IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received March 31, 2019, accepted April 22, 2019, date of publication May 2, 2019, date of current version June 25, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2914461

Linearized ADMM for Nonconvex Nonsmooth Optimization With Convergence Analysis

QINGHUA LIU, (Student Member, IEEE), XINYUE SHEN, (Student Member, IEEE), AND YUANTAO GU, (Senior Member, IEEE)

Beijing National Research Center for Information Science and Technology (BNRist), Tsinghua University, Beijing 100084, China
Department of Electronic Engineering, Tsinghua University, Beijing 100084, China

Corresponding author: Yuantao Gu (gyt@tsinghua.edu.cn)

This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant 61531166005 and Grant 61571263, and in part by the National Key Research and Development Program of China under Grant 2016YFE0201900 and Grant 2017YFC0403600.

ABSTRACT Linearized alternating direction method of multipliers (ADMM) as an extension of ADMM has been widely used to solve linearly constrained problems in signal processing, machine learning, communications, and many other fields. Despite its broad applications in nonconvex optimization, for a great number of nonconvex and nonsmooth objective functions, its theoretical convergence guarantee is still an open problem. In this paper, we propose a two-block linearized ADMM and a multi-block parallel linearized ADMM for problems with nonconvex and nonsmooth objectives. Mathematically, we present that the algorithms can converge for a broader class of objective functions under less strict assumptions compared with previous works. Furthermore, our proposed algorithm can update coupled variables in parallel and work for less restrictive nonconvex problems, where the traditional ADMM may have difficulties in solving subproblems.

INDEX TERMS Linearized ADMM, multi-block ADMM, nonconvex optimization, parallel computation, proximal algorithm.

I. INTRODUCTION

In signal processing [1], machine learning [2], and communication [3], many of the recently most concerned problems, such as compressed sensing [4], dictionary learning [5], and channel estimation [6], can be cast as optimization problems. In doing so, not only has the design of the solving methods been greatly facilitated, but also a more mathematically understandable and manageable description of the problems has been given. While convex optimization has been well studied [7]–[9], nonconvex optimization has also appeared in numerous topics such as matrix factorization [10], [11], phase retrieval [12], and clustering [13].

The alternating direction method of multipliers (ADMM) has been widely used in linearly constrained optimization problems arising in machine learning [14], [15], signal processing [16], as well as other fields [17]–[19]. First proposed in the early 1970s, it has been studied extensively [20]–[22]. At the very beginning, ADMM was mainly applied in solving linearly constrained convex problems in the following form [23]

$$\begin{aligned} &\text{minimize} \quad f(x)+h(y) \\ &\text{subject to} \quad Ax+By=0, \end{aligned} \tag{1}$$

where $x \in \mathbb{R}^p, y \in \mathbb{R}^q$ are variables, and $A \in \mathbb{R}^{n\times p}, B \in \mathbb{R}^{n\times q}$ are given. With an augmented Lagrangian function defined as

$$L_\beta(x,y,\gamma)=f(x)+h(y)+\langle \gamma, Ax+By\rangle+\frac{\beta}{2}\|Ax+By\|_2^2, \tag{2}$$

where γ is the Lagrangian dual variable, the ADMM method updates variables iteratively as the following

$$\begin{aligned} x^{k+1} &= \arg\min_x L_\beta(x, y^k, \gamma^k), \\ y^{k+1} &= \arg\min_y L_\beta(x^{k+1}, y, \gamma^k), \\ \gamma^{k+1} &= \gamma^k+\beta(Ax^{k+1}+By^{k+1}). \end{aligned}$$

For ADMM applied in nonconvex problems, although the theoretical convergence guarantee is still an open problem,

The associate editor coordinating the review of this manuscript and approving it for publication was Hong-Mei Zhang.

it can converge fast in many cases [24], [25]. Under certain assumptions on the objective function and linear constraints, researchers have studied the convergence of ADMM for nonconvex optimization [26]–[31].

The subproblems in ADMM can be hard to solve and have no closed form solution in many cases, so we either use an approximate solution as a substitute in the update which might cause divergence, or solve the subproblems by numerical algorithms which can bring computational burden. Motivated by these issues, linearized ADMM was proposed for convex optimization [32]–[37]. By linearizing differentiable functions in subproblems, they make subproblems easier to solve and reduce computational complexity. It has demonstrated good performances in sparsity recovery [35], [37], [38], low-rank matrix completion [39], and image restoration [40]–[43].

When the problem scale is so large that a two-block ADMM method may no longer be efficient or practical [44], [45], distributed algorithms are in demand to exploit parallel computing resources [8], [46], [47]. Multi-block ADMM was proposed to solve problems in the following form [14]

$$\begin{aligned} &\text{minimize} \quad f_1(x_1)+f_2(x_2)+\cdots+f_K(x_K) \\ &\text{subject to} \quad A_1x_1+A_2x_2+\cdots+A_Kx_K=0. \end{aligned} \tag{3}$$

It allows parallel computation [20], [27], [48]–[51], and has been used in problems such as sparse statistical machine learning [52] and total variation regularized image reconstruction [53].

A. MAIN PROBLEMS

In this paper, we study linearized ADMM algorithms for problems with nonconvex and nonsmooth objective functions. First, we propose a two-block linearized ADMM for problems with *coupled* variables in the following form

$$\begin{aligned} &\text{minimize} \quad g(x,y)+f(x)+h(y) \\ &\text{subject to} \quad Ax+By=0, \end{aligned} \tag{4}$$

where $x\in\mathbb{R}^p, y\in\mathbb{R}^q$ are variables. Functions g and h are differentiable and can be nonconvex. Function f can be both nonconvex and nondifferentiable. The Lagrangian function for problem (4) is defined as follows

$$\begin{aligned} L_\beta(x,y,\gamma) = {} & g(x,y)+f(x)+h(y) \\ & +\langle\gamma, Ax+By\rangle+\frac{\beta}{2}\|Ax+By\|_2^2. \end{aligned} \tag{5}$$

Throughout the paper we make the following assumption.

Assumption 1: Assume that problem (4) satisfies the conditions below.

1. Function $h(y)$ is L_h-Lipschitz differentiable.
2. Function $g(x,y)$ is L_g-Lipschitz differentiable.
3. Function $g(x,y)+f(x)+h(y)$ is lower bounded and coercive with respect to y over the feasible set
$$\left\{(x,y)\in\mathbb{R}^{p+q} : Ax+By=0\right\}.$$
4. Matrix B has full column rank, and $Im(A)\subset Im(B)$.

In Assumption 1, we put relatively weak restriction on function f and matrix A, which is a significant improvement over other works on nonconvex ADMM algorithms.

Then we propose a parallel multi-block ADMM method, which can be seen as a special case of the first algorithm, for problems in the following form

$$\begin{aligned} &\text{minimize} \quad g(x_1,\ldots,x_K,y)+\sum_{i=1}^{K}f_i(x_i)+h(y) \\ &\text{subject to} \quad A_1x_1+\cdots+A_Kx_K+By=0, \end{aligned} \tag{6}$$

where $x=(x_1,\ldots,x_K)$ and y are variables. The assumption we have on problem (6) is the same as Assumption 1.

B. RELATED WORKS

Recently a great deal of attention has been focused on using ADMM to solve nonconvex problems [26]–[31]. The work [26] studies the convergence of traditional ADMM under relatively strict assumptions. For instance, it requires every A_i to have full column rank and all the f_i to satisfy an assumption similar to Holder condition. Besides, the parameter β in their algorithm is required to increase linearly in the number of variable blocks, which can seriously reduce its convergence speed. The work [27] studies the convergence of ADMM for solving nonconvex consensus and sharing problem. However, they require the nonconvex part to be Lipschitz differentiable and the nondifferentiable part to be convex. The work [27] also studies a parallel ADMM, but it is only under the case where the Lagarangian function is separable across all blocks, that is, the objective function and augmented term are both separable. The work [28] studies nonconvex ADMM under less restrictive assumptions. Their algorithm requires matrix B to have full row rank, while our algorithm requires matrix B to have full column rank, so their algorithm adapts to different optimization problems from ours. In addition, our second algorithm allows parallel computation for multi-block cases, while theirs does not. A detailed comparison on conditions for the convergence of these algorithms is listed in Table 1.

Besides ADMM there are also other kinds of dual algorithms for multi-block nonconvex optimization. For instance, [46] studies a distributed dual algorithm for nonconvex constrained problem, where the integral objective function is Lipschitz differentiable and the Lagrangian function is defined without the augmented term. It can be viewed as a variation of the *method of Lagrangian multiplier*, while our algorithms are variations of the *Augmented Lagrangian method*. In addition, our algorithms can adapt to nonsmooth optimization even with indicator functions in the objective, while their algorithm cannot.

C. CONTRIBUTION

Our work has the following improvements compared with some latest works based on ADMM for nonconvex optimization.

TABLE 1. Comparison with related works.

Method	Formulation	Conditions on objective function and feasible set
linearized ADMM in Algorithm 1	$\begin{array}{ll}\text{minimize} & g(x,y)+f(x)+h(y)\\ \text{subject to} & Ax+By=0\end{array}$	g and h are Lipchitz differentiable; $g+f+h$ is lower bounded and coercive w.r.t. y over the feasible set; B has full column rank; $Im(A)\subset Im(B)$
ADMM [26]	$\begin{array}{ll}\text{minimize} & \phi(x,y)=g(x)+\sum_{i=0}^{p}f_i(x_i)+h(y)\\ \text{subject to} & Ax+By=0\end{array}$	g and h are Lipchitz differentiable; ϕ is coercive over the feasible set; f_0 is lower semi-continuous and $f_1,\dots f_p$ are restricted prox-regular; $Im(A)\subset Im(B)$; solution to subproblem is Lipchitz w.r.t. input
ADMM [26]	$\begin{array}{ll}\text{minimize} & \phi(x,y)\\ \text{subject to} & Ax+By=0\end{array}$	ϕ is Lipchitz differentiable; $Im(A)\subset Im(B)$; solution to subproblem is Lipchitz w.r.t. input
flexible ADMM [27]	$\begin{array}{ll}\text{minimize} & f(x)=\sum_{k=1}^{K}g_k(x_k)+l(x_0)\\ \text{subject to} & \sum_{k=1}^{K}A_kx_k=x_0,\\ & x_k\in X_k, k=1,\dots,K\end{array}$	l is Lipchitz differentiable; $g_1,\dots,g_K$ are either Lipchitz differentiable or convex; f is lower bounded over the feasible set; $X_1,\dots,X_K$ are closed convex sets; $A_1,\dots,A_K$ have full column rank
flexible ADMM [27]	$\begin{array}{ll}\text{minimize} & f(x)=\sum_{k=1}^{K}g_k(x_k)+h(x_0)\\ \text{subject to} & x_k=x_0, k=1,\dots,K, x_0\in X\end{array}$	$g_1,\dots,g_K$ are Lipchitz differentiable; h is convex; f is lower bounded on X; X is a closed convex set
flexible Proximal ADMM [27]	$\begin{array}{ll}\text{minimize} & f(x)=\sum_{k=1}^{K}g_k(x_k)+h(x_0)\\ \text{subject to} & x_k=x_0, k=1,\dots,K, x_0\in X\end{array}$	$g_1,\dots,g_K$ are Lipchitz differentiable; h is convex; f is lower bounded on X; X is a closed convex set
Bregman ADMM [55]	$\begin{array}{ll}\text{minimize} & f(x)+g(y)+h(z)\\ \text{subject to} & Ax+By+Cz=0\end{array}$	f and g are proper and lower semi-continuous h is smooth and Lipchitz continuous; $f+g$ is coercive; $f+g+h$ is sub-analytic; g and f are lower bounded; $h(z)-\beta_0\|\nabla h(z)\|^2$ is lower bounded for some $\beta_0>0$; either $h(z)-\beta_0\|\nabla h(z)\|^2$ is coercive or C is square; C has full row rank
proximal gradient-based ADMM (or proximal majorization ADMM) [28]	$\begin{array}{ll}\text{minimize} & f(x_1,\dots,x_N)+\sum_{i=1}^{N-1}r_i(x_i)\\ \text{subject to} & \sum_{i=1}^{N}A_ix_i=b,\\ & x_i\in X_i, i=1,\dots,N-1\end{array}$	f is Lipchitz differentiable; f and $r_1,\dots,r_{N-1}$ are lower bounded over the domain; either r_i is Lipchitz continuous and X_i is compact or r_i is lower semi-continuous and $X_i=\mathbb{R}^{n_i}$; $A_N=I$ (not required for proximal majorization ADMM)
ADMM [29]	$\begin{array}{ll}\text{minimize} & f(x)+g(y)\\ \text{subject to} & Ax+y=b\end{array}$	f is proper, lower semi-continuous, and semi-algebra; g is continuously differentiable, Lipchitz differentiable, and semi-algebra; $A^TA\succeq\mu I$ for some $\mu>0$; the generated sequence is bounded
ADMM [30]	$\begin{array}{ll}\text{minimize} & \Psi(L)+\Phi(S)+0.5\|D-\mathcal{A}(Z)\|_F^2\\ \text{subject to} & \mathcal{B}(L)+\mathcal{C}(S)=Z,\end{array}$ where L and S are variables, D is given, and $\mathcal{A},\mathcal{B}$, and $\mathcal{C}$ are linear mappings	Ψ and Φ are proper, closed, and nonnegative; Ψ is convex and continuous in its domain; $\liminf_{\|L\|_F+\|S\|_F\to\infty}\Psi(L)+\Phi(S):=h_0>\Theta(L^1,S^1,Z^1,\Lambda^1),$ where Θ is an augmented Lagrangian function; $\mathcal{B}^*\mathcal{B}\succeq\sigma_1 I$ and $\mathcal{C}^*\mathcal{C}\succeq\sigma_2 I$ for some $\sigma_1,\sigma_2>0$
proximal ADMM [31]	$\begin{array}{ll}\text{minimize} & h(x)+p(y)\\ \text{subject to} & y=\mathcal{M}(x),\end{array}$ where $\mathcal{M}$ is a linear mapping	p is proper and closed, and proximal mapping of p exists; h is twice continuous differentiable with bounded Hessian; $\mathcal{M}^*\mathcal{M}\succeq\sigma I$ for some $\sigma>0$

- **Nonconvex Linearized ADMM:** To the best of our knowledge, this is the first work to study the theoretical convergence for a *completely linearized* ADMM in nonconvex optimization. By linearizing all the differentiable parts, not only the objective function but also the augmented term, in the Lagrangian function, the subproblems can either be transformed into a proximal problem or a quadratic problem, which are usually easier to solve than the original subproblems.
- **Parallel Computation:** In our second algorithm, the linearization decouples the variables $x_1,\dots,x_K$ originally coupled in the function g and $\frac{\beta}{2}\|\sum_{i=1}^{K}A_ix_i+By\|_2^2$, so we can update every block in parallel. Previous works [20], [49]–[51] have studied some parallel

ADMM algorithms that can deal with coupled variables, but they are all for convex optimization. To the best of our knowledge, our second algorithm is the first one to extend such parallel ADMM to nonconvex optimization. Numerical experiment demonstrates the high efficiency of our algorithm brought by parallel computation in comparison with other latest nonconvex ADMM algorithms.

- **Weaker Assumptions:** Our assumptions are less restrictive in comparison with previous works on nonconvex ADMM (see, e.g., [26], [27], [29]–[31]). Specifically, we put much weaker restrictions on function f (f_i) and matrix A (A_i). The work [28] needs assumptions similar to ours, but the update rules are different, and their algorithm requires matrix B to have full row rank, while we require matrix B to have full column rank.

D. OUTLINE

The remainder of this paper is organized as follows. In Section II some preliminaries are introduced. In Section III-A we propose a two-block linearized ADMM for nonconvex problems and provide convergence analysis under certain broad assumption in Section III-B. In Section III-C we propose a parallel multi-block linearized ADMM that can be seen as a special case of the first algorithm. Section IV gives detailed discussions on the update rules and some applications to demonstrate the advantages of this work. In section V, numerical experiments are performed to demonstrate the effectiveness and high efficiency of our algorithms. We conclude this work in Section VI.

II. PRELIMINARY

A. NOTATION

We use bold capital letters for matrices, bold small case letters for vectors, and non-bold letters for scalars. We use x^k to denote the value of x after the kth iteration and x_i to denote its ith block. The gradient of function f at x for the ith component is denoted as $\nabla_{x_i} f(x)$, and the *regular subgradient* of f for the ith component defined at a point x [55], is denoted as $\partial_i f(x)$. The smallest eigenvalue of matrix X is denoted as λ_X. Without specification, $\|\cdot\|$ denotes ℓ_2 norm. $Im(X)$ denotes the image of matrix X. In multi-block ADMM, $x = (x_1, \ldots, x_K)$ denotes the collection of variables.

B. DEFINITION

Definition 1 (Regular Subgradient [55]): Consider a function $f : \mathbb{R}^n \to \bar{\mathbb{R}}$ and a point x_0 with $f(x_0)$ finite. Then the regular subgradient of function f at x_0 is defined as

$$\partial f(x_0) = \{v : f(x) \geq f(x_0) + \langle v, x - x_0\rangle + o(\|x - x_0\|)\},$$

where for every v the inequality holds for any x in a small neighborhood of x_0.

Remark 1: Notice that the regular subgradient is a set. For a differentiable function, its regular subgradient set at a point contains only its gradient at that point.

Definition 2 (Lipschitz Differentiable): Function $s(y)$ is said to be L_s-Lipschitz differentiable if for all y, y', we have that

$$\|\nabla s(y) - \nabla s(y')\| \leq L_s \|y - y'\|,$$

which is equivalent to that its gradient ∇s is Lipschitz continuous.

Definition 3 (Coercive Function): Assume that function $r(x_1, x_2)$ is defined on $\mathcal{X}$, and for any $\|x_2^k\| \to +\infty$ and $(x_1^k, x_2^k) \in \mathcal{X}$, we have $r(x_1^k, x_2^k) \to +\infty$, then function r is said to be coercive with respect to x_2 over $\mathcal{X}$.

Remark 2: Any function is coercive over bounded set.

III. LINEARIZED ADMM: TWO-BLOCK AND MULTI-BLOCK

In this section, we first propose a linearized ADMM to solve the two-block nonconvex problem (4) possibly with function f nonsmooth. Its convergence assumption is, as far as we know, one of the broadest among the current ADMM algorithms for nonconvex optimization. Then we extend the algorithm to solve the multi-block problem (6), and the linearization renders the coupled multi-blocks of variables to be updated in parallel.

A. TWO-BLOCK LINEARIZED ADMM UPDATING RULES

In the $(k+1)$th update of x, we replace the objective in the subproblem of ADMM $g(x, y^k) + \frac{\beta}{2}\|Ax + By^k\|^2$ by the following

$$\langle x - x^k, \nabla_x g(x^k, y^k) + \beta A^T(Ax^k + By^k)\rangle + \frac{L_x}{2}\|x - x^k\|^2,$$

which is a linearized term plus a regularization term ($L_x > 0$). In the $(k+1)$th update of y, the algorithm replaces $g(x^{k+1}, y) + h(y)$ by the following

$$\langle y - y^k, \nabla_y g(x^{k+1}, y^k) + \nabla h(y^k)\rangle + \frac{L_y}{2}\|y - y^k\|^2,$$

which is again a linearized term plus a regularization term ($L_y > 0$). Replacing the corresponding parts in the augmented Lagrangian function with their approximations above, we readily get the following two auxiliary functions.

$$\begin{aligned}\bar{f}^k(x) = {} & f(x) + \langle \gamma^k, Ax\rangle + \frac{L_x}{2}\|x - x^k\|^2 \\ & + \langle x - x^k, \nabla_x g(x^k, y^k) + \beta A^T(Ax^k + By^k)\rangle; \quad (7)\\ \bar{h}^k(y) = {} & \langle \gamma^k, By\rangle + \frac{L_y}{2}\|y - y^k\|^2 + \frac{\beta}{2}\|Ax^{k+1} + By\|^2 \\ & + \langle y - y^k, \nabla_y g(x^{k+1}, y^k) + \nabla h(y^k)\rangle. \quad (8)\end{aligned}$$

Utilizing the two auxiliary functions above, the update rules are summarized in Algorithm 1. Note that the x and y update rules in Algorithm 1 can be simplified into the following form

$$\begin{aligned}x^{k+1} = prox_{f/L_x}\Bigg\{x^k - \frac{1}{L_x}\Big[\nabla_x g(x^k, y^k) + A^T\gamma^k \\ + \beta A^T(Ax^k + By^k)\Big]\Bigg\};\end{aligned}$$

Algorithm 1 Two-Block Linearized ADMM Algorithm

Initialize x^0, y^0, γ^0.
while $\max\{\|x^k - x^{k-1}\|, \|y^k - y^{k-1}\|, \|\gamma^k - \gamma^{k-1}\|\} > \varepsilon$ **do**
  $x^{k+1} = arg\,min_x \bar{f}^k(x)$
  $y^{k+1} = arg\,min_y \bar{h}^k(y)$
  $\gamma^{k+1} = \gamma^k + \beta(Ax^{k+1} + By^{k+1})$
  $k = k + 1$
end while
return (x^k, y^k, γ^k)

$$y^{k+1} = \left(L_y + \beta B^T B\right)^{-1}\left(L_y y^k - \nabla_y g(x^{k+1}, y^k) - \nabla h(y^k) - B^T\gamma^k - \beta B^T A x^{k+1}\right).$$

The subproblem in updating x is formulated into a proximal problem, which can be easier to solve than the original subproblem and even have closed form solution [56]. The matrix inversion in the y-updating step can be computed beforehand, so we do not need to compute it in every iteration.

B. CONVERGENCE ANALYSIS

We give convergence analysis for Algorithm 1 under Assumption 1. Note that in this part, we refer L_β to the augmented Lagrangian function defined in (5). To begin with, we show that L_β and the primal and dual residues are able to converge in the following theorem.

Theorem 1: For the linearized ADMM in Algorithm 1, under Assumption 1, if we choose parameters L_x, L_y, and β as follows

$$\begin{aligned} L_x &\geq L_g + \beta L_{A^TA} + 6L_w^2 + 1, \\ L_y &\geq L_w + L_w^2 + 3, \\ C_m &= \frac{L_y + L_w^2}{2}, \\ \beta &\geq \max\left\{\frac{L_w + L_y + 2}{\lambda_{B^TB}}, \frac{3(L_w^2 + L_y^2)}{\lambda_{B^TB}C_m}, \frac{3L_y^2}{\lambda_{B^TB}}\right\}, \end{aligned} \quad (9)$$

where L_{A^TA} is the largest eigenvalue of A^TA, λ_{B^TB} is the smallest eigenvalue of B^TB and $L_w = L_g + L_h$, then $\{L_\beta(x^k, y^k, \gamma^k)\}$ is convergent, and the primal residues $\|y^{k+1} - y^k\|$, $\|x^{k+1} - x^k\|$ and dual residue $\|\gamma^{k+1} - \gamma^k\|$ converge to zero as k approaches infinity.

Proof: We briefly introduce the structure of the proof here and the detailed version is postponed to Appendix VII-A.

First, we will prove that the descent of L_β after the $(k+1)$th iteration of x is lower bounded by $\|x^{k+1} - x^k\|$, the descent of L_β after the $(k+1)$th iteration of y is lower bounded by $\|y^{k+1} - y^k\|$, and the ascent of L_β after the $(k+1)$th iteration of γ is upper bounded by $\|x^{k+1} - x^k\|$, $\|y^{k+1} - y^k\|$ and $\|y^k - y^{k-1}\|$. Then, we will elaborately design an auxiliary sequence and prove its monotonicity and convergence. Finally, based on these conclusions, we will obtain the convergence of L_β and both the primal and dual residues. ■

Algorithm 2 Multi-Block Parallel Linearized ADMM Algorithm

Initialize x^0, y^0, γ^0.
while $\max\{\|x^k - x^{k-1}\|, \|y^k - y^{k-1}\|, \|\gamma^k - \gamma^{k-1}\|\} > \varepsilon$ **do**
  for $i = 1, \ldots, K$ **in parallel do**
    $x_i^{k+1} = arg\,min_{x_i} \bar{f}_i^k(x_i)$
  end for
  $y^{k+1} = arg\,min_y \bar{h}^k(y)$
  $\gamma^{k+1} = \gamma^k + \beta(Ax^{k+1} + By^{k+1})$
  $k = k + 1$
end while
return (x^k, y^k, γ^k)

Theorem 1 illustrates that the function L_β will converge, and the increments of x, y, and γ after one iteration, which are the primal and dual residues, will converge to zero.

Corollary 1: For the linearized ADMM in Algorithm 1, under Assumption 1 together with function $g(x, y)$ degenerating to $g(x)$, if we choose the parameters L_x, L_y and β satisfying (9), then the generated dual variable sequence $\{\gamma^k\}$ is bounded.

Proof: The proof is postponed to Appendix VII-B. ■

Theorem 2: For the linearized ADMM in Algorithm 1, under Assumption 1, if we choose the parameters L_x, L_y, and β satisfying (9), then the sequence $\{(x^k, y^k, \gamma^k)\}$ satisfies

$$\lim_{k\to\infty} \nabla_\gamma L_\beta(x^k, y^k, \gamma^k) = \lim_{k\to\infty} Ax^k + By^k = 0,$$
$$\lim_{k\to\infty} \nabla_y L_\beta(x^k, y^k, \gamma^k) = 0,$$

and that there exits

$$\bar{d}^k \in \partial_x L_\beta(x^k, y^k, \gamma^k) \text{ such that } \lim_{k\to\infty} \bar{d}^k = 0.$$

Proof: The proof is postponed to Appendix VII-C. ■

Theorem 2 illustrates that as k goes to infinity, the left-hand-side of the original linear constraint will converge to zero, where the feasibility is reached, and the derivative of the Lagrangian function with respective to primal variables will converge to zero. In other words, the limit points of $\{(x^k, y^k)\}$, if exist, should be saddle points of L_β, alternatively KKT points to the original linearly constrained problem.

Corollary 2: For the linearized ADMM in Algorithm 1, under Assumption 1 together with function $g(x, y)$ degenerating to $g(x)$, if we choose the parameters L_x, L_y, and β satisfying (9), then the sequence $\{g(x^k) + f(x^k) + h(y^k)\}$ is convergent.

Proof: The proof is postponed to Appendix VII-D. ■

C. MULTI-BLOCK PARALLEL LINEARIZED ADMM

In this part, we focus the multi-block optimization problem (6), which can be seen as a special case of problem (4), where $f(x)$ is further assumed to be separable across the

blocks x_i for $i = 1, \ldots, K$. This case is very common in sparse recovery [52], dictionary learning [5], etc, where K is usually very large due to the high dimension of data. We apply Algorithm 1 to problem (6) and arrive at a multi-block linearized ADMM, which can update blocks of variables in parallel even when they are coupled in the Lagrangian function.

To be specific, because of the linearization we use in the x-updating step, the blocks $x_1, \ldots, x_K$ are decoupled in $\bar{f}^k(x)$, so they can be optimized in parallel. In this case, we have $\bar{f}^k(x) = \sum_{i=1}^K \bar{f}_i^k(x_i)$, where

$$\bar{f}_i^k(x_i) = f_i(x_i) + \langle \gamma^k, A_i x_i \rangle + \frac{L_x}{2}\|x_i - x_i^k\|^2 + \left\langle x_i - x_i^k, \nabla_{x_i} g(x^k, y^k) + \beta A_i^T (Ax^k + By^k) \right\rangle. \quad (10)$$

Utilizing the auxiliary functions (8) and (10), the update rules are listed in Algorithm 2. Similar to Algorithm 1, the updating rules for x and y in Algorithm 2 can be simplified into the following form

$$x_i^{k+1} = prox_{f_i/L_x}\left\{x_i^k - \frac{1}{L_x}\left[A_i^T \gamma^k + \nabla_{x_i} g(x^k, y^k) + \beta A_i^T (Ax^k + By^k)\right]\right\};$$

$$y^{k+1} = \left(L_y + \beta B^T B\right)^{-1}\left(L_y y^k - \nabla_y g(x^{k+1}, y^k) - \nabla h(y^k) - B^T \gamma^k - \beta B^T A x^{k+1}\right).$$

Because Algorithm 2 can be seen as a special case of Algorithm 1, by replacing $f(x)$ with $\sum_{i=1}^K f_i(x_i)$ the theoretical convergence analyses for Algorithm 1 can be directly applied to Algorithm 2, so its convergence assumptions and results remain the same.

IV. DISCUSSION

In this section, we give some discussion on our algorithms and their possible applications.

A. PROXIMAL TERM

There are two main reasons why we use the proximal term in our algorithms. Firstly, in the proof of Lemma 5 and 6, we will show that the descent of the Lagrangian function from updating primal variables is guaranteed due to the proximal term, so we do not need to impose any more restriction on f (or f_i). Secondly, while we enjoy the benefits of variable decoupling due to the linearization, the updates in each iteration can be viewed as inexact solutions to the original ADMM subproblems, and intuitively the proximal term controls this inexactness so that the algorithm can converge.

B. TIME EFFICIENCY

As mentioned above, the updates in each iteration are solutions to the linearized subproblems, not the original ADMM subproblems, so intuitively more iterations would be needed. However, the linearization also decouples the variables coupled in the Lagrangian function, which reduces the time cost of a single iteration due to parallel computation. As a result, the time cost of the algorithm is determined by the balance between the increase in number of iterations and the acceleration from parallel computation. In Section V, we will empirically demonstrate that the acceleration can overwhelm the deceleration. Therefore, our algorithm can enjoy higher time efficiency in comparison with other nonconvex ADMM algorithms without linearization.

C. APPLICATION

In this part we present that the following general classes of problems can meet the requirements in Assumption 1. Consequently, our theorems guarantee the convergence of the algorithms, if the problem belongs to one of the following commonly encountered classes.

1) SPARSITY RELATE TOPICS

Assume that $l(x)$ is a loss function satisfying the following conditions.

- **Lipschitz Differentiability:** l is differentiable, and there exits constant L such that $\|\nabla l(x_1) - \nabla l(x_2)\| \leq L\|x_1 - x_2\|$ for any x_1, x_2.
- **Coercivity:** $l(x)$ tends to infinity as $\|x\|$ tends to infinity.

Then the following general sparsity related problem can be solved by our algorithm with convergence guarantee

$$\begin{aligned} &\text{minimize} \quad \lambda \sum_{i=1}^N F(x_i) + l(y - b) \\ &\text{subject to} \quad Ax - y = 0, \end{aligned} \quad (11)$$

where $F(\cdot)$ is some sparsity inducing function. For example, F can be the ℓ_p-norm ($0 \leq q \leq 1$) or other nonconvex sparsity measure. It is easy to verify that the above problem satisfies Assumption 1.

2) INDICATOR FUNCTION OF COMPACT MANIFOLD

The indicator function of a compact manifold $\mathcal{M}$ is defined as follows

$$\tau(x) = \begin{cases} +\infty & x \notin \mathcal{M}, \\ 0 & x \in \mathcal{M}. \end{cases}$$

Remark 3: Consider the following general form of integer programming, where $\mathcal{M}$ is a finite subset of $\mathbb{Z}$, and f is lower bounded over $\mathcal{M}$.

$$\text{minimize } f(y) \quad \text{subject to } y \in \mathcal{M}. \quad (12)$$

Integer programming is widely used in network design [57], smart grid [58], statistic learning [59], and other fields [60]. Problem (12) can be converted to the following

$$\begin{aligned} &\text{minimize} \quad \tau(x) + \big(f(x) - h(x)\big) + h(y) \\ &\text{subject to} \quad x = y, \end{aligned} \quad (13)$$

where $\tau(x)$ is the indicator function of $\mathcal{M}$, and function h can be any nonzero Lipschitz function. It can be verified that

problem (13) satisfies Assumption 1, if h is Lipchitz differentiable. For a specific problem, function h can be appropriately chosen so that the linearized subproblems are easy to solve.

V. NUMERICAL EXPERIMENT

In this section, we solve a nonconvex regularized LASSO by Algorithm 2 and two other reference ADMM algorithms, in order to show the convergence behavior of our method and its advantage in run time brought by parallel computation.

In sparsity related fields, many works have hinted that nonconvex penalties can induce better sparsity than the convex ones (see, e.g., [61]–[63] etc). Our problem of interest is an improvement over LASSO, where the traditional l_1-norm is replaced by a more effective nonconvex sparsity measure [25]. The optimization problem is as the following

$$\text{minimize} \quad \lambda \sum_{i=1}^{N} F(x_i) + \|Ax - b\|^2, \tag{14}$$

where $x \in \mathbb{R}^N$ is the variable, and $A \in \mathbb{R}^{M\times N}$ and $b \in \mathbb{R}^M$ are given. The function F is defined as

$$F(t) = \begin{cases} |t| - \eta t^2, & |t| \le \frac{1}{2\eta}; \\ \frac{1}{4\eta}, & |t| > \frac{1}{2\eta}, \end{cases}$$

where $\eta > 0$ is a parameter and $F(t)$ is nonconvex and nonsmooth.

By introducing $y = Ax$, problem (14) is rewritten as

$$\begin{aligned} &\text{minimize} \quad \lambda \sum_{i=1}^{N} F(x_i) + \|y - b\|^2, \\ &\text{subject to} \quad Ax - y = 0, \end{aligned} \tag{15}$$

where x and y are variables. To the best of our knowledge, among the existing nonconvex ADMM algorithms, only the Algorithm 1 in [26] (referred as Ref1 here), the Algorithm 3 in [28] (referred as Ref2 here), and our algorithm can be theoretically guaranteed to converge for this problem. We will compare the efficiency of these algorithms.

A. RUNNING TIME AND CONVERGENCE CURVE

In the experiment, we set $N = 1024$, $M = 256$, $\lambda = 0.1$, and $\eta = 0.1$. Matrix A is a Gaussian random matrix and vector b is a Gaussian random vector. In order to simplify the procedure of choosing parameters, matrix A is normalized by a scalar, so that the largest eigenvalue of AA^T is 1.

For our algorithm, the parameters are set according to Theorem 1 as $\beta = 12$, $L_x = 37$, and $L_y = 8$, and we implement the parallel computation by matrix multiplication in MATLAB. For the reference algorithms, according to Lemma 7 and Lemma 9 in [26] the parameter β in Ref1 should be no less than 100, so we set it to be 100, considering that the larger the β is, the slower the convergence becomes. Similarly, according to Theorem 3.18 in [28], we choose its parameters as $L = 2$ and $\beta = 36$ in Ref2. The stopping criterion of all these methods are set as

$$\max\left\{\|x^k - x^{k-1}\|, \|y^k - y^{k-1}\|, \|Ax^k - y^k\|\right\} < \epsilon. \tag{16}$$

TABLE 2. Average CPU running time with parameters chosen by theorems.

Algorithm	$\epsilon = 10^{-3}$	$\epsilon = 10^{-4}$
Ref1	$> 1000s$	$> 1000s$
Ref2	$162.76s$	$205.21s$
Our Algorithm	$45.98s$	$60.91s$

TABLE 3. Average CPU running time with best parameters.

Algorithm	$\epsilon = 10^{-3}$	$\epsilon = 10^{-4}$	$\epsilon = 10^{-7}$
Ref1	$230.23s$	$257.84s$	$439.3s$
Ref2	$75.92s$	$83.31s$	$109.57s$
Our Algorithm	$12.56s$	$15.23s$	$20.76s$

We perform 1000 independent trials on MATLAB 2016a with a 3.4 GHz Intel i7 processor, and the A and b in each trial is generated randomly. The average CPU running time is shown in Table 2. We can see that our algorithm enjoys higher time efficiency in comparison with the other two algorithms. In fact, the number of iterations of our method is around two times the numbers of iterations of the reference methods, while their computing time for every iteration is around 7 times of ours. This corresponds with the analysis in section IV-B.

Considering that the bounds on the parameters are not the tightest in our paper and the two references [26], [28], the parameters chosen in the above experiment may not be the best for the three algorithms. Therefore, we scan the parameters to find the best ones for every algorithm. For our algorithm, the best parameters found are $L_x = 1$, $L_y = 1$, and $\beta = 0.5$. For Ref1, the best parameter is $\beta = 9.5$, and for Ref2 the best parameters are $L_y = 2$ and $\beta = 5.5$. We perform 1000 independent trials with the best parameters again, and the average CPU running time is shown in Table 3. We can see that our algorithm still enjoys higher time efficiency in comparison with the other two algorithms.

Define the maximum variable gap as follows

$$\max\left\{\|x^k - x^{k-1}\|, \|y^k - y^{k-1}\|, \|Ax^k - y^k\|\right\}. \tag{17}$$

The curves of the maximum variable gap during the iterations in one random trial are plotted in FIGURE 1 which displays that our algorithm converges with the fastest speed. Considering that the objective function in problem (14) is nonconvex and it may have more than one saddle point, it is interesting to see where the value of objective function converges to, so we plot its convergence curve in one random trail in FIGURE 2, where the parameters are set as the same as the ones in the first experiment. We can see that Ref2 and our algorithm converges to the same saddle point, while Ref1 converges to another saddle point with a higher objective value. We repeat the trial for 1000 times with both the theoretically chosen parameters and the best parameters and always observe the same phenomenon.

B. RECONSTRUCTION ERROR AND TIME COST

Now, let us consider a slightly different setting, where A is still a Gaussian random matrix in $\mathbb{R}^{M\times N}$ consisting of entries

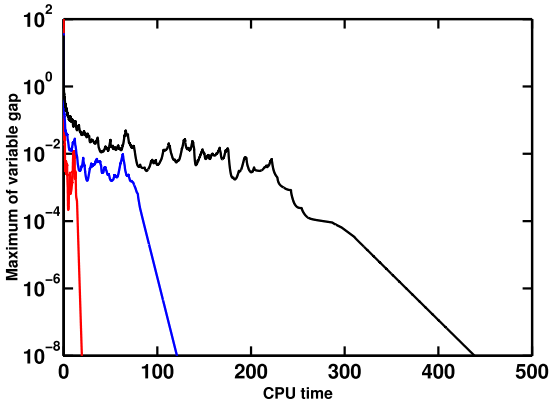

FIGURE 1. Convergence curves of the maximum variable gap. The red, blue, and black lines are our algorithm, Ref2, and Ref1, respectively.

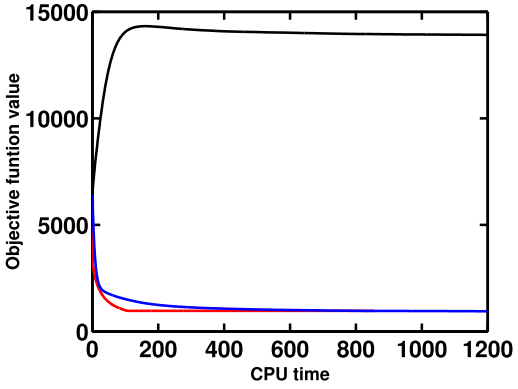

FIGURE 2. Convergence curve of the objective function value. The red, blue, and black lines are our algorithm, Ref2, and Ref1, respectively.

i.i.d. sampled from $\mathcal{N}(0, 1/N)$, but

$$b = Ax^* + z,$$

where x^* is an unknown k-sparse vector, and z is an unknown white noise consisting of entries i.i.d. sampled from $\mathcal{N}(0, \sigma^2/M)$ with $\sigma^2/\|Ax^*\|_2^2 = 10^{-3}$. In the experiment, nonzero positions of x^* are uniformly sampled from $[N]$, and nonzero entries are i.i.d. sampled from $\mathcal{N}(0, 1/k)$.

We apply ref1, ref2, and our algorithm to reconstruct x^* from A and b by solving the optimization problem defined in (15) with $\lambda = 10^{-3}$ and $\eta = 1$. The stopping criteria is set as

$$\max\left\{\|x^k - x^{k-1}\|, \|y^k - y^{k-1}\|, \|Ax^k - y^k\|\right\} < 10^{-6}. \tag{18}$$

From the experiment we have learned that the reconstruction error of these three algorithms is stable w.r.t. the choice of β, L_x, and L_y, so we continue to use the best parameters found in the previous experiment.

We perform 100 independent trials for each $k \in \{2, 4, 8, 16, 32, 64, 128, 256\}$. The results of the reconstruction errors defined as $\|\hat{x} - x^*\|_2/\|x^*\|_2$ are shown in Table 4. We can see when the sparsity of x^* is smaller than 256, i.e., when the algorithms success, their reconstruction errors are approximately the same, indicating that the linearization technique (or proximal operator) does not significantly change the evolving trajectory of the variables, when the optimization landscape is sufficiently benign and the linearized part is smooth enough. We also provide the average CPU running time in Table 5, which again demonstrates the high time efficiency of our algorithm.

TABLE 4. Average reconstruction error ($\times 10^{-3}$).

$\log_2 k$	1	2	3	4	5	6	7
Ref1	2.5	1.7	1.8	1.7	1.8	2.2	> 100
Ref2	2.5	1.7	1.8	1.7	1.8	2.2	> 100
Our algorithm	2.5	1.7	1.8	1.7	1.8	2.2	> 100

TABLE 5. Average CPU running time (seconds).

$\log_2 k$	1	2	3	4	5	6	7
Ref1	0.98	1.3	1.5	1.7	2.0	2.7	30
Ref2	6.5	8.9	13	21	35	79	349
Our algorithm	0.08	0.10	0.14	0.21	0.35	0.81	3.2

VI. CONCLUSION

In this work we study linearized ADMM algorithms for nonconvex optimization problems with nonconvex nonsmooth objective function. We propose a two-block linearized ADMM algorithm that introduces linearization for both the differentiable part in the objective and the augmented term, and provide theoretical convergence analysis under Assumption 1. Then we extend it to a multi-block parallel ADMM algorithm which can update coupled variables in parallel and render subproblems easier to solve, and the convergence analysis is still applicable. By arguing that Assumption 1 is not only plausible, but also relatively broad compared with other recent works on ADMM for nonconvex optimization, we show that the algorithms and their convergence analyses are general enough to work for many interesting problems such as some sparsity related problems and integer programming. In the numerical experiments, we show that the proposed algorithm enjoys higher time efficiency than the reference methods do, with parameters chosen according to theoretical bounds and best values obtained by scanning.

VII. APPENDIX

In this section, all notations x^k, y^k, and γ^k refer to the ones in Algorithm 1 and L_β refers to the augmented Lagrangian function defined in (5).

Lemma 1: Suppose we have a differentiable function f_1, a possibly nondifferentiable function f_2, and a point x. If there exists $d_2 \in \partial f_2(x)$, then we have

$$d = d_2 - \nabla f_1(x) \in \partial\left(f_2(x) - f_1(x)\right).$$

Proof: Firstly, by the definition of regular subgradient, we have

$$f_2(y) \geq f_2(x) + \langle d_2, y - x\rangle + o(\|y - x\|). \tag{19}$$

Secondly, because function f_1 is differentiable, we have

$$-f_1(y) = -f_1(x) - \langle \nabla f_1(x), y - x\rangle + o(\|y - x\|). \quad (20)$$

Adding (20) to (19), we get

$$f_2(y) - f_1(y) \geq f_2(x) - f_1(x) + \langle d_2 - \nabla f_1(x), y - x\rangle + o(\|y - x\|),$$

which together with the definition of regular subgradient leads to the conclusion. ∎

Lemma 2: If $h(y)$ is L_h-Lipschitz differentiable, then

$$h(y_2) - h(y_1) \geq \nabla h(s) \cdot (y_2 - y_1) - \frac{L_h}{2}\|y_2 - y_1\|^2, \quad (21)$$

where s denotes y_1 or y_2.

Proof:

$$\begin{aligned} &h(y_2) - h(y_1) \\ &= \int_0^1 \nabla h(ty_2 + (1-t)y_1) \cdot (y_2 - y_1)dt \\ &= \int_0^1 \nabla h(s) \cdot (y_2 - y_1)dt \\ &\quad + \int_0^1 \big(\nabla h(ty_2 + (1-t)y_1) - \nabla h(s)\big) \cdot (y_2 - y_1)dt, \end{aligned}$$

where $\nabla h(\cdot)$ defines the gradient of $h(\cdot)$. If we take $s = y_1$, then by inequality

$$\|\nabla h(ty_2 + (1-t)y_1) - \nabla h(y_1)\| \leq L_h\|t(y_2 - y_1)\|$$

we have

$$\begin{aligned} &\int_0^1 \nabla h(y_1) \cdot (y_2 - y_1)dt \\ &\quad + \int_0^1 \big(\nabla h(ty_2 + (1-t)y_1) - \nabla h(y_1)\big) \cdot (y_2 - y_1)dt \\ &\geq \nabla h(y_1) \cdot (y_2 - y_1) - \int_0^1 L_h t\|y_2 - y_1\|^2 dt \\ &= \nabla h(y_1) \cdot (y_2 - y_1) - \frac{L_h}{2}\|y_2 - y_1\|^2. \end{aligned}$$

Therefore, we get

$$h(y_2) - h(y_1) \geq \nabla h(y_1) \cdot (y_2 - y_1) - \frac{L_h}{2}\|y_2 - y_1\|^2.$$

Similarly, if we take $s = y_2$, we can get

$$h(y_2) - h(y_1) \geq \nabla h(y_2) \cdot (y_2 - y_1) - \frac{L_h}{2}\|y_2 - y_1\|^2.$$

∎

Lemma 3: Under Assumption 1, for any $l > k$, we have

$$\|\gamma^l - \gamma^k\|^2 \leq \frac{1}{\lambda_{B^TB}}\|B^T(\gamma^l - \gamma^k)\|^2,$$

where λ_{B^TB} is the smallest eigenvalue of B^TB.

Proof: By the γ-updating rule and the assumption $Im(A) \subset Im(B)$, for two integers $l > k$, we have

$$\gamma^l - \gamma^k = \sum_{i=k+1}^{l} \beta(Ax^i + By^i) \in Im(B).$$

Because $B \in \mathbb{R}^{n\times q}$ has full column rank, there exists $R \in \mathbb{R}^{q\times q}$, $Q \in \mathbb{R}^{q\times n}$ such that R is invertible, $QQ^T = I_{n\times n}$, and $B^T = RQ$. Noticing that $Im(B) = Im(Q^T)$, we get $\gamma^l - \gamma^k \in Im(Q^T)$. Thus, $\|\gamma^l - \gamma^k\|^2 = \|Q(\gamma^l - \gamma^k)\|^2$. Consequently, we have

$$\begin{aligned} \|B^T(\gamma^l - \gamma^k)\|^2 &= \|RQ(\gamma^l - \gamma^k)\|^2 \\ &\geq \lambda_{R^TR}\|Q(\gamma^l - \gamma^k)\|^2 \\ &= \lambda_{R^TR}\|\gamma^l - \gamma^k\|^2, \end{aligned}$$

where λ_{R^TR} denotes the minimum eigenvalue of R^TR.

By the definition of R and Q, we have $\lambda_{B^TB} = \lambda_{RR^T}$. Together with the common conclusion in linear algebra $\lambda_{R^TR} = \lambda_{RR^T}$, we get $\lambda_{R^TR} = \lambda_{B^TB}$, which completes the proof. ∎

Lemma 4: Under Assumption 1, the following equality holds for γ^{k+1}, y^k, and y^{k+1}

$$B^T\gamma^{k+1} = -\nabla_y g(x^{k+1}, y^k) - \nabla h(y^k) - L_y(y^{k+1} - y^k).$$

Proof: By calculating the derivative of $\bar{h}^k(y)$ defined in (8), we have

$$\begin{aligned} \nabla\bar{h}^k(y) = \nabla_y g(x^{k+1}, y^k) + \nabla h(y^k) + L_y(y - y^k) \\ + B^T\gamma^k + \beta B^T(Ax^{k+1} + By). \end{aligned}$$

Plug $y = y^{k+1}$ into it, and by the y-updating rule we have

$$\begin{aligned} &B^T\gamma^k + \beta B^T(Ax^{k+1} + By^{k+1}) \\ &\quad = -\nabla_y g(x^{k+1}, y^k) - \nabla h(y^k) - L_y(y^{k+1} - y^k). \quad (22) \end{aligned}$$

Besides, by the γ-updating rule, we have

$$B^T\gamma^{k+1} = B^T\gamma^k + \beta B^T(Ax^{k+1} + By^{k+1}). \quad (23)$$

By replacing the RHS of (23) with (22), we get

$$B^T\gamma^{k+1} = -\nabla_y g(x^{k+1}, y^k) - \nabla h(y^k) - L_y(y^{k+1} - y^k).$$

∎

Lemma 4 provides a way to express γ^{k+1} using y^k and y^{k+1}, which is a technique widely used in the convergence proof for nonconvex ADMM algorithms [26], [27].

Now we are ready to prove Theorem 1. We first give bounds on the descent or ascent of the Lagrangian function (2) after every update by using the quadratic form of the primal residual. Specifically, in the following, Lemma 5 presents that the descent of L_β is lower bounded after the x-updating step, Lemma 6 shows that the descent of L_β is lower bounded after the y-updating step, and Lemma 7 demonstrates that the ascent of L_β is upper bounded after the γ-updating step.

Lemma 5: Under Assumption 1, the following inequality holds for the update of x

$$L_\beta(x^k, y^k, \gamma^k) - L_\beta(x^{k+1}, y^k, \gamma^k) \geq C_0\|x^{k+1} - x^k\|^2,$$

where $C_0 = \frac{L_x - L_g - \beta L_{A^TA}}{2}$, and L_{A^TA} denotes the largest singular value of A^TA.

Proof: By x-updating rule in Algorithm 1, we have

$$\bar{f}^k(x^k) \geq \bar{f}^k(x^{k+1}). \quad (24)$$

Plugging the definition of $\bar{f}^k$ in (7) into (24), we get

$$f(x^k)-f(x^{k+1})+\langle x^k-x^{k+1},\beta A^T(Ax^k+By^k)+A^T\gamma^k\rangle \geq \langle x^{k+1}-x^k,\nabla_x g(x^k,y^k)\rangle+\tfrac{L_x}{2}\|x^{k+1}-x^k\|^2. \quad (25)$$

Then we have

$$\begin{aligned}
&L_\beta(x^k,y^k,\gamma^k)-L_\beta(x^{k+1},y^k,\gamma^k)\\
&=f(x^k)+g(x^k,y^k)-f(x^{k+1})-g(x^{k+1},y^k)\\
&\quad+\langle\gamma^k,Ax^k-Ax^{k+1}\rangle\\
&\quad+\frac{\beta}{2}\|Ax^k+By^k\|^2-\frac{\beta}{2}\|Ax^{k+1}+By^k\|^2\\
&=f(x^k)+g(x^k,y^k)-f(x^{k+1})-g(x^{k+1},y^k)\\
&\quad+\langle x^k-x^{k+1},A^T\gamma^k\rangle+\langle x^k-x^{k+1},\beta A^T(Ax^k+By^k)\rangle\\
&\quad-\frac{\beta}{2}\|A(x^{k+1}-x^k)\|^2\\
&\geq g(x^k,y^k)-g(x^{k+1},y^k)+\langle x^{k+1}-x^k,\nabla_x g(x^k,y^k)\rangle\\
&\quad+\frac{L_x}{2}\|x^{k+1}-x^k\|^2-\frac{\beta}{2}\|A(x^{k+1}-x^k)\|^2\\
&\geq\frac{L_x-L_g-\beta L_{A^TA}}{2}\|x^{k+1}-x^k\|^2,
\end{aligned}$$

where the last inequality is from Lemma 2 and L_{A^TA} denotes the largest singular value of A^TA. ■

Lemma 6: Under Assumption 1, the following inequality holds for the update of y

$$L_\beta(x^{k+1},y^k,\gamma^k)-L_\beta(x^{k+1},y^{k+1},\gamma^k)\geq C_1\|y^k-y^{k+1}\|^2,$$

where $C_1=\frac{2L_y-L_\omega}{2}$ and $L_\omega=L_g+L_h$.

Proof: According to that $\bar{h}^k(y)$ is L_y-convex, by Proposition 4.8 in [64] we have

$$\bar{h}^k(y^k)\geq\bar{h}^k(y^{k+1})+\left\langle y^k-y^{k+1},\nabla\bar{h}(y^{k+1})\right\rangle+\frac{L_y}{2}\|y^k-y^{k+1}\|^2.$$

According to the updating rule of y, i.e., $\nabla\bar{h}^k(y^{k+1})=0$, the above inequality is reshaped to

$$\bar{h}^k(y^k)\geq\bar{h}^k(y^{k+1})+\frac{L_y}{2}\|y^k-y^{k+1}\|^2. \quad (26)$$

Denote $w^k(y)=g(x^k,y)+h(y)$ and recall that $g(x,y)$ and $h(y)$ are L_g and L_h Lipschitz-differentiable, respectively. We get that $w^k(y)$ is L_w Lipschitz-differentiable, where $L_w=L_g+L_h$. Then by Lemma 2 we have

$$w^{k+1}(y^k)\geq w^{k+1}(y^{k+1})+\left\langle y^k-y^{k+1},\nabla w^{k+1}(y^k)\right\rangle-\frac{L_w}{2}\|y^k-y^{k+1}\|^2. \quad (27)$$

Now we consider the descent of L_β in y-updating step.

$$\begin{aligned}
&L_\beta(x^{k+1},y^k,\gamma^k)-L_\beta(x^{k+1},y^{k+1},\gamma^k)\\
&=w^{k+1}(y^k)-w^{k+1}(y^{k+1})+\left\langle\gamma^k,B(y^k-y^{k+1})\right\rangle \quad (28)\\
&\quad+\frac{\beta}{2}\|Ax^{k+1}+By^k\|^2-\frac{\beta}{2}\|Ax^{k+1}+By^{k+1}\|^2. \quad (29)
\end{aligned}$$

By plugging (27) into (29), we have

$$\begin{aligned}
&\text{RHS of (29)} \quad (30)\\
&\geq\left\langle y^k-y^{k+1},\nabla w^{k+1}(y^k)\right\rangle+\langle\gamma^k,B(y^k-y^{k+1})\rangle\\
&\quad-\frac{L_w}{2}\|y^k-y^{k+1}\|^2+\frac{\beta}{2}\|Ax^{k+1}+By^k\|^2\\
&\quad-\frac{\beta}{2}\|Ax^{k+1}+By^{k+1}\|^2. \quad (31)
\end{aligned}$$

By the definition of $\bar{h}^k(y)$ in (8), we further derive

$$\text{RHS of(31)}=\bar{h}^k(y^k)-\bar{h}^k(y^{k+1})+\frac{L_y-L_w}{2}\|y^k-y^{k+1}\|^2. \quad (32)$$

By inserting (26) into (32), we finally reach

$$L_\beta(x^{k+1},y^k,\gamma^k)-L_\beta(x^{k+1},y^{k+1},\gamma^k)\geq C_1\|y^k-y^{k+1}\|^2,$$

where

$$C_1:=\frac{2L_y-L_w}{2}. \quad (33)$$

■

Lemma 7: Under Assumption 1, the following inequality holds for the update of γ

$$\begin{aligned}
&L_\beta(x^{k+1},y^{k+1},\gamma^{k+1})-L_\beta(x^{k+1},y^{k+1},\gamma^k)\\
&=\frac{1}{\beta}\|\gamma^{k+1}-\gamma^k\|^2\\
&\leq C_2\|x^{k+1}-x^k\|^2+C_3\|y^{k+1}-y^k\|^2+C_4\|y^k-y^{k-1}\|^2,
\end{aligned} \quad (34)$$

where $L_w=L_g+L_h$, $C_2=\frac{3L_w^2}{\beta\lambda_{B^TB}}$, $C_3=\frac{3L_y^2}{\beta\lambda_{B^TB}}$ and $C_4=\frac{3(L_w^2+L_y^2)}{\beta\lambda_{B^TB}}$.

Proof: By definition, the ascent of L_β after the $(k+1)$th iteration of γ is

$$L_\beta(x^{k+1},y^{k+1},\gamma^{k+1})-L_\beta(x^{k+1},y^{k+1},\gamma^k)=\left\langle\gamma^{k+1}-\gamma^k,Ax^{k+1}+By^{k+1}\right\rangle. \quad (35)$$

By inserting the γ-updating rule in (35) and applying Lemma 3, we have

$$\begin{aligned}
&L_\beta(x^{k+1},y^{k+1},\gamma^{k+1})-L_\beta(x^{k+1},y^{k+1},\gamma^k)\\
&=\frac{1}{\beta}\|\gamma^{k+1}-\gamma^k\|^2\\
&\leq\frac{1}{\beta\lambda_{B^TB}}\|B^T(\gamma^{k+1}-\gamma^k)\|^2,
\end{aligned} \quad (36)$$

where λ_{B^TB} denotes the smallest singular value of B^TB. By Lemma 4 and AM-GM Inequality we have

$$\begin{aligned}
&\|B^T(\gamma^{k+1}-\gamma^k)\|^2\\
&=\left\|\nabla w^{k+1}(y^k)+L_y(y^{k+1}-y^k)-\nabla w^k(y^{k-1})\right.\\
&\quad\left.-L_y(y^k-y^{k-1})\right\|^2
\end{aligned} \quad (37)$$

$$\leq 3\Big(\|\nabla w^{k+1}(y^k)-\nabla w^k(y^{k-1})\|^2+L_y^2\|y^{k+1}-y^k\|^2 +L_y^2\|y^k-y^{k-1}\|^2\Big), \tag{38}$$

where $w^k(y)=g(x^k,y)+h(y)$ has been defined in the proof of Lemma 6.

Because $g(x,y)+h(y)$ is L_w Lipschitz differentiable, we have

$$\begin{aligned}&\|\nabla w^{k+1}(y^k)-\nabla w^k(y^{k-1})\|^2\\ &=\|\nabla_y g(x^{k+1},y^k)+\nabla h(y^k)-\nabla_y g(x^k,y^{k-1})-\nabla h(y^{k-1})\|^2\\ &\leq L_w^2\Big(\|x^{k+1}-x^k\|^2+\|y^k-y^{k-1}\|^2\Big),\end{aligned}$$

and together with (36) and (38) we have

$$\begin{aligned}&L_\beta(x^{k+1},y^{k+1},\gamma^{k+1})-L_\beta(x^{k+1},y^{k+1},\gamma^k)\\ &\leq C_2\|x^{k+1}-x^k\|^2+C_3\|y^{k+1}-y^k\|^2+C_4\|y^k-y^{k-1}\|^2,\end{aligned}$$

where

$$C_2:=\frac{3L_w^2}{\beta\lambda_{B^TB}}, \tag{39}$$

$$C_3:=\frac{3L_y^2}{\beta\lambda_{B^TB}}, \tag{40}$$

$$C_4:=\frac{3(L_w^2+L_y^2)}{\beta\lambda_{B^TB}}. \tag{41}$$

■

Then we design a sequence $\{m_k\}_{k=1}^{+\infty}$ by

$$m_k=L_\beta(x^k,y^k,\gamma^k)+C_m\|y^k-y^{k-1}\|^2, \tag{42}$$

where C_m is set according to (9) in Theorem 1. We will first prove the convergence of $\{m_k\}_{k=1}^{+\infty}$ and then prove the convergence of $\{L_\beta(x^k,y^k,\gamma^k)\}$.

Lemma 8: For the linearized ADMM in Algorithm 1, under Assumption 1, if we choose the parameters L_x, L_y and β satisfying (9), then the sequence $\{m_k\}$ defined in (42) is convergent.

Proof:

1) Monotonicity of $\{m_k\}$

By using Lemma 5, Lemma 6, and Lemma 7, we have

$$\begin{aligned}&L_\beta(x^k,y^k,\gamma^k)-L_\beta(x^{k+1},y^{k+1},\gamma^{k+1})\\ &\geq L_\beta(x^{k+1},y^k,\gamma^k)-L_\beta(x^{k+1},y^{k+1},\gamma^{k+1})\\ &\quad+C_0\|x^{k+1}-x^k\|^2\\ &\geq L_\beta(x^{k+1},y^{k+1},\gamma^k)-L_\beta(x^{k+1},y^{k+1},\gamma^{k+1})\\ &\quad+C_1\|y^{k+1}-y^k\|^2+C_0\|x^{k+1}-x^k\|^2\\ &\geq (C_1-C_3)\|y^{k+1}-y^k\|^2-C_4\|y^k-y^{k-1}\|^2\\ &\quad+(C_0-C_2)\|x^{k+1}-x^k\|^2.\end{aligned} \tag{43}$$

By combining (43) with the definition of m_k, we have

$$\begin{aligned}m_k-m_{k+1}\geq{}&(C_1-C_3-C_m)\|y^{k+1}-y^k\|^2\\ &+(C_m-C_4)\|y^k-y^{k-1}\|^2\\ &+(C_0-C_2)\|x^{k+1}-x^k\|^2.\end{aligned} \tag{44}$$

Recall the definition of C_0, C_1, C_3, C_4 and the parameters L_x, L_y, C_m, β we choose in (9), we get

$$C_0-C_2=\frac{1}{2}(L_x-L_g-\beta L_{A^TA}-\frac{6L_w^2}{\beta\lambda_{B^TB}})\geq\frac{1}{2}, \tag{45}$$

$$C_1-C_3-C_m=\frac{2L_y-L_w}{2}-\frac{3L_y^2}{\beta\lambda_{B^TB}}-C_m\geq\frac{1}{2}, \tag{46}$$

$$C_m-C_4=C_m-\frac{3(L_w^2+L_y^2)}{\beta\lambda_{B^TB}}>0. \tag{47}$$

Therefore, $\{m_k\}$ is monotonically decreasing.

2) Lower bound of $\{m_k\}$

Next we will argue that $\{m_k\}$ is also lower bounded. By the assumption $Im(A)\subset Im(B)$, there exists y_k' such that $By_k'=-Ax^k$, so we have

$$\begin{aligned}m_k={}&g(x^k,y^k)+f(x^k)+h(y^k)+\langle\gamma^k,B(y^k-y_k')\rangle\\ &+\frac{\beta}{2}\|B(y^k-y_k')\|^2+C_m\|y^k-y^{k-1}\|^2.\end{aligned} \tag{48}$$

By applying Lemma 4 to the third item in the RHS of (48), we have

$$\begin{aligned}&\langle\gamma^k,B(y^k-y_k')\rangle\\ &=\langle B^T\gamma^k,y^k-y_k'\rangle\\ &=\langle-\nabla w^k(y^{k-1})-L_y(y^k-y^{k-1}),y^k-y_k'\rangle\\ &=\big\langle\nabla w^k(y^k)-\nabla w^k(y^{k-1})-L_y(y^k-y^{k-1}),y^k-y_k'\big\rangle\\ &\quad-\langle\nabla w^k(y^k),y^k-y_k'\rangle.\end{aligned} \tag{49}$$

By AM-GM Inequality, we bound the first item in the RHS of (49)

$$\begin{aligned}&\langle\nabla w^k(y^k)-\nabla w^k(y^{k-1})-L_y(y^k-y^{k-1}),y^k-y_k'\rangle\\ &=\langle\nabla w^k(y^k)-\nabla w^k(y^{k-1}),y^k-y_k'\rangle\\ &\quad-L_y\langle y^k-y^{k-1},y^k-y_k'\rangle\\ &\geq-\frac{1}{2}\Big(\|\nabla w^k(y^k)-\nabla w^k(y^{k-1})\|^2+\|y^k-y_k'\|^2\Big)\\ &\quad-\frac{L_y}{2}\Big(\|y^k-y^{k-1}\|^2+\|y^k-y_k'\|^2\Big)\\ &\geq-\frac{1}{2}\Big((L_w^2+L_y)\|y^k-y^{k-1}\|^2+(L_y+1)\|y^k-y_k'\|^2\Big),\end{aligned} \tag{50}$$

where the last inequality is from the Lipschitz differentiability of $w^k(y)$.

Considering that B has full rank and $\|Bz\|^2\geq\lambda_{B^TB}\|z\|^2$, for all z, the fourth item in the RHS of (48) can be bounded by

$$\|B(y^k-y_k')\|^2\geq\lambda_{B^TB}\|y^k-y_k'\|^2, \tag{51}$$

By plugging (49), (50), and (51) into (48), we get

$$m_k\geq Q_1^k+Q_2^k,$$

where

$$Q_1^k:=g(x^k,y^k)+f(x^k)+h(y^k)-\langle\nabla w^k(y^k),y^k-y_k'\rangle$$

$$+\frac{1}{2}\left(\beta\lambda_{B^TB}-L_y-1\right)\|y^k-y'_k\|^2,$$

$$Q_2^k:=\left(C_m-\frac{L_y}{2}-\frac{L_w^2}{2}\right)\|y^k-y^{k-1}\|^2.$$

If both Q_1^k and Q_2^k are lower bounded, the proof will be completed. Let us first check Q_2^k. Recall the C_m and L_y we choose in (9), we get

$$Q_2^k=\left(C_m-\frac{L_y}{2}-\frac{L_w^2}{2}\right)\|y^k-y^{k-1}\|^2=0. \tag{52}$$

For Q_1^k, recall the β and L_y we choose in (9) and we get

$$\beta\lambda_{B^TB}\geq L_w+L_y+2, \tag{53}$$

then by Lemma 2 we have

$$\begin{aligned}Q_1^k&\geq g(x^k,y^k)+f(x^k)+h(y^k)-\langle\nabla w^k(y^k),y^k-y'_k\rangle\\&\quad+\frac{L_w}{2}\|y^k-y'_k\|^2+\frac{1}{2}\|y^k-y'_k\|^2\\&\geq g(x^k,y'_k)+f(x^k)+h(y'_k)+\frac{1}{2}\|y^k-y'_k\|^2,\end{aligned}$$

where $g(x^k,y'_k)+f(x^k)+h(y'_k)$ is lower bounded, because (x^k,y'_k) belongs to the feasible set. Therefore, $\{m_k\}$ is lower bounded. Together with its monotonic decrease, we get $\{m_k\}$ is convergent. ■

A. PROOF OF THEOREM 1

Recall in Lemma 8, we first prove that $\{m_k\}$ is monotonically decreasing by

$$\begin{aligned}m_k-m_{k+1}\geq{}&(C_1-C_3-C_m)\|y^{k+1}-y^k\|^2\\&+(C_m-C_4)\|y^k-y^{k-1}\|^2\\&+(C_0-C_2)\|x^{k+1}-x^k\|^2,\end{aligned}$$

and then prove that $\{m_k\}$ is lower bounded by

$$m_k\geq g(x^k,y'_k)+f(x^k)+h(y'_k)+\frac{1}{2}\|y^k-y'_k\|^2, \tag{54}$$

where y'_k is defined by $By'_k=-Ax^k$. Notice that y'_k always exists because of the assumption $Im(A)\subset Im(B)$.

By the convergence of $\{m_k\}$, $\|x^{k+1}-x^k\|$ and $\|y^{k+1}-y^k\|$ converges to zero. By the definition of $\{m_k\}$ and its convergence, we readily get the convergence of $L_\beta(x^k,y^k,\gamma^k)$. According to Lemma 7, $\|\gamma^{k+1}-\gamma^k\|$ converges to zero as well.

B. PROOF OF COROLLARY 1

Recall (54) in the proof of Lemma 8. Because $g(x,y)+f(x)+h(y)$ is coercive over the feasible set with respect to y, if $\{y'_k\}$ diverges, then the RHS of (54) diverges to positive infinity, which contradicts with the convergence of $\{m_k\}$.

Because of the term $\frac{1}{2}\|y^k-y'_k\|^2$ on the RHS of (54), the boundedness of $\{y^k\}$ can be derived from the boundedness of $\{y'_k\}$.

In order to prove that $\{\gamma^k\}$ is bounded, we only need to prove $\{\gamma^k-\gamma^0\}$ is bounded. By Lemma 3, it is equivalent to the boundedness of $\{B^T(\gamma^k-\gamma^0)\}$ and further equivalent to the boundedness of $\{B^T\gamma^k\}$. When function $g(x,y)$ degenerates to $g(x)$, by Lemma 4, we get

$$B^T\gamma^{k+1}=-\nabla h(y^k)-L_y(y^{k+1}-y^k),$$

which implies that the boundedness of $\{B^T\gamma^k\}$ can be deduced from the boundedness of $\{y^k\}$.

C. PROOF OF THEOREM 2

1) LIMIT OF $\nabla_\gamma L_\beta$

When k approaches infinity, we have

$$\begin{aligned}\nabla_\gamma L_\beta(x^{k+1},y^{k+1},\gamma^{k+1})&=Ax^{k+1}+By^{k+1}\\&=\frac{1}{\beta}(\gamma^{k+1}-\gamma^k)\to 0.\end{aligned}$$

2) LIMIT OF $\nabla_y L_\beta$

By Theorem 1 and Lemma 4, when k approaches infinity, we have

$$\begin{aligned}\nabla_y L_\beta(x^k,y^k,\gamma^k)&=\nabla_y g(x^k,y^k)+\nabla h(y^k)+B^T\gamma^k\\&\quad+\beta B^T(Ax^k+By^k)\\&\to\nabla_y g(x^k,y^{k-1})+\nabla h(y^{k-1})\\&\quad+B^T\gamma^k+B^T(\gamma^k-\gamma^{k-1})\\&=-L_y(y^k-y^{k-1})+B^T(\gamma^k-\gamma^{k-1})\to 0.\end{aligned}$$

3) LIMIT OF $\partial_x L_\beta$

By x-updating rule, x^{k+1} is the minimum point of $\bar{f}^k(x)$, which implies $0\in\partial\bar{f}^k(x^{k+1})$. Therefore, by the definition of $\bar{f}^k$ in (7) and Lemma 1, there exists $d^{k+1}\in\partial f(x^{k+1})$ such that

$$\begin{aligned}&\nabla_x g(x^k,y^k)+d^{k+1}+A^T\gamma^k\\&\quad+\beta A^T(Ax^k+By^k)+L_x(x^{k+1}-x^k)=\mathbf{0}.\end{aligned} \tag{55}$$

We further define

$$\begin{aligned}\bar{d}^{k+1}:={}&\nabla_x g(x^{k+1},y^{k+1})+d^{k+1}+A^T\gamma^{k+1}\\&+\beta A^T\left(Ax^{k+1}+By^{k+1}\right),\end{aligned} \tag{56}$$

which, one may readily check, satisfies

$$\bar{d}^{k+1}\in\partial_x L_\beta(x^{k+1},y^{k+1},\gamma^{k+1}).$$

By Theorem 1, we have that the primal residues $\|y^{k+1}-y^k\|$, $\|x^k-x^{k+1}\|$ and dual residue $\|\gamma^{k+1}-\gamma^k\|$ converge to zero as k approaches infinity, therefore

$$\begin{aligned}&\lim_{k\to+\infty}\bar{d}^{k+1}\\&=\lim_{k\to+\infty}\left[\nabla_x g(x^{k+1},y^{k+1})+d^{k+1}+A^T\gamma^{k+1}\right.\\&\qquad\left.+\beta A^T(Ax^{k+1}+By^{k+1})\right]\\&=\lim_{k\to+\infty}\left[\nabla_x g(x^k,y^k)+d^{k+1}+A^T\gamma^k\right.\\&\qquad\left.+\beta A^T(Ax^k+By^k)+L_x(x^{k+1}-x^k)\right]=0,\end{aligned}$$

where the last equality is from (55).

D. PROOF OF COROLLARY 2

As k tends to infinity, by Corollary 1 and Theorem 2, we have that γ^k is bounded and $Ax^k + By^k \to 0$. Then we have that

$$\begin{aligned} f(x^k) + h(y^k) &= L_\beta(x^k, y^k, \gamma^k) - \langle \gamma^k, Ax^k + By^k \rangle \\ &\quad - \frac{\beta}{2}\|Ax^k + By^k\|^2 \\ &\to L_\beta(x^k, y^k, \gamma^k) - 0 - 0 = L_\beta(x^k, y^k, \gamma^k). \end{aligned}$$

Therefore, the value of objective function will converge, because L_β will converge.

REFERENCES

[1] M. Zibulevsky and M. Elad, "ℓ_1-ℓ_2 optimization in signal and image processing," *IEEE Signal Process. Mag.*, vol. 27, no. 3, pp. 76–88, May 2010.

[2] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: A review," *ACM Comput. Surv.*, vol. 31, no. 3, pp. 264–323, Sep. 1999.

[3] Z.-Q. Luo and W. Yu, "An introduction to convex optimization for communications and signal processing," *IEEE J. Sel. Areas Commun.*, vol. 24, no. 8, pp. 1426–1438, Aug. 2006.

[4] D. L. Donoho, "Compressed sensing," *IEEE Trans. Inf. Theory*, vol. 52, no. 4, pp. 1289–1306, Apr. 2006.

[5] Q. Zhang and B. Li, "Discriminative K-SVD for dictionary learning in face recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2010, pp. 2691–2698.

[6] J. J. Xiao, S. Cui, Z. Q. Luo, and A. J. Goldsmith, "Linear coherent decentralized estimation," *IEEE Trans. Signal Process.*, vol. 56, no. 2, pp. 757–770, Feb. 2008.

[7] D. P. Bertsekas, *Nonlinear Programming*. Belmont, WA, Australia: Athena Scientific, 1999.

[8] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*, vol. 23. Englewood Cliffs, NJ, USA: Prentice-Hall, 1989.

[9] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[10] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2001, pp. 556–562.

[11] R. Gemulla, E. Nijkamp, P. J. Haas, and Y. Sismanis, "Large-scale matrix factorization with distributed stochastic gradient descent," in *Proc. 17th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2011, pp. 69–77.

[12] E. J. Candes, Y. C. Eldar, T. Strohmer, and V. Voroninski, "Phase retrieval via matrix completion," *SIAM Rev.*, vol. 57, no. 2, pp. 225–251, 2015.

[13] E. Januzaj, H.-P. Kriegel, and M. Pfeifle, "Scalable density-based distributed clustering," in *Proc. Eur. Conf. Princ. Data Mining Knowl. Discovery*. Berlin, Germany: Springer, 2004, pp. 231–244.

[14] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, Jan. 2011.

[15] K. Scheinberg, S. Ma, and D. Goldfarb, "Sparse inverse covariance selection via alternating linearization methods," in *Proc. Adv. Neural Inf. Process. Syst.*, 2010, pp. 2101–2109.

[16] W. Shi, Q. Ling, K. Yuan, G. Wu, and W. Yin, "On the linear convergence of the ADMM in decentralized consensus optimization," *IEEE Trans. Signal Process.*, vol. 62, no. 7, pp. 1750–1761, Apr. 2014.

[17] W. Yin, S. Osher, D. Goldfarb, and J. Darbon, "Bregman iterative algorithms for ℓ_1-minimization with applications to compressed sensing," *SIAM J. Imag. Sci.*, vol. 1, no. 1, pp. 143–168, 2008.

[18] X. Zhang, M. Burger, and S. Osher, "A unified primal-dual algorithm framework based on Bregman iteration," *J. Sci. Comput.*, vol. 46, no. 1, pp. 20–46, 2011.

[19] C. Feng, H. Xu, and B. Li, "An alternating direction method approach to cloud traffic management," *IEEE Trans. Parallel Distrib. Syst.*, vol. 28, no. 8, pp. 2145–2158, Aug. 2017.

[20] W. Deng, M.-J. Lai, Z. Peng, and W. Yin, "Parallel multi-block ADMM with $o(1/k)$ convergence," *J. Sci. Comput.*, vol. 71, no. 2, pp. 712–736, May 2017.

[21] T.-Y. Lin, S.-Q. Ma, and S.-Z. Zhang, "On the sublinear convergence rate of multi-block ADMM," *J. Oper. Res. Soc. China*, vol. 3, no. 3, pp. 251–274, Sep. 2015.

[22] C. Chen, B. He, Y. Ye, and X. Yuan, "The direct extension of ADMM for multi-block convex minimization problems is not necessarily convergent," *Math. Program.*, vol. 155, nos. 1–2, pp. 57–79, 2016.

[23] R. Glowinski and A. Marrocco, "Sur l'approximation, par éléments finis d'ordre un, et la résolution, par pénalisation-dualité d'une classe de problèmes de Dirichlet non linéaires," *Rev. Française d'autom., Inf., Recherche Opér.*, vol. 9, no. 2, pp. 41–76, 1975.

[24] X. Shen, L. Chen, Y. Gu, and H.-C. So, "Square-root Lasso with nonconvex regularization: An ADMM approach," *IEEE Signal Process. Lett.*, vol. 23, no. 7, pp. 934–938, Jul. 2016.

[25] L. Chen and Y. Gu, "The convergence guarantees of a non-convex approach for sparse recovery," *IEEE Trans. Signal Process.*, vol. 62, no. 15, pp. 3754–3767, Aug. 2014.

[26] Y. Wang, W. Yin, and J. Zeng, "Global convergence of ADMM in nonconvex nonsmooth optimization," *J. Sci. Comput.*, vol. 78, no. 1, pp. 29–63, Jan. 2019.

[27] M. Hong, Z.-Q. Luo, and M. Razaviyayn, "Convergence analysis of alternating direction method of multipliers for a family of nonconvex problems," *SIAM J. Optim.*, vol. 26, no. 1, pp. 337–364, 2016.

[28] B. Jiang, T. Lin, S. Ma, and S. Zhang, "Structured nonconvex and nonsmooth optimization: Algorithms and iteration complexity analysis," *Comput. Optim. Appl.*, vol. 72, no. 1, pp. 115–157, Jan. 2019.

[29] K. Guo, D. R. Han, and T. T. Wu, "Convergence of alternating direction method for minimizing sum of two nonconvex functions with linear constraints," *Int. J. Comput. Math.*, vol. 94, no. 8, pp. 1653–1669, 2017.

[30] L. Yang, T. K. Pong, and X. Chen, "Alternating direction method of multipliers for a class of nonconvex and nonsmooth problems with applications to background/foreground extraction," *SIAM J. Imag. Sci.*, vol. 10, no. 1, pp. 74–110, 2017.

[31] G. Li and T. K. Pong, "Global convergence of splitting methods for nonconvex composite optimization," *SIAM J. Optim.*, vol. 25, no. 4, pp. 2434–2460, 2015.

[32] T. Lin, S. Ma, and S. Zhang, "An extragradient-based alternating direction method for convex minimization," *Found. Comput. Math.*, vol. 17, no. 1, pp. 35–59, Feb. 2017.

[33] Y. Ouyang, Y. Chen, G. Lan, and E. Pasiliao, Jr., "An accelerated linearized alternating direction method of multipliers," *SIAM J. Imag. Sci.*, vol. 8, no. 1, pp. 644–681, 2015.

[34] R. H. Chan, M. Tao, and X. Yuan, "Linearized alternating direction method of multipliers for constrained linear least-squares problem," *East Asian J. Appl. Math.*, vol. 2, no. 4, pp. 326–341, 2012.

[35] Z. Lin, R. Liu, and Z. Su, "Linearized alternating direction method with adaptive penalty for low-rank representation," in *Proc. Adv. Neural Inf. Process. Syst.*, 2011, pp. 612–620.

[36] Q. Ling, W. Shi, G. Wu, and A. Ribeiro, "DLM: Decentralized linearized alternating direction method of multipliers," *IEEE Trans. Signal Process.*, vol. 63, no. 15, pp. 4051–4064, Aug. 2015.

[37] Z.-Z. Yang and Z. Yang, "Fast linearized alternating direction method of multipliers for the augmented ℓ_1-regularized problem," *Signal, Image Video Process.*, vol. 9, no. 7, pp. 1601–1612, Oct. 2015.

[38] R. Gu, A. Dogndžić, "A fast proximal gradient algorithm for reconstructing nonnegative signals with sparse transform coefficients," in *Proc. 48th Asilomar Conf. Signals, Syst. Comput.*, Nov. 2014, pp. 1662–1667.

[39] J. Yang and X. Yuan, "Linearized augmented Lagrangian and alternating direction methods for nuclear norm minimization," *Math. Comput.*, vol. 82, no. 281, pp. 301–329, 2012.

[40] T. Jeong, H. Woo, and S. Yun, "Frame-based Poisson image restoration using a proximal linearized alternating direction method," *Inverse Problems*, vol. 29, no. 7, 2013, Art. no. 075007.

[41] H. Nien and J. A. Fessler, "Fast X-ray CT image reconstruction using a linearized augmented lagrangian method with ordered subsets," *IEEE Trans. Med. Imag.*, vol. 34, no. 2, pp. 388–399, Feb. 2015.

[42] H. Woo and S. Yun, "Proximal linearized alternating direction method for multiplicative denoising," *SIAM J. Sci. Comput.*, vol. 35, no. 2, pp. B336–B358, 2013.

[43] M. K. Ng, F. Wang, and X. Yuan, "Inexact alternating direction methods for image recovery," *SIAM J. Sci. Comput.*, vol. 33, no. 4, pp. 1643–1668, 2011.

[44] O. Y. Al-Jarrah, P. D. Yoo, S. Muhaidat, G. K. Karagiannidis, and K. Taha, "Efficient machine learning for big data: A review," *Big Data Res.*, vol. 2, no. 3, pp. 87–93, 2015.

[45] G. B. Giannakis, F. Bach, R. Cendrillon, M. Mahoney, and J. Neville, "Signal processing for big data [from the guest editors]," *IEEE Signal Process. Mag.*, vol. 31, no. 5, pp. 15–16, Sep. 2014.
[46] G. Scutari, F. Facchinei, L. Lampariello, and P. Song. (2014). "Parallel and distributed methods for nonconvex optimization—Part I: Theory." [Online]. Available: https://arxiv.org/abs/1410.4754
[47] G. Scutari, F. Facchinei, L. Lampariello, P. Song, and S. Sardellitti. (2016). "Parallel and distributed methods for nonconvex optimization—Part II: applications." [Online]. Available: https://arxiv.org/abs/1601.04059
[48] B. He, H.-K. Xu, and X. Yuan, "On the proximal Jacobian decomposition of ALM for multiple-block separable convex minimization problems and its relationship to ADMM," *J. Sci. Comput.*, vol. 66, no. 3, pp. 1204–1217, Mar. 2016.
[49] H. Wang, A. Banerjee, and Z.-Q. Luo, "Parallel direction method of multipliers," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 181–189.
[50] T. H. Chang, M. Hong, and X. Wang, "Multi-agent distributed optimization via inexact consensus ADMM," *IEEE Trans. Signal Process.*, vol. 63, no. 2, pp. 482–497, Jan. 2015.
[51] K. Wang, J. Desai, and H. He, "A note on augmented Lagrangian-based parallel splitting method," *Optim. Lett.*, vol. 9, no. 6, pp. 1199–1212, 2015.
[52] Y. Hu, E. C. Chi, and G. I. Allen, "ADMM algorithmic regularization paths for sparse statistical machine learning," in *Splitting Methods in Communication, Imaging, Science, and Engineering*. Cham, Switzerland: Springer, 2016, pp. 433–459.
[53] W. Bo, S. Boyd, M. Annergren, and Y. Wang, "An ADMM algorithm for a class of total variation regularized estimation problems," *IFAC Proc.*, vol. 45, no. 16, pp. 83–88, 2012.
[54] F. Wang, W. Cao, and Z. Xu, "Convergence of multi-block Bregman ADMM for nonconvex composite problems," *Sci. China Inf. Sci.*, vol. 61, no. 12, 2018, Art. no. 122101.
[55] R. T. Rockafellar and R. J.-B. Wets, *Variational Analysis*, vol. 317. Berlin, Germany: Springer, 2009.
[56] N. Parikh and S. Boyd, "Proximal algorithms," *Found. Trends Optim.*, vol. 1, no. 3, pp. 127–239, Jan. 2014.
[57] A. Merchant and B. Sengupta, "Assignment of cells to switches in PCS networks," *IEEE/ACM Trans. Netw.*, vol. 3, no. 5, pp. 521–526, Oct. 1995.
[58] T. Li and M. Shahidehpour, "Price-based unit commitment: A case of Lagrangian relaxation versus mixed integer programming," *IEEE Trans. Power Syst.*, vol. 20, no. 4, pp. 2015–2025, Nov. 2005.
[59] F. Glover, "Future paths for integer programming and links to artificial intelligence," *Comput. Oper. Res.*, vol. 13, no. 5, pp. 533–549, 1986.
[60] L. Pallottino, E. M. Feron, and A. Bicchi, "Conflict resolution problems for air traffic management systems solved with mixed integer programming," *IEEE Trans. Intell. Transp. Syst.*, vol. 3, no. 1, pp. 3–11, Mar. 2002.
[61] G. Gasso, A. Rakotomamonjy, and S. Canu, "Recovering sparse signals with a certain family of nonconvex penalties and DC programming," *IEEE Trans. Signal Process.*, vol. 57, no. 12, pp. 4686–4698, Dec. 2009.
[62] R. Chartrand, "Exact reconstruction of sparse signals via nonconvex minimization," *IEEE Signal Process. Lett.*, vol. 14, no. 10, pp. 707–710, Oct. 2007.
[63] R. Saab and Ö. Yılmaz, "Sparse recovery by non-convex optimization—Instance optimality," *Appl. Comput. Harmon. Anal.*, vol. 29, no. 1, pp. 30–48, 2010.
[64] J.-P. Vial, "Strong and weak convexity of sets and functions," *Math. Oper. Res.*, vol. 8, no. 2, pp. 231–259, 1983.

Authors' photographs and biographies not available at the time of publication.