^{k_r} > 0$, we have

$$\frac{d_1^{k_r}}{\beta^{k_r}} - \frac{\nabla f(x^{k_r})}{\beta^{k_r}} - A^\top \tilde{y}^{k_r} \in N_{\mathcal{X}}(x^{k_r}), \quad \frac{d_2^{k_r}}{\beta^{k_r}} - B^\top \tilde{y}^{k_r} \in N_{\bar{\mathcal{X}}}(\bar{x}^{k_r}), \tag{48}$$

where $\tilde{y}^{k_r} := \frac{y^{k_r}}{\beta^{k_r}}$. Due to the closedness of normal cones, we can take limit on (48), then (46) and (13d) implies $(x^*, \bar{x}^*)$ is a stationary point of the problem (18). □

A.3 Proof of Theorem 2

Proof We assume the subsequence $\{(x^{k_r}, \bar{x}^{k_r}, z^{k_r}, y^{k_r})\}$ converges to the limit point $(x^*, \bar{x}^*, z^*, y^*)$. Using a similar argument in the proof of Theorem 1, we have $x^* \in \mathcal{X}$, $\bar{x}^* \in \bar{\mathcal{X}}$, and $Ax^* + B\bar{x}^* + z^* = 0$. It remains to show $z^* = 0$ to complete primal feasibility. If β^k is bounded, then we have $z^k \to 0$ so $z^* = 0$; if β^k is unbounded, by taking limits on both sides of (46), we also have $z^* = 0$, since λ^k is bounded and y^{k_r} converges to y^*. Therefore $(x^*, \bar{x}^*)$ satisfies (7c). Taking limits on (13a) and (13b) as $k \to \infty$, we get (7a) and (7b), respectively. This completes the proof. □

A.4 Proof of Theorem 3

Proof We use k to index outer-level iterations of Algorithm 3 and t to index inner-level iterations of Algorithm 1. By Proposition 1, Assumption 4, and the fact that $\beta^k = \beta^0\gamma^k$, the number of iterations T_k of the k-th inner ADMM, defined in (17), satisfies

$$T_k = \mathcal{O}\left(\frac{\beta^k}{\epsilon^2}\right) = \mathcal{O}\left(\frac{\gamma^k}{\epsilon^2}\right). \tag{49}$$

Summing T_k over $k \in [K]$, we obtain the following bound on the total number of ADMM iterations:

$$\sum_{k=1}^{K} T_k = \mathcal{O}\left(\frac{1}{\epsilon^2}\frac{\gamma(\gamma^K - 1)}{\gamma - 1}\right) = \mathcal{O}\left(\frac{\gamma^K}{\epsilon^2}\right). \tag{50}$$

Since conditions (19a) and (19b) are maintained at the termination of each inner-level ADMM, the total number of outer-level ALM iterations, K, depends on the rate at which (19c) is satisfied. By inequality (44) and Assumption 4, at the termination of each ADMM, we have

$$\overline{L} \geq L_{\rho^k}(x^0, \bar{x}^0, z^0, y^0) \geq f(x^k) - \langle \lambda^k, Ax^k + B\bar{x}^k \rangle + \frac{\beta^k}{2}\|Ax^k + B\bar{x}^k\|^2. \tag{51}$$

The Assumption 2, the fact that $\|\lambda^k\|$ is bounded, and the above inequality imply that

$$\|Ax^k + B\bar{x}^k\|^2 = \mathcal{O}\left(\frac{1}{\beta^k}\right) = \mathcal{O}\left(\frac{1}{\gamma^k}\right).$$

As a result, there exists an index K such that $\|Ax^K + B\bar{x}^K\| \leq \epsilon$ and $\gamma^K = \mathcal{O}\left(1/\epsilon^2\right)$. Plugging $\gamma^K = \mathcal{O}\left(1/\epsilon^2\right)$ into (50) gives the claimed $\mathcal{O}(1/\epsilon^4)$ complexity upper bound.

For the second claim, consider the K-th inner ADMM, at the termination of which we have $\|Ax^K + B\bar{x}^K + z^K\| \leq \frac{\epsilon}{2}$. Since $\|Ax^K + B\bar{x}^K\| \leq \|Ax^K + B\bar{x}^K + z^K\| + \|z^K\| \leq \frac{\epsilon}{2} + \|z^K\|$. It suffices to find an index K such that $\|z^K\| \leq \frac{\epsilon}{2}$. Since $\hat{\lambda}^k$ and λ^k are bounded, we have $\|z^k\| = \|\hat{\lambda}^k - \lambda^k\|/\beta^k = \mathcal{O}\left(1/\gamma^k\right)$. As a result, we can choose K such that $\gamma^K = \mathcal{O}(1/\epsilon)$. Plugging $\gamma^K = \mathcal{O}(1/\epsilon)$ into (50) gives the claimed $\mathcal{O}(1/\epsilon^3)$ complexity upper bound. □

A.5 Proof of Theorem 4

Proof According to [32, Theorem 4.2], given the inner ADMM penalty ρ^k, which is a constant multiple of β^k, it is sufficient to let the k-th ADMM run $T_k = \mathcal{O}((\rho^k)^2/\epsilon^2)$ iterations in order to have some $t \in [T_k]$ such that the primal and dual residuals of ADMM at iteration t are less than $\epsilon/2$. Denote this solution by $x^k = (x_1^k, \cdots, x_p^k)$. Since we update penalties in each outer iteration as $\beta^k = \beta^0\gamma^k$, the total number of inner-level iterations is bounded by

$$\sum_{k=1}^{K} T_k = \mathcal{O}\left(\sum_{k=1}^{K}\frac{(\rho^k)^2}{\epsilon^2}\right) = \mathcal{O}\left(\frac{1}{\epsilon^2}\frac{\gamma^2(\gamma^{2K} - 1)}{\gamma^2 - 1}\right) = \mathcal{O}\left(\frac{\gamma^{2K}}{\epsilon^2}\right), \tag{52}$$

where K is the total number of outer-level iterations. It remains to choose K such that $\|Ax^K - b\| \leq \epsilon$, and we consider two cases.

1. Suppose the "true" dual variable $\hat{\lambda}^k = \lambda^k + \beta^k z^k$ stays bounded. It immediately follows that $\|z^k\| = \mathcal{O}(1/\beta^k)$. To get $\|z^K\| \leq \frac{\epsilon}{2}$ so that $\|Ax^k - b\| \leq \|Ax^k + z^k - b\| + \|z^k\| \leq \epsilon$, it suffices to choose some K with $\beta^K = \mathcal{O}(1/\epsilon)$, which follows $\gamma^K = \mathcal{O}(1/\epsilon)$.
2. Otherwise, similar as in Theorem 3, since there is a uniform upper bound on the values of augmented Lagrangians, it suffices to let $\beta^K = \mathcal{O}(1/\epsilon^2)$, which follows $\gamma^K = \mathcal{O}(1/\epsilon^2)$.

Finally, plugging $\gamma^K = \mathcal{O}(1/\epsilon)$ and $\gamma^K = \mathcal{O}(1/\epsilon^2)$ into (52) will give $\mathcal{O}(1/\epsilon^4)$ and $\mathcal{O}(1/\epsilon^6)$ respectively. This completes the proof. □

B Additional Proofs in Section 5

B.1 Proof of Proposition 3

Proof Denote $L = \nabla^2 f(x^*) + \sum_{i=1}^p \mu_i^* \nabla^2 h_i(x^*)$. We firstly show under Assumption 6, there exists $\underline{\beta} > 0$ such that for all $\beta \geq \underline{\beta}$, we have $u^\top L u + \frac{\beta}{2}\|Au + Bv\|^2 > 0$ for all $(u, v) \neq 0$ and $\nabla h(x^*)^\top u = 0$. Suppose for any $k \in \mathbb{Z}_{++}$, there exists (u^k, v^k) on the unit sphere such that $\nabla h(x^*)^\top u^k = 0$ and $(u^k)^\top L u^k + \frac{k}{2}\|Au^k + Bv^k\|^2 \leq 0$. Without loss of generality, assume (u^k, v^k) converges to some $(\bar{u}, \bar{v})$, which is located on the unit sphere as well. Then we have $\bar{u}^\top L \bar{u} + \limsup_{k\to\infty} \frac{k}{2}\|Au^k + Bv^k\|^2 \leq 0$, and it follows $A\bar{u} + B\bar{v} = 0$ and $\bar{u}^\top L \bar{u} \leq 0$, which is a desired contradiction since $\nabla h(x^*)^\top \bar{u} = 0$ and $(\bar{u}, \bar{v}) \neq 0$.

Since $\bar{x}^* \in \operatorname{Int} \bar{\mathcal{X}}$, we temporarily ignore the constraint $\bar{x} \in \bar{\mathcal{X}}$ and consider the system in variables $(x, \bar{x}, \tilde{\lambda}, \mu, t, \gamma, \tilde{d}_1, \tilde{d}_2)$:

$$\begin{aligned} &\nabla f(x) - A^\top \tilde{\lambda} + \nabla h(x)\mu = \tilde{d}_1, \quad -B^\top \tilde{\lambda} = \tilde{d}_2, \\ &-Ax - B\bar{x} + t + \gamma\lambda^* - \gamma\tilde{\lambda} = 0, \quad h(x) = 0, \end{aligned}$$

which has a solution $(x, \bar{x}, \tilde{\lambda}, \mu) = (x^*, \bar{x}^*, \lambda^*, \mu^*)$ for $(t, \tilde{d}_1, \tilde{d}_2) = (0, 0, 0)$ and any $\gamma \in \mathbb{R}$. We claim that for any $\gamma \in [0, 1/\underline{\beta}]$, the Jacobian of the above system evaluated at $(x^*, \bar{x}^*, \lambda^*, \mu^*, 0, \gamma, 0, 0)$ with respect to $(x, \bar{x}, \tilde{\lambda}, \mu)$, namely, the matrix

$$\begin{bmatrix} L & 0 & -A^\top & \nabla h(x^*) \\ 0 & 0 & -B^\top & 0 \\ -A & -B & -\gamma I & 0 \\ \nabla h(x^*)^\top & 0 & 0 & 0 \end{bmatrix}, \tag{53}$$

is invertible. To see this, consider the linear system in (u, v, w, z) of proper dimensions,

$$Lu - A^\top w + \nabla h(x^*) z = 0, \tag{54a}$$
$$-B^\top w = 0, \tag{54b}$$
$$Au + Bv + \gamma w = 0, \tag{54c}$$
$$\nabla h(x^*)^\top u = 0. \tag{54d}$$

For $\gamma > 0$, notice that $u^\top$(54a) + $v^\top$(54b), together with (54c) and (54d), yields $u^\top L u + \frac{1}{\gamma}\|Au + Bv\|^2 = 0$. By the first claim we know $(u, v) = 0$; thus, $w = 0$ by (54c), and $z = 0$ by (54a) and the fact that $\nabla h(x^*)$ has full column rank. For $\gamma = 0$, using the same technique as above and (24b), we can show $(u, v) = 0$; since we also assume gradients of all equality constraints are linearly independent, we have $(w, z) = 0$ as well.

Now the Implicit Function Theorem [5, Chapter 1.2], together with a change of variable with $t = (\lambda - \lambda^*)/\beta$ and $\gamma = 1/\beta$, proves the existence and uniqueness of the continuous differentiable mappings $x(\cdot)$, $\bar{x}(\cdot)$, $\mu(\cdot)$, and $\tilde{\lambda}(\cdot)$ over S as well as (26)-(27); in addition, the δ defining S can be chosen small enough so that (28) holds. Finally, (29) follows from the Mean Value Theorem for Integrals [5, Proposition 2.14]. □

B.2 Proof of Proposition 4

Proof Notice that

$$\begin{aligned}
&\beta^k\|Ax^k + B\bar{x}^k\| = \|\tilde{\lambda}(s^k) - \lambda^k\| \le \|\tilde{\lambda}(s^k) - \lambda^*\| + \|\lambda^k - \lambda^*\| \\
\overset{(29)}{\le}& M(\|\lambda^k - \lambda^*\|^2/(\beta^k)^2 + \|\tilde{d}_1^k\|^2 + \|\tilde{d}_2^k\|^2)^{1/2} + \|\lambda^k - \lambda^*\| \\
\le& \frac{M+\beta^k}{\beta^k}\|\lambda^k - \lambda^*\| + M(\|d_1^k\| + \beta^k\|A\|\|d_3^k\|) + M(\|d_2^k\| + \beta^k\|B\|\|d_3^k\|) \\
\overset{(30)}{\le}& \frac{M+\beta^k}{\beta^k}\|\lambda^k - \lambda^*\| + M\eta\|Ax^k + B\bar{x}^k\|,
\end{aligned}$$

which implies for $\beta^k > M\eta$,

$$\|Ax^k + B\bar{x}^k\| \le \frac{M+\beta^k}{\beta^k(\beta^k - M\eta)}\|\lambda^k - \lambda^*\|. \tag{55}$$

Similarly, we have

$$\begin{aligned}
&\|\hat{\lambda}^k - \lambda^*\| \le \|\tilde{\lambda}(s^k) - \lambda^*\| + \|\hat{\lambda}^k - \tilde{\lambda}(s^k)\| \\
\overset{(29)}{\le}& \frac{M}{\beta^k}\|\lambda^k - \lambda^*\| + M(\|d_1^k\| + \beta^k\|A\|\|d_3^k\|) + M(\|d_2^k\| + \beta^k\|B\|\|d_3^k\|) + \beta^k\|d_3^k\| \\
\overset{(30)}{\le}& \frac{M}{\beta^k}\|\lambda^k - \lambda^*\| + M\eta\|Ax^k + B\bar{x}^k\| \overset{(55)}{\le} \left(\frac{M}{\beta^k} + \frac{M\eta(M+\beta^k)}{\beta^k(\beta^k - M\eta)}\right)\|\lambda^k - \lambda^*\|.
\end{aligned}$$

This completes the proof. □