CONVERGENCE ANALYSIS OF ALTERNATING DIRECTION METHOD OF MULTIPLIERS FOR A FAMILY OF NONCONVEX PROBLEMS

Mingyi Hong[1], *Zhi-Quan Luo*[2] *and Meisam Razaviyayn*[3]

[1] Dept. of IMSE, Iowa State Univ., Ames, IA, 50011, USA

[2] Dept. of Elect. & Compt. Eng. Univ. of Minnesota, Minneapolis, 55454, USA

[3] Dept. of Elect. Eng. Stanford Univ., Stanford, 94305, USA

ABSTRACT

In this paper, we analyze the behavior of the alternating direction method of multipliers (ADMM), for solving a family of nonconvex problems. Our focus is given to the well-known *consensus* and *sharing* problems, both of which have wide applications in signal processing. We show that in the presence of nonconvex objective function, classical ADMM is able to reach the set of stationary solutions for these problems, if the stepsize is chosen large enough. An interesting consequence of our analysis is that the ADMM is convergent for a family of sharing problems, regardless of the number of blocks or the convexity of the objective function. Our analysis is broadly applicable to many ADMM variants involving proximal update rules and various flexible block selection rules.

1. INTRODUCTION

Consider the following linearly constrained (possibly nonsmooth or/and nonconvex) problem with K blocks of variables $\{x_k\}_{k=1}^K$:

$$\begin{aligned} \min \quad & f(x_1, \cdots, x_K) := \sum_{k=1}^K g_k(x_k) + \ell(x_1, \cdots, x_K) \\ \text{s.t.} \quad & \sum_{k=1}^K A_k x_k = q, \; x_k \in X_k, \; \forall\, k = 1, \cdots, K \end{aligned} \tag{1.1}$$

where $A_k \in \mathbb{R}^{M \times N_k}$ and $q \in \mathbb{R}^M$; $X_k \subseteq \mathbb{R}^{N_k}$ is a closed convex set; $\ell(\cdot)$ is a smooth (possibly nonconvex) function; each $g_k(\cdot)$ can be either a smooth function, or a convex nonsmooth function. The augmented Lagrangian for problem (1.1) is given by

$$\begin{aligned} L(x_1, \cdots, x_K; y) = & \sum_{k=1}^K g_k(x_k) + \ell(x_1, \cdots, x_K) \\ & + \langle y, q - \sum_{k=1}^K A_k x_k \rangle + \frac{\rho}{2} \|q - \sum_{k=1}^K A_k x_k\|^2, \end{aligned} \tag{1.2}$$

where $\rho > 0$ is a constant representing the primal step-size.

To solve problem (1.1), consider the popular alternating direction method of multipliers (ADMM) displayed below:

Algorithm 0. ADMM for Problem (1.1)

At each iteration $t+1$, update the primal variables:
$$x_k^{t+1} = \underset{x_k \in X_k}{\operatorname{argmin}}\, L(x_1^{t+1}, \cdots, x_{k-1}^{t+1}, x_k, x_{k+1}^t, \cdots, x_K^t; y^t), \; \forall\, k.$$

Update the dual variable:
$$y^{t+1} = y^t + \rho(q - \sum_{k=1}^K A_k x_k^{t+1}).$$

The ADMM algorithm was originally introduced in early 1970s [1, 2], and has since been studied extensively [3–5]. Recently it has become popular in big data related problems arising in various engineering domains; see, e.g., [6–13].

There is a vast literature that applies the ADMM algorithm for solving problems in the form of (1.1). Most of its convergence analysis is done for certain special form of problem (1.1) — the *two-block convex separable* problems, where $K = 2$, $\ell = 0$ and g_1, g_2 are both convex. In this case, ADMM is known to converge under very mild conditions; see [6]. Recent analysis on its rate of convergence can be found in [14–18]. For the *multi-block* separable convex problems where $K \geq 3$, it is known that the original ADMM can diverge for certain pathological problems [19]. Therefore, most research effort in this direction has been focused on analyzing the convergence of variants of the ADMM; see for example [19–25]. When the objective function is no longer separable among the variables, the convergence of the ADMM is still open, even in the case where $K = 2$ and $f(\cdot)$ is convex. Recent works of [22, 26] have shown that when problem (1.1) is convex but not necessarily separable, and when certain error bound condition is satisfied, then the ADMM iteration converges to the set of primal-dual optimal solutions, provided that the dual stepsize is decreasing.

Unlike the convex case, the behaviour of the ADMM is rarely analyzed when it is applied to solve nonconvex problems. Nevertheless, it has been observed by many researchers that the ADMM works well for various applications involving nonconvex objectives, such as the nonnegative matrix factorization, phase retrieval, distributed matrix factorization; see [27–36] and the references therein. However, to the best of our knowledge, existing convergence analysis of ADMM for nonconvex problems is limited — most of the known global convergence analysis needs to impose overly restrictive conditions on the sequence generated by the algorithm. Reference [37] analyzes a family of splitting algorithms (which includes the ADMM as a special case) for certain nonconvex quadratic optimization problem, and shows that they converge to the stationary solution when certain condition on the dual stepsize is met.

In this paper, we analyze the convergence of ADMM for two special types of nonconvex problems in the form of (1.1) – a family of nonconvex consensus and sharing problems. We show that for those problems ADMM converges without any assumptions on the iterates. That is, as long as the problem (1.1) satisfies certain regularity conditions, and the stepsize ρ is chosen large enough (with computable bounds), then the algorithm is guaranteed to converge to the set of stationary solutions. Further, we generalize the ADMM to allow per-block proximal update as well as flexible block selection. An interesting consequence of our analysis is that for a particular reformulation of the sharing problem, the *multi-block* ADMM con-

978-1-4673-6997-8/15/$31.00

verges, regardless of the convexity of the objective function.

2. THE NONCONVEX CONSENSUS PROBLEM

Consider the following nonconvex consensus problem

$$\min \quad f(x) := \sum_{k=1}^{K} g_k(x) + h(x), \quad \text{s.t.} \quad x \in X \tag{2.3}$$

where each g_k is a smooth but possibly nonconvex function; $h(x)$ is a convex possibly nonsmooth function. This problem is related to the convex consensus problem discussed in [6, Section 7], but with the important difference that g_k can be nonconvex.

In many practical applications, each g_k is handled by a single agent, such as a thread or processor. This motivates the following consensus formulation. Let us introduce a set of new variables $\{x_k\}_{k=1}^K$, and transform problem (2.3) equivalently to the following linearly constrained problem

$$\begin{aligned} \min \quad & \sum_{k=1}^{K} g_k(x_k) + h(x) \\ \text{s.t.} \quad & x_k = x,\ \forall\, k = 1, \cdots, K, \quad x \in X. \end{aligned} \tag{2.4}$$

The augmented Lagrangian function is given by

$$\begin{aligned} L(\{x_k\}, x; y) = & \sum_{k=1}^{K} g_k(x_k) + h(x) + \sum_{k=1}^{K} \langle y_k, x_k - x\rangle \\ & + \sum_{k=1}^{K} \frac{\rho_k}{2}\|x_k - x\|^2. \end{aligned} \tag{2.5}$$

Problem (2.4) can be solved distributedly by applying the classical ADMM algorithm. The details are given in the table below.

Algorithm 1. The classical ADMM for Problem (2.4)

At each iteration $t+1$, compute:

$$x^{t+1} = \arg\min_{x\in X} L(\{x_k^t\}, x; y^t).$$

Each node k computes x_k in parallel, by solving:

$$x_k^{t+1} = \arg\min_{x_k} g_k(x_k) + \langle y_k^t, x_k - x^{t+1}\rangle + \frac{\rho_k}{2}\|x_k - x^{t+1}\|^2.$$

Each node k updates the dual variable:

$$y_k^{t+1} = y_k^t + \rho_k\left(x_k^{t+1} - x^{t+1}\right).$$

In Algorithm 1, the x update step can be usually expressed in closed form (depending on the choice of X and h). For example when $h \equiv 0$, then it is a simple projection

$$x^{t+1} = \text{proj}_X\left[\frac{\sum_{k=1}^K \rho_k x_k^t + \sum_{k=1}^K y_k^t}{\sum_{k=1}^K \rho_k}\right] \tag{2.6}$$

where proj_X is the *projection operator* on to the set X. If h is present, then the projection operator can be replaced by the well-known proximity operator. Note that x can be viewed as the first block and $\{x_k\}_{k=1}^K$ together is the second block. Therefore the two primal blocks are updated in a sequential (i.e., Gauss-Seidel) manner. In this paper we will analyze a more general version, in which the blocks are updated in a *flexible* manner; see Algorithm 2. Specifically, let $k = 1, \cdots, K$ denote the indices for the primal variables $x_1, \cdots, x_K$ and $k = 0$ be the index for primal block x. Use an index set $\mathcal{C}^t \subseteq \{0, \cdots, K\}$ to denote the set of variables updated in iteration t. We consider the following two types of index update rules:

1. *Randomized update rule*: At each iteration $t+1$, the index set $\mathcal{C}^{t+1}$ is chosen randomly so that
$$\Pr\left(k \in \mathcal{C}^{t+1} \mid \left\{x^i, y^i, \{x_k^i\}\right\}_{i=1}^t\right) = p_k^{t+1} \geq p_{\min} > 0.$$
2. *Essentially cyclic (EC) update rule*: There exists a period $T \geq 1$ during which each index is updated at least once, i.e.,
$$\bigcup_{i=1}^{T} \mathcal{C}^{t+i} = \{0, 1, \cdots, K\},\ \forall\, t. \tag{2.7}$$

We call this update rule a *period-T* EC rule.

Algorithm 2. The Flexible ADMM for Problem (2.4)

At each iteration $t+1$, pick an index set $\mathcal{C}^{t+1} \subseteq \{0, \cdots, K\}$.

If $0 \in \mathcal{C}^{t+1}$, compute:

$$x^{t+1} = \arg\min_{x\in X} L(\{x_k^t\}, x; y^t). \tag{2.8}$$

Else $x^{t+1} = x^t$.

If $0 \neq k \in \mathcal{C}^{t+1}$, node k computes x_k by solving:

$$x_k^{t+1} = \arg\min_{x_k} g_k(x_k) + \langle y_k^t, x_k - x^{t+1}\rangle + \frac{\rho_k}{2}\|x_k - x^{t+1}\|^2.$$

Update the dual variable:

$$y_k^{t+1} = y_k^t + \rho_k\left(x_k^{t+1} - x^{t+1}\right).$$

Else $x_k^{t+1} = x_k^t$, $y_k^{t+1} = y_k^t$.

The randomized version of Algorithm 2 is similar to that of the convex consensus algorithms studied in [38, 39]. It is also related to the *randomized BSUM-M* algorithm studied in [22]. The difference with the latter is that in the randomized BSUM-M, the dual variable is viewed as an additional block that can be randomly picked (independent of the way that the primal blocks are picked), whereas in Algorithm 2, the dual variable y_k is always updated whenever the corresponding primal variable x_k is updated. To the best of our knowledge, the period-T essentially cyclic update rule is a new variant of the ADMM.

Clearly Algorithm 1 is simply Algorithm 2 with period-1 EC rule. Therefore we will focus on analyzing Algorithm 2. To this end, we make the following assumption.

Assumption A.

A1. There exists a positive constant $L_k > 0$ such that
$$\|\nabla_k g_k(x_k) - \nabla_k g_k(z_k)\| \leq L_k\|x_k - z_k\|,\ \forall\, x_k, z_k,\ \forall\, k.$$
Moreover, h is convex (possible nonsmooth); X is a closed convex set.

A2. For all k, the stepsize ρ_k is chosen large enough such that:

1. The x_k subproblem is strongly convex with the strongly convexity coefficient being $\gamma_k(\rho_k)$;

2. $\rho_k \gamma_k(\rho_k) > 2L_k^2$ and $\rho_k \geq L_k$.

A3. $f(x)$ is lower bounded for all $x \in X$.

We have the following remarks regarding to Assumption A.

- If we are able to increase ρ_k to make the x_k subproblem strongly convex with respect to (w.r.t.) x_k, then the modulus $\gamma_k(\rho_k)$ is a monotonic increasing function of ρ_k.
- Whenever $g_k(\cdot)$ is nonconvex (therefore $\rho_k > \gamma_k(\rho_k)$), the condition $\rho_k \gamma_k(\rho_k) \geq 2L_k^2$ implies $\rho_k \geq L_k$.
- By Assumption A2., $L(\{x_k\}, x; y)$ is strongly convex w.r.t. x_k for all k, with modulus $\gamma_k(\rho_k)$; and $L(\{x_k\}, x; y)$ is strongly convex w.r.t. x, with modulus $\gamma := \sum_{k=1}^K \rho_k$.
- Assumption A does not impose any restriction on the *iterates* generated by the algorithm. This is in contrast to the existing analysis of the nonconvex ADMM algorithms [27, 33, 35].

Now we state the first main result of this paper. Due to space limitation, we refer the readers to [40] for detailed proof. We briefly mention that the key of the proof is to use the *reduction of the augmented Lagrangian* to measure the progress of the algorithm.

Theorem 2.1 *Assume that Assumption A is satisfied. Then the following is true for Algorithm 2:*

1. $\lim_{t\to\infty} \|x_k^{t+1} - x^{t+1}\| = 0$, $\forall, k$, *deterministically for the EC rule and almost surely (a.s.) for randomized rule.*
2. *Let* $(\{x_k^*\}, x^*, y^*)$ *denote any limit point of the sequence* $\{\{x_k^{t+1}\}, x^{t+1}, y^{t+1}\}$ *generated by Algorithm 2. Then the following statement is true (deterministically for the EC rule and a.s. for the randomized update rule)*

$$0 = \nabla g_k(x_k^*) + y_k^*, \quad x_k^* = x^*, \; k = 1, \cdots, K.$$

$$x^* \in \arg\min_{x\in X} \quad h(x) + \sum_{k=1}^K \langle y_k^*, x_k^* - x\rangle$$

That is, any limit point of Algorithm 2 is a stationary solution of problem (2.4).

3. *If X is a compact set, then Algorithm 2 converges to the set of stationary solutions of problem* (2.4).

3. THE NONCONVEX SHARING PROBLEM

Consider the following well-known sharing problem (see, e.g., [6, Section 7.3] for motivation)

$$\begin{aligned} \min \quad & f(x_1, \cdots, x_K) := \sum_{k=1}^K g_k(x_k) + \ell\left(\sum_{k=1}^K A_k x_k\right) \\ \text{s.t.} \quad & x_k \in X_k, \; k = 1, \cdots, K \end{aligned} \tag{3.9}$$

where $x_k \in \mathbb{R}^{N_k}$ is the variable associated with a given agent k, and $A_k \in \mathbb{R}^{M\times N_k}$ is some data matrix. The variables are coupled through the function $\ell(\cdot)$.

To facilitate distributed computation, this problem can be equivalently formulated into a linearly constrained problem by introducing an additional variable $x \in \mathbb{R}^M$:

$$\begin{aligned} \min \quad & \sum_{k=1}^K g_k(x_k) + \ell(x) \\ \text{s.t.} \quad & \sum_{k=1}^K A_k x_k = x, \quad x_k \in X_k, \; k = 1, \cdots, K. \end{aligned} \tag{3.10}$$

The augmented Lagrangian for this problem is given by

$$\begin{aligned} L(\{x_k\}, x; y) = & \sum_{k=1}^K g_k(x_k) + \ell(x) + \left\langle x - \sum_{k=1}^K A_k x_k, y \right\rangle \\ & + \frac{\rho}{2}\left\| x - \sum_{k=1}^K A_k x_k \right\|^2. \end{aligned} \tag{3.11}$$

Note that (3.10) is a special reformulation of (3.9). When applying classical ADMM to this reformulation leads to the so-called *multi-block* ADMM algorithm. That is, in classical ADMM, there are $K+1$ blocks of primal variables $\{x, \{x_k\}_{k=1}^K\}$ to be sequentially updated. As mentioned in the introduction, even in the case where the objective is convex, it is not known whether the multi-block ADMM converges. Interestingly, we will show that the classical ADMM, together with several of its extensions using different block selection rules, converges for a special form of the $K+1$ block sharing problem (3.10). The table below shows the flexible ADMM algorithm for the sharing problem, where $\mathcal{C}^{t+1}$ again denotes a flexible block update rule.

Algorithm 3. The Flexible ADMM for the Sharing Problem (3.10)

At each iteration $t \geq 1$, pick an index set $\mathcal{C}^{t+1} \in \{0, \cdots, K\}$.
For $k = 1, \cdots, K$
If $k \in \mathcal{C}^{t+1}$, then agent k updates x_k by:

$$\begin{aligned} x_k^{t+1} = \arg\min_{x_k\in X_k} \; & g_k(x_k) - \langle y^t, A_k x_k\rangle \\ & + \frac{\rho}{2}\left\| x^t - \sum_{j<k} A_j x_j^{t+1} - \sum_{j>k} A_j x_j^t - A_k x_k \right\|^2 \end{aligned} \tag{3.12}$$

Else $x_k^{t+1} = x_k^t$.
If $0 \in \mathcal{C}^{t+1}$, update the variable x by:

$$x^{t+1} = \arg\min_x \ell(x) + \langle y^t, x\rangle + \frac{\rho}{2}\left\| x - \sum_{k=1}^K A_k x_k^{t+1} \right\|^2. \tag{3.13}$$

Update the dual variable:

$$y^{t+1} = y^t + \rho\left(x^{t+1} - \sum_{k=1}^K A_k x_k^{t+1}\right). \tag{3.14}$$

Else $x^{t+1} = x^t$, $y^{t+1} = y^t$.

Clearly, when the update rule $\mathcal{C}^{t+1} = \{0, \cdots, K\}$, we recover the classical ADMM, which has $K+1$ blocks of variables.

The analysis of Algorithm 3 follows similar argument as that of Algorithm 2. We again refer the readers to [40] for details.

First, we make the following assumptions in this section.

Assumption B.

B1. There exists a positive constant $L > 0$ such that

$$\|\nabla \ell(x) - \nabla \ell(z)\| \leq L\|x - z\|, \; \forall\, x, z.$$

Moreover, X_k's are closed convex sets; each A_k is full column rank, with $\sigma_{\min}(A_k^T A_k) > 0$.

B2. The stepsize ρ is chosen large enough such that:

(1) each x_k subproblem (3.12) as well as the x subproblem (3.13) is strongly convex, with modulus $\{\gamma_k(\rho)\}_{k=1}^K$ and $\gamma(\rho)$, respectively.

(2) $\rho\gamma(\rho) > 2L^2$, and that $\rho \geq L$.

B3. $f(x_1, \cdots, x_K)$ is lower bounded over $\prod_{k=1}^K X_k$.

B4. g_k is either smooth nonconvex or convex (possibly nonsmooth). For the former case, there exists $L_k > 0$ such that $\|g_k(x_k) - g_k(z_k)\| \leq L_k\|x_k - z_k\|, \forall\, x_k, z_k \in X_k$.

Define an index set $\mathcal{K} \subseteq \{1, \cdots, K\}$, such that g_k is convex if $k \in \mathcal{K}$, and nonconvex smooth otherwise. Note that the requirement that A_k is full column rank is needed to make the x_k subproblem (3.12) strongly convex. Our second main result is given below.

Theorem 3.1 *Assume that Assumption B is satisfied. Then the following is true for Algorithm 3:*

1. $\lim_{t\to\infty} \|x_k^{t+1} - x^{t+1}\| = 0, \forall\, k$, *deterministically for the EC rule and a.s. for the randomized update rule.*
2. *Let* $(\{x_k^*\}, x^*, y^*)$ *denote any limit point of the sequence* $\{\{x_k^{t+1}\}, x^{t+1}, y^{t+1}\}$ *generated by Algorithm 3. We have*

$$x_k^* \in \arg\min_{x_k \in X_k}\; g_k(x_k) + \langle y^*, -A_k x_k^*\rangle,\; k \in \mathcal{K},$$

$$\left\langle x_k - x_k^*, \nabla g_k(x_k^*) - A_k^T y^* \right\rangle \geq 0,\; \forall\, x_k \in X_k,\; k \notin \mathcal{K},$$

$$\nabla\ell(x^*) + y^* = 0, \quad \sum_{k=1}^K A_k x_k^* = x^*.$$

That is, every limit point of Algorithm 3 is a stationary solution of problem (3.10)*.*

3. *If* X_k *is a compact set for all* k*, then Algorithm 3 converges to the set of stationary solutions of problem* (3.10)*.*

The following corollary specializes the previous convergence result to the case where all g_k's as well as ℓ are convex (but not necessarily strongly convex). We emphasize that this is still a nontrivial result, as it is not known whether the classical ADMM converges for the multi-block problem (3.10), even in the convex case.

Corollary 3.1 *Suppose that assumptions B1 and B3 are true. Further suppose that assumption B2 is weakened by the following*

1. *The stepsize* ρ *is chosen to satisfy* $\rho > \sqrt{2}L$.

Then the flexible ADMM algorithm (i.e., Algorithm 3) converges to the set optimal primal-dual solutions of problem (3.10)*, deterministically for the EC rule and a.s. for the randomized update rule.*

To close this section, we provide a remark related to the possibility of generalizing Algorithm 2-3 to include proximal steps.

Remark 3.1 *In certain applications it is beneficial to have cheap updates for the subproblems. Algorithm 2 -3 can be further generalized to the case where the subproblems are not solved exactly – only a single proximal update is sufficient for each* x_k *subproblem. Due to space limitations, we refer the readers to [40] for detailed discussion of the proximal versions of Algorithm 2 and 3.*

4. NUMERICAL EXPERIMENTS

We consider a special case of the consensus problem (2.3)

$$\min_x \quad \sum_{k=1}^K x^T B_k x + \lambda\|x\|_1, \quad \text{s.t.} \quad \|x\|_2^2 \leq 1, \tag{4.15}$$

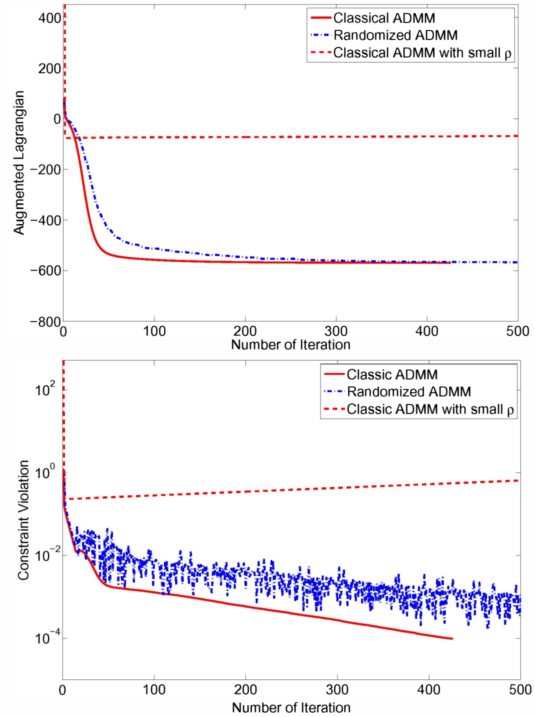

Fig. 1. Top: The value of $L(x^t; y^t)$ for different algorithms. Bottom: The value of $\max_k\{\|x_k^t - x^t\|\}$ for different algorithms.

where $B_k \in \mathbb{R}^{N\times N}$ is a symmetric matrix, and $\lambda \geq 0$ is some constant. This problem is related to the L_1 penalized version of the sparse principal component analysis (PCA) problem, in which case $-B_k \succ 0$ and $K = 1$. Suppose that there are K agents in the system, and agent k possesses data matrix B_k. Then by introducing a set of new variable $\{x_k\}$, the above problem can be formulated similarly as in (2.4). Note that when applying ADMM, each subproblem can be solved in closed form.

In our experiment, we set $N = 1000$, $K = 10$, $\lambda = 100$. Each $B_k = -\sum_{j=1}^5 \xi_j\xi_j^T$, where $\xi_j \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ is a standard Gaussian vector of size N. Each stepsize ρ_k is chosen according to Assumption A.2 [1]. We run both the classical and the randomized versions of Algorithm 1, and for the latter case we choose $p_k^t = 0.9$ for all k, t. We also run the classical ADMM with *small* stepsizes that violate the rule set in Assumption A.2, i.e., we let $\hat{\rho}_k = \rho_k/1000, \forall\, k$. The performance of different algorithms is shown in Fig. 1. By using the stepsize indicated by Assumption A.2, both algorithms converge, while the algorithm diverges when using the smaller stepsize. Moreover, we observe that when $\hat{\rho}_k = \rho_k/2, \forall\, k$, both the violation and the augmented Lagrangian diverge to infinity (although not shown in the figures). This experiment demonstrates one critical difference between the nonconvex and convex ADMM – unlike the latter case, the stepsize ρ for the nonconvex ADMM needs to be picked carefully to ensure convergence.

[1]Note that the stepsizes can be easily calculated with closed-form, as $\gamma_k(\rho_k) = \rho_k - 2\lambda_{\max}(B_k)$, and $L = 2\lambda_{\max}(B_k)$.

5. REFERENCES

[1] R. Glowinski and A. Marroco, "Sur l'approximation, par elements finis d'ordre un,et la resolution, par penalisation-dualite, d'une classe de problemes de dirichlet non lineares," *Revue Franqaise d'Automatique, Informatique et Recherche Opirationelle*, vol. 9, pp. 41–76, 1975.

[2] D. Gabay and B. Mercier, "A dual algorithm for the solution of nonlinear variational problems via finite element approximation," *Computers & Mathematics with Applications*, vol. 2, pp. 17–40, 1976.

[3] J. Eckstein, "Splitting methods for monotone operators with applications to parallel optimization," 1989, Ph.D Thesis, Operations Research Center, MIT.

[4] J. Eckstein and D. P. Bertsekas, "On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators," *Mathematical Programming*, vol. 55, no. 1, pp. 293–318, 1992.

[5] R. Glowinski, *Numerical methods for nonlinear variational problems*, Springer-Verlag, New York, 1984.

[6] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends in Machine Learning*, vol. 3, 2011.

[7] W. Yin, S. Osher, D. Goldfarb, and J. Darbon, "Bregman iterative algorithms for l1-minimization with applications to compressed sensing," *SIAM Journal on Imgaging Science*, vol. 1, no. 1, pp. 143–168, Mar. 2008.

[8] J. Yang, Y. Zhang, and W. Yin, "An efficient TVL1 algorithm for deblurring multichannel images corrupted by impulsive noise," *SIAM Journal on Scientific Computing*, vol. 31, no. 4, pp. 2842–2865, 2009.

[9] X. Zhang, M. Burger, and S. Osher, "A unified primal-dual algorithm framework based on Bregman iteration," *Journal of Scientific Computing*, vol. 46, no. 1, pp. 20–46, 2011.

[10] K. Scheinberg, S. Ma, and D. Goldfarb, "Sparse inverse covariance selection via alternating linearization methods," in *Twenty-Fourth Annual Conference on Neural Information Processing Systems (NIPS)*, 2010.

[11] I. Schizas, A. Ribeiro, and G. Giannakis, "Consensus in ad hoc wsns with noisy links - part i: Distributed estimation of deterministic signals," *IEEE Transactions on Signal Processing*, vol. 56, no. 1, pp. 350 – 364, 2008.

[12] C. Feng, H. Xu, and B. Li, "An alternating direction method approach to cloud traffic management," 2014, preprint.

[13] W.-C. Liao, M. Hong, Hamid Farmanbar, Xu Li, Z.-Q. Luo, and Hang Zhang, "Min flow rate maximization for software defined radio access networks," *IEEE Journal on Selected Areas in Communication*, vol. 32, no. 6, pp. 1282–1294, 2014.

[14] B. He and X. Yuan, "On the O(1/n) convergence rate of the Douglas-Rachford alternating direction method," *SIAM Journal on Numerical Analysis*, vol. 50, no. 2, pp. 700–709, 2012.

[15] R. Monteiro and B. Svaiter, "Iteration-complexity of block-decomposition algorithms and the alternating direction method of multipliers," *SIAM Journal on Optimization*, vol. 23, no. 1, pp. 475–507, 2013.

[16] T. Goldstein, B. O'Donoghue, and S. Setzer, "Fast alternating direction optimization methods," *UCLA CAM technical report*, 2012.

[17] D. Goldfarb, S. Ma, and K. Scheinberg, "Fast alternating linearization methods for minimizing the sum of two convex functions," *Mathematical Programming*, vol. 141, no. 1-2, pp. 349–382, 2012.

[18] W. Deng and W. Yin, "On the global linear convergence of alternating direction methods," 2012, preprint.

[19] C. Chen, B. He, X. Yuan, and Y. Ye, "The direct extension of ADMM for multi-block convex minimization problems is not necessarily convergent," 2013, preprint.

[20] M. Hong and Z.-Q. Luo, "On the linear convergence of the alternating direction method of multipliers," *arXiv preprint arXiv:1208.3922*, 2012.

[21] B. He, M. Tao, and X. Yuan, "Alternating direction method with Gaussian back substitution for separable convex programming," *SIAM Journal on Optimization*, vol. 22, pp. 313–340, 2012.

[22] M. Hong, T.-H. Chang, X. Wang, M. Razaviyayn, S. Ma, and Z.-Q. Luo, "A block successive upper bound minimization method of multipliers for linearly constrained convex optimization," 2013, Preprint, available online arXiv:1401.7079.

[23] D. Han and X. Yuan, "A note on the alternating direction method of multipliers," *Journal of Optimization Theory and Applications*, vol. 155, no. 1, pp. 227–238, 2012.

[24] W. Deng, M. Lai, Z. Peng, and W. Yin, "Parallel multi-block ADMM with o(1/k) convergence," Preprint, available online at arXiv: 1312.3040., 2014.

[25] B. He, H. Xu, and X. Yuan, "On the proximal jacobian decomposition of alm for multipleblock separable convex minimization problems and its relationship to ADMM," 2013, Preprint, available on Optimization-Online.

[26] M. Hong, T.-H. Chang, X. Wang, M. Razaviyayn, S. Ma, and Z.-Q. Luo, "A block coordinate descent method of multipliers: Convergence analysis and applications," in *International Conference on Acoustics, Speech and Signal Processing*, 2014.

[27] Y. Zhang, "An alternating direction algorithm for nonnegative matrix factorization," 2010, Preprint.

[28] D. L. Sun and C. Fevotte, "Alternating direction method of multipliers for non-negative matrix factorization with the beta-divergence," in *the Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2014.

[29] Z. Wen, C. Yang, X. Liu, and S. Marchesini, "Alternating direction methods for classical and ptychographic phase retrieval," *Inverse Problems*, vol. 28, no. 11, pp. 1–18, 2012.

[30] R. Zhang and J. T. Kwok, "Asynchronous distributed admm for consensus optimization," in *Proceedings of the 31st International Conference on Machine Learning*, 2014.

[31] P.A. Forero, A. Cano, and G.B. Giannakis, "Distributed clustering using wireless sensor networks," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 4, pp. 707–724, Aug 2011.

[32] B. Ames and M. Hong, "Alternating directions method of multipliers for l1-penalized zero variance discriminant analysis and principal component analysis," 2014, Preprint.

[33] B. Jiang, S. Ma, and S. Zhang, "Alternating direction method of multipliers for real and complex polynomial optimization models," 2013, Preprint.

[34] Y. Shen, Z. Wen, and Y. Zhang, "Augmented lagrangian alternating direction method for matrix separation based on low-rank factorization," *Optimization Methods Software*, vol. 29, no. 2, pp. 239–263, Mar. 2014.

[35] Y. Xu, W. Yin, Z. Wen, and Y. Zhang, "An alternating direction algorithm for matrix completion with nonnegative factors," *Journal of Frontiers of Mathematics in China, Special Issues on Computational Mathematics*, pp. 365–384, 2011.

[36] Z. Wen, X. Peng, X. Liu, X. Bai, and X. Sun, "Asset allocation under the basel accord risk measures," 2013, Preprint.

[37] Y. Zhang, "Convergence of a class of stationary iterative methods for saddle point problems," 2010, Preprint.

[38] E. Wei and A. Ozdaglar, "On the O(1/k) convergence of asynchronous distributed alternating direction method of multipliers," 2013, Preprint, available at arXiv:1307.8254.

[39] T.-H. Chang, "A proximal dual consensus admm method for multi-agent constrained optimization," 2014, Preprint, available at arXiv:1409.3307.

[40] M. Hong, Z.-Q. Luo, and M. Razaviyayn, "Convergence analysis of alternating direction method of multipliers for a family of nonconvex problems," 2014, technical report, University of Minnesota.