Parallel Multi-splitting Proximal Method for Star Networks

Ermin Wei
Department of Electrical Engineering and Computer Science
Northwestern University
Evanston, IL 60202
ermin.wei@northwestern.edu

Abstract— **We develop a parallel algorithm based on proximal method to solve the problem of minimizing summation of convex (not necessarily smooth) functions over a star network. We show that this method converges to an optimal solution for** *any choice of constant stepsize* **for convex objective functions. Under further assumption of Lipschitz-gradient and strong convexity of objective functions, the method converges linearly.**

I. Introduction

We consider the following class of optimization problem $\min_x \sum_{i=1}^n f_i(x)$, which has gained much research attention recently. It captures many important applications such as distributed control for a team of autonomous robots/UAVs pursuing/aiming at a common target, sensor networks constructing an estimation for the entire surrounding, communication systems maximizing system throughput, and machine learning applications [16], [11], [17], [6], [10], [13], [18]. Most of the existing literature for solving this problem either does not explore parallel potential [17], [6], [10], [23] or requires a careful selection of stepsize to guarantee convergence [18], [13]. The requirement of stepsize tuning can be computationally expensive, and undermines the robustness of the entire system to provide an optimal solution.

The only line of distributed algorithms that does not suffer from the drawbacks of stepsize selection is Alternating Direction Method of Multiplier (ADMM) based algorithms [4], [16], [21], [22], [12], [11], [24], [16], [23], [2], [9], [15], which has gained much popularity due to great numerical performance. A closer look of the standard ADMM reveals that it is a two-way splitting proximal algorithm [7], where a two-way splitting of the dual function is formed and then proximal method is applied iteratively to both parts. However, as observed in recent work [5], while the standard two-way splitting ADMM (corresponding to a two-agent setting in a multi-agent setup) can converge for any stepsize choice, a three-way splitting of the dual function may result in an algorithm that diverges. Hence, in order to use ADMM in a distributed setting with more than two agents, complex reformulation of the problem and introduction of auxiliary (primal and dual) variables are required [3], [4], [21]. Despite the two promising features that proximal-based methods do not require stepsize selection and that multi-splitting arises naturally from multi-agent setup, the question of whether we can design a convergent algorithm based on more than two-way splitting proximal method remains open.

In this paper, we combine ideas from proximal method and projection to develop a multi-splitting proximal algorithm that works with non-smooth convex objective function, takes advantage of parallel processing power and *guarantees asymptotic convergence for any positive stepsize*. We also analyze its rate of convergence under stronger assumptions of Lipschitz gradient and strong convexity and show that the algorithm converges linearly.

Our paper is related to the large literature on distributed/parallel computation, building upon seminal works [3] and [20]. In particular, the distributed gradient descent method [13] and EXTRA [18] method. The distributed gradient method can be applied to non-smooth objective function, however a constant stepsize would only guarantee convergence to an error neighborhood of the optimal solution. The recently proposed distributed first-order method, EXTRA, uses a constant stepsize and converges to an optimal point. However, the algorithm does require careful selection of stepsizes to guarantee convergence and smoothness of the objective functions, which limits its applicability to important problems with non-smooth regularization term, such as the LASSO.

The most closely related literature for our algorithm is [19] from 1983, which was later generalized in [8]. These authors combine multi-splitting proximal method and projection to form a new algorithm. Spingarn's algorithm is a special case of our algorithm with a unit stepsize. We also note that these papers do not have rate of convergence analysis (under Lipschitz gradient and strong convexity assumptions). The proposed algorithm shares the same rate of convergence as some existing algorithms, such as EXTRA and ADMM, the main advantage is its robustness against stepsize selection and simple implementation in distributed setting. While we focus on the star network in this work, this serves as a building block to develop distributed methods for general network topologies. The rest of the paper, we will first present the algorithm along with some preliminary simulation results and then the convergence and rate of convergence analysis.

II. Algorithm

We present the proposed algorithm in this section. First, we note that the original problem can be equivalently expressed

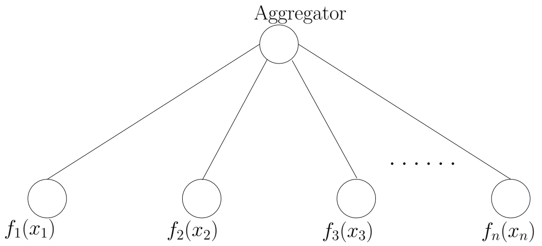

Fig. 1. Parallel architecture to implement the proposed algorithm.

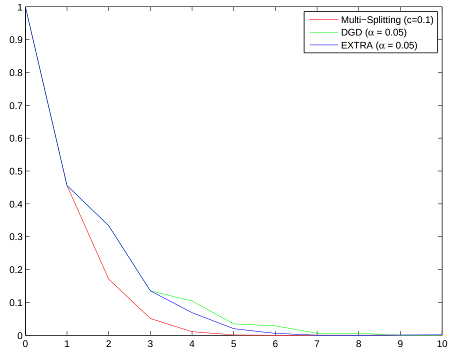

Fig. 2. Preliminary numerical result. Y-axis: relative error $\frac{f(x^t)-f^*}{f(x^0)-f^*}$, x-axis: iteration count.

as

$$\begin{aligned} \min_{x} \quad & \sum_{i=1}^{n} f_i(x_i), \\ \text{s.t.} \quad & x_1 = \ldots = x_n. \end{aligned} \tag{1}$$

We adopt a general setup where each function $f_i : \mathbb{R}^m \to \mathbb{R}$ is convex but not necessarily differentiable. We aim at developing algorithm to solve this reformulated problem under the following standard assumption.

Assumption 1: Problem (1) has a non-empty optimal solution set, denoted by X^*.

This condition does not require uniqueness of the optimal solution. The parallel algorithm is implemented on $n+1$ machines, connected in a star graph as shown in Figure 1. We call the one in the center the *aggregator*, the rest of them *workers* labeled $\{1, \ldots, n\}$. [1] Each worker has information about one function f_i and specializes in computing the proximal operator related to f_i. Collectively, the workers and aggregator are solving problem (1). Our algorithm is an iterative method, where the updates are related to decision variable and first order information. At each iteration, the workers in parallel perform a proximal point updates for their respective f_i using the current state information received from the aggregator (related to decision variable and corresponding first order information) and sends the updated information to the aggregator. The aggregator then averages the information according to a specific rule and sends the averaged information as the new state back to each worker. In particular, we use the following set of notation to describe our algorithm. In our algorithm, we use superscript to indicate the iteration count and subscript to indicator the worker that is associated with the variable. Positive parameter c is the stepsize and is a constant throughout the algorithm. Our algorithm is presented in Algorithm 1.

Algorithm 1 Parallel Multi-splitting Proximal Method

Initialization: The aggregator starts from arbitrary $\tilde{x}^0$ and $\tilde{v}_i^0$ in $\mathbb{R}^m$ for i in $1, \ldots, n$, compute $x_i^0 = \tilde{x}^0$, for all $i = 1, \ldots n$ and $v_i^0 = \tilde{v}_i^0 - \frac{\sum_{i=1}^n \tilde{v}_i^0}{n}$. The aggregator sends information $x_i^0 + cv_i^0$ (in $\mathbb{R}^m$) to each worker i.

Iteration: for $k = 0, 1, \ldots$

- Worker $i = 1, \ldots n$ computes in parallel

$$y_i^{k+1} \in \operatorname*{argmin}_{p} f_i(p) + \frac{1}{2c} \left|\left|p - x_i^k - cv_i^k\right|\right|^2, \tag{2a}$$

$$w_i^{k+1} = \frac{1}{c}(x_i^k + cv_i^k - y_i^{k+1}), \tag{2b}$$

  and reports y_i^{k+1} and w_i^{k+1} (each in $\mathbb{R}^m$) back to the aggregator.
- After receiving y_i^{k+1} and w_i^{k+1} information from all n workers, the aggregator generates

$$x_i^{k+1} = \frac{\sum_{i=1}^n y_i^{k+1}}{n}, \quad v_i^{k+1} = w_i^{k+1} - \frac{\sum_{i=1}^n w_i^{k+1}}{n},$$

  for $i = 1, \ldots, n$ and then sends information $x_i^{k+1} + cv_i^{k+1}$ (in $\mathbb{R}^m$) to each worker i.

The $\{y_i^k\}_k$ sequence can be viewed as local estimates of x^*. At each time instant k, x_i^k is the same for all i and equals to the average of all local estimates. The $\{w_i^k\}$ sequence as shown later in Lemma 3.1, represents a local subgradient associated with x_i^{k+1} of function f_i. The variable v_i^k captures the difference between local subgradient and the average of all subgradients.

This algorithm is well suited for problems where step (2a), minimization related to one component of the objective function, can be implemented in an efficient way. Examples include SVM, quadratic objective functions, Lasso (Least Absolute Shrinkage and Selection Operator) (see [4], [15] for more examples). When analyzing convergence speed for this algorithm, we focus on the iteration count of k, and not counting the time needed to solve step (2a). We have performed some initial numerical studies to compare our method against distributed gradient descent (DGD) [13] and EXTRA [18] with $n = 4$, $m = 1$ and quadratic objective functions. We plot the relative error in objective function in Figure 2. We used stepsize of 0.1 for the proposed method and 0.05 for DGD and EXTRA,as they both diverge for the stepsize choice of 0.1 and needed smaller stepsize.

III. Convergence analysis

In this section, we analyze the convergence and speed of convergence of the proposed algorithm. For concise representation, we introduce the following notation. Vector

[1] The number of machines can be reduced via mini-batching and/or requiring the aggregator to also process information about one of the function f_i. We chose to present the setting with maximum parallelism.

$x^k = [x_i^k]_i$ in $\mathbb{R}^{nm}$ is a long vector formed by stacking x_i^k, i.e., $x^k = \begin{bmatrix} x_1^k \\ \vdots \\ x_n^k \end{bmatrix}$. Similarly, we form vectors $y^k = [y_i^k]_i$, $v^k = [v_i^k]_i$ and $w^k = [w_i^k]_i$, all in $\mathbb{R}^{nm}$. Unless otherwise specified, vectors such as x_i with sub-indices lie in $\mathbb{R}^m$ and those without sub-indices, such as x^k are in $\mathbb{R}^{nm}$. We denote by $F : \mathbb{R}^{nm} \to \mathbb{R}^n$, and $\partial F : \mathbb{R}^{nm} \rightrightarrows \mathbb{R}^{nm}$, the mappings $F\left(\begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}\right) = \begin{bmatrix} f_1(x_1) \\ \vdots \\ f_n(x_n) \end{bmatrix}$, $\partial F\left(\begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}\right) = \left\{\begin{bmatrix} v_1 \\ \vdots \\ v_n \end{bmatrix}, v_i \in \partial f_i(x_i)\right\}$,where the notation $\partial f_i(x)$ denotes the subdifferential set, i.e., the set consists of all subgradient of f_i at point x_i. We use $x'y$ to denote the inner product between two vectors x and y. We next show that our algorithm has two components: proximal method and projection, which serves as the basis for convergence analysis.

A. Proximal method

We start by analyzing the sequences y^k and w^k. Step (2a) can be equivalently expressed as

$$y_i^{k+1} \in \mathrm{prox}_{cf_i}(x_i^k + cv_i^k)$$

using definition of proximal operator. We next give a characterization of w_i^{k+1}.

Lemma 3.1: For each iteration k, w_i^{k+1} is in the set $\partial f_i(y_i^{k+1})$.

Proof: By first order optimality condition (see [1] for details) of (2a), we have that there exists a subgradient q of f_i at y_i^{k+1}, such that $q + \frac{1}{c}(y_i^{k+1} - x_i^k - cv_i^k) = 0$. This suggests that $q = \frac{1}{c}(x_i^k + cv_i^k - y_i^{k+1})$. By definition of w_i^{k+1} in Eq. (2b), we have $w_i^{k+1} = q$, and thus w_i^{k+1} is a subgradient and therefore is in the set $\partial f_i(y_i^{k+1})$. ■

The proceeding lemma illustrates that at each iteration, at each worker i, we have a pair of primal decision variable and an associated subgradient (y_i^{k+1}, w_i^{k+1}) obtained based on a proximal step. Hence the $nm-$dimensional vectors y^{k+1}, w^{k+1} also corresponds to decision variable and subgradient pair generated based on a proximal step at $x^k + cv^k$.

B. Projection

We next study the sequence x^k, v^k. Motivated by optimality condition of problem (1), we next introduce the following two subspaces: $A = \left\{\begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}, \quad x_1 = \ldots = x_n\right\}$, $B = \left\{\begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}, \quad \sum_{i=1}^n x_i = 0\right\}$. We use $z(A)$ and $z(B)$ to denote the projection of vector z onto subspaces A and B respectively. We observe that for any optimal solution x^* of (1), first order optimality conditions imply that x^* is in A and there exists a subgradient v in $\partial F(x^*)$, with v in B. The next lemma qualifies the connection between spaces A and B.

Lemma 3.2: The spaces A and B are orthogonal complements.

Proof: For any x in A and y in B, we have $x'y = \sum_{i=1}^n x_i y_i = x_1 \sum_{i=1}^n y_i = 0$. Thus the elements in the two subspaces are orthogonal. We next need to show that any vector z in $\mathbb{R}^{nm}$ can be decomposed as a linear combination of elements of those two sets. Define $\bar{z}_i = \frac{\sum_{i=1}^n z_i}{n}$ and for $i = 1, \ldots, n$, $z_i(A) = \bar{z}_i$, $\quad z_i(B) = z_i - \bar{z}_i$. We can then obtain projections $z(A)$ and $z(B)$ both in $\mathbb{R}^{nm}$ by $z(A) = [z_i(A)]_i$ and $z(B) = [z_i(B)]_i$, which gives $z(A)$ in A, and

$$\sum_{i=1}^n z_i(B) = \sum_{i=1}^n z_i - n\bar{z}_i = \sum_{i=1}^n z_i - \sum_{i=1}^n z_i = 0,$$

thus $z(B)$ in B. We also have $z = z(A) + z(B)$ and this completes the proof. ■

We now note that in our algorithm x^k is a projection of decision variable y^k onto space A and v^k is a projection of subgradient w^k onto the space B. These projections are performed to guide the decision variables and subgradients towards the appropriate subspaces where the optimal solutions live.

C. Convergence

Based on the previous two sections, we conclude that our algorithm is a combination of the proximal method and orthogonal projection method. The convergence analysis is also motivated by the nonexpansive properties of these methods.

Before we proceed to the analysis, we first observe that by definition of w_i^{k+1}, we have

$$y_i^{k+1} + cw_i^{k+1} = x_i^{k+1} + cv_i^{k+1}.$$

Therefore, one iteration of the algorithm can be represented as follows:

$$x^k + cv^k = y^{k+1} + cw^{k+1} \tag{4}$$

$$\downarrow \tag{5}$$

$$x^{k+1} = y^{k+1}(A), \quad v^{k+1} = w^{k+1}(B), \tag{6}$$

where w^{k+1} is in $\partial F(y^{k+1})$ by Lemma 3.1.

Since the two sequences x^k and v^k are in two orthogonal spaces, their sum has a unique orthogonal decomposition and convergence of the sum automatically implies convergence of x^k and v^k. We will focus on the convergence of the sum $x^k + cv^k$.

We first show that any fixed point of the above iteration and the set of optimal solutions to problem (1) are equivalent.

Lemma 3.3: The vector $x + cv$ where x in A and v in B is a fixed point of iteration (6) if and only if x is an optimal solution of problem (1) and v is in $\partial F(x)$.

Proof: We first assume that (x, v) is a fixed point of iteration (6). We use y, w, x^+, v^+ to denote the updates starting from $x^k = x, v^k = v$. Since $x + cv$ is a fixed point, we have $x^+ + cv^+ = x + cv$. Since x, x^+ both are in A and

v, v^+ are both in B, by orthogonality of A, and B, we have $x = x^+$, $v = v^+$. Since $x + cv$ is a fixed point, we have

$$x_i^+ + cv_i^+ = y_i(A) + cw_i(B) = y_i + cw_i = x_i + cv_i, \quad (7)$$

for $i = 1, \ldots, n$. We can then sum over i and have $\sum_{i=1}^n y_i(A) + cw_i(B) = \sum_{i=1}^n x_i^+ + cv_i^+$. By construction of x_i^+, we have $\sum_{i=1}^n y_i(A) = \sum_{i=1}^n x_i^+$. Therefore $\sum_{i=1}^n w_i(B) = \sum_{i=1}^n v_i^+$. Since v^+ is in B, we have $\sum_{i=1}^n v_i^+ = 0$, which implies that $\sum_{i=1}^n w_i(B) = 0$ and w is in the subspace B. Hence $v^+ = w(B) = w = v$. We combine this with Eq. (7), and obtain $x = y = y^+$. Therefore, v in B is also in $\partial F(x)$ with x in A. This suggests that the first order optimality condition is satisfied and therefore the pair (x, v) is an optimal solution and subgradient pair.

Next we start from an optimal solution and subgradient pair (x, v). We have x is in A and v is in B. Since v is in $\partial F(x)$, we have $w = v$ and $y = x$, and the projection will give the original pair back. Thus (x, v) is a pair of fixed point with the components lying in orthogonal subspaces, which implies that $x + cv$ is a fixed point of the iteration. ■

By Assumption 1, we have that the set of fixed points of iteration (6) is nonempty. For the rest of the paper, we use x^*, v^* to denote a fixed point of the iteration (6). We next show that the mapping from $x^k + cv^k$ to $x^{k+1} + cv^{k+1}$ is nonexpansive, which is our building block for convergence analysis.

Theorem 3.4: Let x^* in A denote an optimal solution of problem (1) and v^* in B be a subgradient of $\partial F(x^*)$. Then any sequence of x^k, v^k, y^k, w^k generated by Algorithm 1, we have for all k

$$\begin{aligned} &\left\|x^{k+1} - x^*\right\|^2 + \left\|cv^{k+1} - cv^*\right\|^2 \qquad (8)\\ &= \left\|x^k - x^*\right\|^2 + \left\|cv^k - cv^*\right\|^2 - \left\|y^{k+1} - x^{k+1}\right\|^2\\ &- \left\|cw^{k+1} - cv^{k+1}\right\|^2 - 2(y^{k+1} - x^*)'(cw^{k+1} - cv^*), \end{aligned}$$

$$(y^k - x^*)'(cw^k - cv^*) = \sum_{i=1}^n (y_i^k - x_i^*)'(w_i^k - v_i^*) \geq 0. \quad (9)$$

The sequence $\{\left\|x^k - x^*\right\|^2 + \left\|cv^k - cv^*\right\|^2\}_k$ is monotonically nonincreasing.

Proof: We first apply the equality of the form $\left\|z^{k+1} - z^*\right\|^2 + \left\|z^k - z^{k+1}\right\|^2 + 2(z^k - z^{k+1})'(z^{k+1} - z^*) = \left\|z^k - z^*\right\|^2$, where $z = x + cv$ with the corresponding superscript, to $\left\|x^{k+1} + cv^{k+1} - x^* - cv^*\right\|^2$ and obtain

$$\begin{aligned} &\left\|x^{k+1} + cv^{k+1} - x^* - cv^*\right\|^2 \qquad (10)\\ &= \left\|x^k + cv^k - x^* - cv^*\right\|^2\\ &- \left\|x^k + cv^k - x^{k+1} - cv^{k+1}\right\|^2\\ &- 2(x^k + cv^k - x^{k+1} - cv^{k+1})'(x^{k+1} + cv^{k+1} - x^* - cv^*). \end{aligned}$$

The rest of the proof relies on the fact that A and B are orthogonal complements and the inner products between any elements of these sets are zero.

For the second term on the right hand side of Eq. (10), we use Eq. (6) and have $\left\|x^k + cv^k - x^{k+1} - cv^{k+1}\right\|^2 = \left\|y^{k+1} + cw^{k+1} - x^{k+1} - cv^{k+1}\right\|^2$. Since x^{k+1} is the projection of y^{k+1} in A we have that $y^{k+1} - x^{k+1}$ lies in B and similarly $cw^{k+1} - cv^{k+1}$ is in A. Their inner product is zero and thus the term can be further decomposed into $\left\|x^k + cv^k - x^{k+1} - cv^{k+1}\right\|^2 = \left\|y^{k+1} - x^{k+1}\right\|^2 + \left\|cw^{k+1} - cv^{k+1}\right\|^2$.

We next analyze the inner product term on the right hand of Eq. (10). By Eq. (6), we have $x^k + cv^k = y^{k+1} + cw^{k+1}$, and thus we have $x^k + cv^k - x^{k+1} - cv^{k+1} = yk + 1 + cw^{k+1} - x^{k+1} - cv^{k+1}$. We recall that $y^{k+1} - x^{k+1}$ lies in B and $cw^{k+1} - cv^{k+1}$ is in A. We also observe that $x^{k+1} - x^*$ is in A and $cv^{k+1} - cv^*$ is in B. Combine these observations together, we have

$$\begin{aligned} &(x^k + cv^k - x^{k+1} - cv^{k+1})'(x^{k+1} + cv^{k+1} - x^* - cv^*)\\ &\quad = (y^{k+1} - x^{k+1})'(cv^{k+1} - cv^*)\\ &\qquad + (cw^{k+1} - cv^{k+1})'(x^{k+1} - x^*),\\ &\quad = (y^{k+1} - x^{k+1} + x^{k+1} - x^*)'(cv^{k+1} - cv^*)\\ &\qquad + (cw^{k+1} - cv^{k+1})'(x^{k+1} - x^* + y^{k+1} - x^{k+1}), \end{aligned}$$

where in the last equality we add terms $(x^{k+1} - x^*)'(cv^{k+1} - cv^*)$ to the first term and $(cw^{k+1} - cv^{k+1})'(y^{k+1} - x^{k+1})$ to the second term of the second equality, both of which are zero due to the orthogonality of A and B. We can now combine the terms and have

$$\begin{aligned} &(x^k + cv^k - x^{k+1} - cv^{k+1})'(x^{k+1} + cv^{k+1} - x^* - cv^*)\\ &\qquad\qquad (11)\\ &= (y^{k+1} - x^*)'(cw^{k+1} - cv^*). \end{aligned}$$

We can now combine Eqs. (10)-(11) and conclude

$$\begin{aligned} &\left\|x^{k+1} + cv^{k+1} - x^* - cv^*\right\|^2 - \left\|x^k + cv^k - x^* - cv^*\right\|^2\\ &= -\left\|y^{k+1} - x^{k+1}\right\|^2 - \left\|cw^{k+1} - cv^{k+1}\right\|^2\\ &\quad - 2(y^{k+1} - x^*)'(cw^{k+1} - cv^*). \end{aligned}$$

For the terms on the left hand side, we once again use the orthogonality of A and B, along with the fact that all x related terms are in A and v related terms are in B to break down the norm, and have

$$\begin{aligned} &\left\|x^{k+1} - x^*\right\|^2 + \left\|cv^{k+1} - cv^*\right\|^2\\ &\quad - \left\|x^k - x^*\right\|^2 - \left\|cv^k - cv^*\right\|^2\\ &= -\left\|y^{k+1} - x^{k+1}\right\|^2 - \left\|cw^{k+1} - cv^{k+1}\right\|^2\\ &\quad - 2(y^{k+1} - x^*)'(cw^{k+1} - cv^*), \end{aligned}$$

which shows Eq. (8).

To see that the sequence $\{\left\|x^k - x^*\right\|^2 + \left\|cv^k - cv^*\right\|^2\}_k$ is monotonically nonincreasing, we need to show that the inner product term in the above line satisfies $(y^{k+1} - x^*)'(cw^{k+1} - cv^*) \geq 0$, i.e., Eq. (9). We note that since w^{k+1} is in $\partial F(y^{k+1})$, we have by convexity of f_i, $(y_i^{k+1} - x_i^*)'(w_i^{k+1} - v_i^*) \geq 0$ for $i = 1, \ldots, n$. This establishes Eq. (9).

■

The previous theorem establishes that the sequence $\{\left|\left|x^k - x^*\right|\right|^2 + \left|\left|cv^k - cv^*\right|\right|^2\}_k$ is monotonically nonincreasing, we are now equipped to show convergence of the sequence $\{x^k\}$ to an optimal solution.

Theorem 3.5: Let $\{x^k\}$ be a sequence generated by Algorithm 1. Then the sequence converges to an optimal solution of problem (1).

Proof: The monotonicity results from previous theorem implies that the sequence $\{\left|\left|x^k - x^*\right|\right|^2 + \left|\left|cv^k - cv^*\right|\right|^2\}_k$ is bounded. Hence sequence $\{x^k, v^k\}_k$ has subsequent convergent sequence. We now focus on a convergent subsequent $\{x^{k_t}, v^{k_t}\}_t$ and denote its limit point as $\tilde{x}, \tilde{v}$, and the corresponding $\{y^{k_t}, w^{k_t}\}$ also converge and its limit point as $(\tilde{y}, \tilde{w})$. Eq. (8) suggests that

$$\begin{aligned}
&\left|\left|x^{k_{t+1}} - x^*\right|\right|^2 + \left|\left|cv^{k_{t+1}} - cv^*\right|\right|^2 \\
&= \left|\left|x^{k_t} - x^*\right|\right|^2 + \left|\left|cv^{k_t} - cv^*\right|\right|^2 \\
&- \sum_{k={k_t}^{k_{t+1}}} \left|\left|y^{k+1} - x^{k+1}\right|\right|^2 - \left|\left|cw^{k+1} - cv^{k+1}\right|\right|^2 \\
&- 2(y^{k+1} - x^*)'(cw^{k+1} - cv^*).
\end{aligned}$$

By Eq. (9), we have the inner product term is nonnegative, and thus

$$\begin{aligned}
&\left|\left|x^{k_t} - x^*\right|\right|^2 + \left|\left|cv^{k_t} - cv^*\right|\right|^2 - \left|\left|x^{k_{t+1}} - x^*\right|\right|^2 \\
&\quad - \left|\left|cv^{k_{t+1}} - cv^*\right|\right|^2 \geq \\
&\qquad \sum_{k=k_t}^{k_{t+1}} \left|\left|y^{k+1} - x^{k+1}\right|\right|^2 + \left|\left|cw^{k+1} - cv^{k+1}\right|\right|^2.
\end{aligned}$$

We then take limit as $t \to \infty$ on both sides and have

$$\begin{aligned}
&\lim_{t\to\infty} \left|\left|x^{k_t} - x^*\right|\right|^2 + \left|\left|cv^{k_t} - cv^*\right|\right|^2 \\
&\quad - \left|\left|x^{k_{t+1}} - x^*\right|\right|^2 - \left|\left|cv^{k_{t+1}} - cv^*\right|\right|^2 \\
&\quad \geq \lim_{t\to\infty} \sum_{k=k_t}^{k_{t+1}} \left|\left|y^{k+1} - x^{k+1}\right|\right|^2 + \left|\left|cw^{k+1} - cv^{k+1}\right|\right|^2.
\end{aligned}$$

Since the sequence $\{x^{k_t}, v^{k_t}\}$ is convergent to $(\tilde{x}, \tilde{v})$, we have

$$\begin{aligned}
&\left|\left|\tilde{x} - x^*\right|\right|^2 + \left|\left|c\tilde{v} - cv^*\right|\right|^2 - \left|\left|\tilde{x} - x^*\right|\right|^2 - \left|\left|c\tilde{v} - cv^*\right|\right|^2 \\
&\geq \lim_{t\to\infty} \sum_{k=k_t}^{k_{t+1}} \left|\left|y^{k+1} - x^{k+1}\right|\right|^2 + \left|\left|cw^{k+1} - cv^{k+1}\right|\right|^2.
\end{aligned}$$

The left hand side is 0 and each summand on the right hand side is nonnegative, therefore we have $\lim_{t\to\infty} y^{k_t} - x^{k_t} = 0$, $\lim_{k\to\infty} cw^{k_t} - cv^{k_t} = 0$, i.e., $\tilde{y} = \tilde{x}, \tilde{w} = \tilde{v}$. Hence the point $(\tilde{x}, \tilde{v})$ is a fixed point of iteration (6). We can then use $x^* = \tilde{x}$, and $v^* = \tilde{v}$ in Eq. (8). Since the value of $\left|\left|x^{k_t} - \tilde{x}\right|\right|^2 + \left|\left|cv^{k_t} - c\tilde{v}\right|\right|^2$ is going to 0 along the k_t sequence and the original sequence in k is monotone, we have $\lim_{k\to 0} \left|\left|x^k - \tilde{x}\right|\right|^2 + \left|\left|cv^k - c\tilde{v}\right|\right|^2 = 0$. Therefore sequence $\{x^k, v^k\}$ converges. The limit point of $(\tilde{x}, \tilde{v})$ is a fixed point of iteration (6) and thus by Lemma 3.3, $\tilde{x}$ is an optimal solution of problem (1). ■

We remark that the above theorem guarantees convergence of the algorithm for any stepsize choice $c > 0$.

D. Rate of Convergence

We next show that under some assumptions of the objective functions, we can establish linear rate of convergence of the algorithm, and the stepsize choice c becomes a parameter in the rate of convergence. For this section, we assume our objective functions f_i are continuously differentiable and satisfy the following assumption.

Assumption 2: Each component of the objective function f_i has Lipschitz gradient with Lipschitz constant L and is $\mu-$ strongly convex.

Note that in the event where precise values of L and μ are missing, an upper bound on L and a lower bound on μ can be used in place of L and μ for the rest of analysis. The following lemma relates $\left|\left|y^{k+1} - x^{k+1}\right|\right|^2 + \left|\left|cw^{k+1} - cv^{k+1}\right|\right|^2 + 2(y^{k+1} - x^*)'(cw^{k+1} - cv^*)$ to $\left|\left|x^{k+1} - x^*\right|\right|^2 + \left|\left|cv^{k+1} - cv^*\right|\right|^2$. We later combine this lemma with Theorem 3.4 to show linear convergence rate.

Lemma 3.6: For any sequence of x^k, v^k, y^k, w^k generated by Algorithm 1, we have that

$$\begin{aligned}
&\left|\left|y^{k+1} - x^{k+1}\right|\right|^2 + \left|\left|cw^{k+1} - cv^{k+1}\right|\right|^2 \\
&\quad + 2(y^{k+1} - x^*)'(cw^{k+1} - cv^*) \\
&\geq \min\left\{\frac{1}{2}, \frac{c\mu\beta}{2}\right\} \left|\left|x^{k+1} - x^*\right|\right|^2 \\
&\quad + \min\left\{\frac{1}{2}, \frac{\mu(1-\beta)}{2cL^2}\right\} \left|\left|cv^{k+1} - cv^*\right|\right|^2
\end{aligned} \tag{12}$$

for any β in $(0, 1)$.

Proof: We first focus on the inner product term on the left hand side, then later combine it with the rest of the norm terms. Since each f_i is differentiable, the vector w^{k+1} is composed of gradient vectors, i.e., $w^{k+1} = [\nabla f_i(y_i^{k+1})]_i$ and $v^* = [\nabla f_i(x_i^*)]_i$. By strong convexity of f_i in Assumption 2, we have by [14], $(y_i - x_i^*)'(\nabla f_i(y_i) - \nabla f_i(x_i^*)) \geq \mu \left|\left|y_i - x_i^*\right|\right|^2$, for any y_i in $\mathbb{R}^m$. For the long vector in $\mathbb{R}^{nm}$, we therefore have

$$\begin{aligned}
&(y^{k+1} - x^*)'(cw^{k+1} - cv^*) \\
&= c\sum_{i=1}^{n} (y_i^{k+1} - x_i^*)'(\nabla f_i(y_i^{k+1}) - \nabla f_i(x_i^*)) \\
&\geq c\mu \sum_{i=1}^{n} \left|\left|y_i^{k+1} - x_i^*\right|\right|^2 = c\mu \left|\left|y^{k+1} - x^*\right|\right|^2.
\end{aligned} \tag{13}$$

We also note that due to the fact that each f_i has Lipschitz gradient with Lipschitz constant L, we have by results in [14], $\left|\left|\nabla f_i(y_i) - \nabla f_i(x_i^*)\right|\right| \leq L \left|\left|y_i - x_i^*\right|\right|$, for any y_i in $\mathbb{R}^m$. Therefore, we have

$$\begin{aligned}
&\left|\left|y^{k+1} - x^*\right|\right|^2 = \sum_{i=1}^{n} \left|\left|y_i^{k+1} - x_i^*\right|\right|^2 \\
&\geq \frac{1}{L^2} \sum_{i=1}^{n} \left|\left|\nabla f_i(y_i^{k+1}) - \nabla f_i(x_i^*)\right|\right|^2 = \frac{1}{L^2} \left|\left|w^{k+1} - v^*\right|\right|^2.
\end{aligned}$$

Thus we can introduce the factor β in $(0, 1)$ to Eq. (13) and have $(y^{k+1} - x^*)'(cw^{k+1} - cv^*) \geq$

$c\mu\beta \left|\left|y^{k+1}-x^*\right|\right|^2 + c\mu(1-\beta)\left|\left|y^{k+1}-x^*\right|\right|^2 \geq c\mu\beta\left|\left|y^{k+1}-x^*\right|\right|^2 + \frac{\mu(1-\beta)}{cL^2}\left|\left|cw^{k+1}-cv^*\right|\right|^2$.

We can now bring in the norm terms and have

$$\begin{aligned}
&\left|\left|y^{k+1}-x^{k+1}\right|\right|^2 + \left|\left|cw^{k+1}-cv^{k+1}\right|\right|^2 \\
&\quad + 2(y^{k+1}-x^*)'(cw^{k+1}-cv^*) \\
&\geq \left|\left|y^{k+1}-x^{k+1}\right|\right|^2 + c\mu\beta\left|\left|y^{k+1}-x^*\right|\right|^2 \\
&\quad + \left|\left|cw^{k+1}-cv^{k+1}\right|\right|^2 + \frac{\mu(1-\beta)}{cL^2}\left|\left|cw^{k+1}-cv^*\right|\right|^2
\end{aligned}$$

We next use the inequality [2] that $||a||^2 + ||b||^2 \geq \frac{1}{2}||a+b||^2$, and have

$$\begin{aligned}
&\left|\left|y^{k+1}-x^{k+1}\right|\right|^2 + c\mu\beta\left|\left|y^{k+1}-x^*\right|\right|^2 \\
&\geq \min\left\{\frac{1}{2}, \frac{c\mu\beta}{2}\right\}\left|\left|x^{k+1}-x^*\right|\right|^2,
\end{aligned}$$

$$\begin{aligned}
&\left|\left|cw^{k+1}-cv^{k+1}\right|\right|^2 + \frac{\mu(1-\beta)}{cL^2}\left|\left|cw^{k+1}-cv^*\right|\right|^2 \\
&\geq \min\left\{\frac{1}{2}, \frac{\mu(1-\beta)}{2cL^2}\right\}\left|\left|cv^{k+1}-cv^*\right|\right|^2.
\end{aligned}$$

By combining the previous three inequalities, we obtain Eq. (12). ■

We next show linear rate of convergence.

Theorem 3.7: For any sequence of x^k, v^k generated by Algorithm 1, we have that for any β in $(0,1)$,

$$\begin{aligned}
&\left(1+\min\left\{\frac{1}{2}, \frac{c\mu\beta}{2}\right\}\right)\left|\left|x^{k+1}-x^*\right|\right|^2 \\
&\quad + \left(1+\min\left\{\frac{1}{2}, \frac{\mu(1-\beta)}{2cL^2}\right\}\right)\left|\left|cv^{k+1}-cv^*\right|\right|^2 \\
&\leq \left|\left|x^k-x^*\right|\right|^2 + \left|\left|cv^k-cv^*\right|\right|^2.
\end{aligned}$$

Proof: Recall Eqs. (8) and (12)

$$\begin{aligned}
&\left|\left|x^k-x^*\right|\right|^2 + \left|\left|cv^k-cv^*\right|\right|^2 - \left|\left|x^{k+1}-x^*\right|\right|^2 - \\
&\left|\left|cv^{k+1}-cv^*\right|\right|^2 = \left|\left|y^{k+1}-x^{k+1}\right|\right|^2 + \left|\left|cw^{k+1}-cv^{k+1}\right|\right|^2 \\
&+ 2(y^{k+1}-x^*)'(cw^{k+1}-cv^*),
\end{aligned}$$

and

$$\begin{aligned}
&\left|\left|y^{k+1}-x^{k+1}\right|\right|^2 + \left|\left|cw^{k+1}-cv^{k+1}\right|\right|^2 \\
&+ 2(y^{k+1}-x^*)'(cw^{k+1}-cv^*) \geq \\
&\min\left\{\frac{1}{2}, \frac{c\mu\beta}{2}\right\}\left|\left|x^{k+1}-x^*\right|\right|^2 \\
&+ \min\left\{\frac{1}{2}, \frac{\mu(1-\beta)}{2cL^2}\right\}\left|\left|cv^{k+1}-cv^*\right|\right|^2.
\end{aligned}$$

Hence, we can combine the previous two lines and establish the desired relation. ■

The above theorem establishes linear convergence rate for the algorithm. To match the two constants, we can set $\frac{c\mu\beta}{2} = \frac{\mu(1-\beta)}{2cL^2}$, and have $c = \sqrt{\frac{1-\beta}{\beta}}\frac{1}{L}$. This choice of c gives $\frac{c\mu\beta}{2} = \frac{\mu(1-\beta)}{2cL^2} = \sqrt{(1-\beta)\beta}\frac{\mu}{2L}$. This value is maximized at $\beta = \frac{1}{2}$. We can then have $\left(1+\min\left\{\frac{1}{2}, \frac{\mu}{4L}\right\}\right)\left[\left|\left|x^{k+1}-x^*\right|\right|^2 + \left|\left|cv^{k+1}-cv^*\right|\right|^2\right] \leq \left|\left|x^k-x^*\right|\right|^2 + \left|\left|cv^k-cv^*\right|\right|^2$. For problems with $\frac{\mu}{L} > 2$, we have

$$\begin{aligned}
&\left|\left|x^{k+1}-x^*\right|\right|^2 + \left|\left|cv^{k+1}-cv^*\right|\right|^2 \\
&\qquad \leq \frac{2}{3}\left[\left|\left|x^k-x^*\right|\right|^2 + \left|\left|cv^k-cv^*\right|\right|^2\right].
\end{aligned}$$

For problems with $\frac{\mu}{L} \leq 2$, we have

$$\begin{aligned}
&\left|\left|x^{k+1}-x^*\right|\right|^2 + \left|\left|cv^{k+1}-cv^*\right|\right|^2 \\
&\qquad \leq \frac{4}{4+\kappa}\left[\left|\left|x^k-x^*\right|\right|^2 + \left|\left|cv^k-cv^*\right|\right|^2\right],
\end{aligned}$$

where $\kappa = \frac{\mu}{L}$. We conclude that the rate of linear convergence depends on the condition number of the objective functions.

IV. Conclusions

In this paper, we propose a parallel multi-splitting proximal method and show that it converges for any positive stepsize. When the objective functions are Lipschitz gradient and strongly convex, the algorithm converges linearly. Future works include extend this algorithm to stochastic setting where delays and errors are involved.

References

[1] D. P. Bertsekas. *Nonlinear Programming*. Athena Scientific, 1999.

[2] D. P. Bertsekas. Incremental Gradient, Subgradient, and Proximal Methods for Convex Optimization: A Survey. *LIDS Report 2848*, 2010.

[3] D. P. Bertsekas and J. N. Tsitsiklis. *Parallel and Distributed Computation: Numerical Methods*. Athena Scientific, Belmont, MA, 1997.

[4] 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*, 3(1):1–122, 2010.

[5] Caihua Chen, Bingsheng He, Yinyu Ye, and Xiaoming Yuan. The direct extension of admm for multi-block convex minimization problems is not necessarily convergent. *Mathematical Programming*, 155(1-2):57–79, 2016.

[6] Aaron Defazio, Francis Bach, and Simon Lacoste-Julien. Saga: A fast incremental gradient method with support for non-strongly convex composite objectives. In *Advances in Neural Information Processing Systems*, pages 1646–1654, 2014.

[7] J. Eckstein. Augmented Lagrangian and Alternating Direction Methods for Convex Optimization: A Tutorial and Some Illustrative Computational Results. *Rutcor Research Report*, 2012.

[8] Jonathan Eckstein and Benar Fux Svaiter. General projective splitting methods for sums of maximal monotone operators. *SIAM Journal on Control and Optimization*, 48(2):787–811, 2009.

[9] Pontus Giselsson and Stephen Boyd. Diagonal scaling in douglas-rachford splitting and admm. In *Decision and Control (CDC), 2014 IEEE 53rd Annual Conference on*, pages 5033–5039. IEEE, 2014.

[10] Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in Neural Information Processing Systems*, pages 315–323, 2013.

[11] J. Mota, J Xavier, P. Aguiar, and M. Püschel. ADMM For Consensus On Colored Networks. *Proceedings of IEEE Conference on Decision and Control (CDC)*, 2012.

[12] J. Mota, J. Xavier, P. Aguiar, and M. Püschel. D-ADMM : A Communication-Efficient Distributed Algorithm For Separable Optimization. *IEEE Transactions on Signal Processing*, 61(10):2718–2723, 2013.

[2] To see why this inequality is true, we have $||a||^2 + ||b||^2 - 2a'b = ||a-b||^2 \geq 0$. Therefore, $\frac{1}{2}[||a||^2 + ||b||^2] \geq a'b$, which implies that $||a||^2 + ||b||^2 \geq \frac{1}{2}||a||^2 + \frac{1}{2}||b||^2 + a'b = \frac{1}{2}||a+b||^2$.

[13] A. Nedić and A. Ozdaglar. Distributed subgradient methods for multi-agent optimization. *Automatic Control, IEEE Transactions on*, 54(1):48 –61, Jan 2009.

[14] Yurii Nesterov. *Introductory lectures on convex optimization: A basic course*, volume 87. Springer Science & Business Media, 2013.

[15] Neal Parikh and Stephen P Boyd. Proximal algorithms. *Foundations and Trends in optimization*, 1(3):127–239, 2014.

[16] I. D. Schizas, R. Ribeiro, and G. B. Giannakis. Consensus in Ad Hoc WSNs with Noisy Links - Part I: Distributed Estimation of Deterministic Signals. *IEEE Transactions on Singal Processing*, 56:350–364, 2008.

[17] Mark Schmidt, Nicolas Le Roux, and Francis Bach. Minimizing finite sums with the stochastic average gradient. *arXiv preprint arXiv:1309.2388*, 2013.

[18] Wei Shi, Qing Ling, Gang Wu, and Wotao Yin. Extra: An exact first-order algorithm for decentralized consensus optimization. *SIAM Journal on Optimization*, 25(2):944–966, 2015.

[19] Jonathan E Spingarn. Partial inverse of a monotone operator. *Applied mathematics and optimization*, 10(1):247–265, 1983.

[20] J. N. Tsitsiklis. *Problems in Decentralized Decision Making and Computation*. PhD thesis, Massachusetts Institute of Technology, 1984.

[21] E. Wei and A. Ozdaglar. Distributed Alternating Direction Method of Multipliers. *Proceedings of IEEE Conference on Decision and Control (CDC)*, 2012.

[22] E. Wei and A. Ozdaglar. On the $O(1/k)$ convergence of asynchronous distributed alternating Direction Method of Multipliers. In *Global Conference on Signal and Information Processing (GlobalSIP), 2013 IEEE*, pages 551–554. IEEE, 2013.

[23] Lin Xiao and Tong Zhang. A proximal stochastic gradient method with progressive variance reduction. *SIAM Journal on Optimization*, 24(4):2057–2075, 2014.

[24] H. Zhu, A. Cano, and G. B. Giannakis. In-Network Channel Decoding Using Consensus on Log-Likelihood Ratio Averages. *Proceedings of Conference on Information Sciences and Systems (CISS)*, 2008.