

A Chebyshev-Accelerated Primal-Dual Method for Distributed Optimization

Jacob H. Seidman¹, Mahyar Fazlyab², George J. Pappas², Victor M. Preciado²

Abstract—We consider a distributed optimization problem over a network of agents aiming to minimize a global objective function that is the sum of local convex and composite cost functions. To this end, we propose a distributed Chebyshev-accelerated primal-dual algorithm to achieve faster ergodic convergence rates. In standard distributed primal-dual algorithms, the speed of convergence towards a global optimum (i.e., a saddle point in the corresponding Lagrangian function) is directly influenced by the eigenvalues of the Laplacian matrix representing the communication graph. In this paper, we use Chebyshev matrix polynomials to generate gossip matrices whose spectral properties result in faster convergence speeds, while allowing for a fully distributed implementation. As a result, the proposed algorithm requires fewer gradient updates at the cost of additional rounds of communications between agents. We illustrate the performance of the proposed algorithm in a distributed signal recovery problem. Our simulations show how the use of Chebyshev matrix polynomials can be used to improve the convergence speed of a primal-dual algorithm over communication networks, especially in networks with poor spectral properties, by trading local computation by communication rounds.

I. INTRODUCTION

In distributed consensus optimization, a network of agents aims to minimize a function constructed from individual (private) functions situated at the nodes of the network. As the size of the network increases, it is important to develop fast and efficient distributed methods where agents can perform local computations to collectively find the global optimal solution. This scenario arises in, for example, distributed learning problems [1], economic dispatch problems [2], and multi-agent coordination problems [3].

Methods to solve such distributed optimization problems fall into three main categories: (i) primal methods, in which the problem is solved entirely in the primal domain and asymptotic consensus among agents is forced using an additional penalty function representing disagreements among agents [4], [5]; (ii) primal-dual methods, in which the consensus constraint is relaxed by introducing Lagrangian multipliers and the agents iterate over the primal and dual variables to seek a saddle point of the Lagrangian [6]–[10]; and (iii) dual methods, in which the constraints are encoded into a dual problem and the agents seek the optimum of a dual function instead [11]–[14]. The advantages of each of these methods over the others are determined by what assumptions are made about the objective function (smoothness, strong convexity, easily computable proximal

operators, etc.), as well as how any additional constraints are encoded in the problem.

In general, distributed first-order algorithms where agents update their local states based exclusively on the gradients of the cost functions can become slow for problems with ill-conditioned linear constraints. In particular, the convergence speed of these algorithms is strongly conditioned by the second smallest eigenvalue of the Laplacian matrix of the communication graph. The performance of distributed algorithms can be improved by preconditioning the matrix representing the topology of the communication graph. For example, in [15] the authors successfully applied this preconditioning technique to a dual gradient method. In the numerical analysis literature, it has been known that condition numbers of linear operators can also be improved by applying specifically scaled and shifted Chebyshev polynomials [16]. The first applications of this idea to a linear consensus constraint can be found in [17] and [18], where the use of Chebyshev polynomials gave a prescription for averaging information received from an agent's neighbors over multiple rounds of communication. It was shown that this results in faster consensus over traditional choices of gossip matrices. One interpretation of this method is that, by decreasing the condition number of the gossip matrix, we effectively create a more favorable connectivity structure of the network.

In [19], the authors combined this idea with an accelerated dual ascent to create the *multi-step dual accelerated* (MSDA) method for solving smooth and strongly convex distributed optimization problems. By preconditioning the gossip matrix with a particular Chebyshev polynomial, the authors demonstrate that the effective condition number of the network decreases, and faster convergence for dual ascent is achieved. In practice, when this algorithm is implemented, the cost of applying a Chebyshev polynomial of degree K to the gossip matrix is that each iteration of the algorithm now requires K rounds of communication. When the cost of communication is cheap compared to the cost of computing gradients, their algorithm performs well even with the extra rounds of communication.

While the performance advantages of MSDA are clear, the performance guarantees of the algorithm only apply in the case of smooth and strongly convex objective functions. Further, there is the hidden complexity of the computation of the dual gradients which either must be done analytically or through an inner minimization. In cases where an analytic expression is not possible or the inner minimization is costly, a primal-dual scheme might be preferable.

In this paper, we propose a Chebyshev-Accelerated

1: Department of Applied Mathematics and Computational Science, University of Pennsylvania Email: seidj@sas.upenn.edu. 2: Department of Electrical and Systems Engineering, University of Pennsylvania. Email: {mahyarfa, pappasg, preciado}@seas.upenn.edu. Work was supported by the NSF under grants CAREER-ECCS-1651433 and IIS-1447470.

Primal-Dual method for solving a distributed optimization problem with an objective function of the form $\sum_i f_i + \sum_i g_i$, where each f_i is smooth and convex with Lipschitz gradient, and each g_i is convex and non-differentiable. Our algorithm is based on the primal-dual method proposed in [20], and later extended in [10], which was developed to minimize a similar composite objective function. In Theorem 1, we show that the largest possible primal and dual step sizes for this method are inversely related to the largest eigenvalue of the Laplacian matrix. This motivates preconditioning the Laplacian matrix to reduce the spectral radius and allow the use of larger step sizes. To this end, we combine the primal-dual method with an accelerated gossip strategy with Chebyshev polynomials from [17]–[19]. Our algorithm reaches consensus more quickly than the standard primal-dual algorithm, and results in faster convergence as measured by the number of gradient updates. We illustrate the performance of this algorithm with a distributed signal recovery problem and find that when the cost of communications are low compared to the cost of computing gradients, there is a noticeable increase in the convergence rate to the optimal value.

A. Preliminaries

We denote the set of real numbers by \mathbb{R} , the set of real d -dimensional vectors by \mathbb{R}^d , the d -dimensional vector of 1's and 0's by $\mathbf{1}_d$ and $\mathbf{0}_d$ respectively, and the d -dimensional identity matrix by \mathbf{I}_d . A graph is defined as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where \mathcal{V} is the set of vertices, and the edge set \mathcal{E} contains pairs of vertices (i, j) such that $(i, j) \in \mathcal{E}$ if and only if vertices i and j are connected in the graph. Define the neighborhood set of a vertex $\mathcal{N}_i = \{\ell \in \mathcal{V} \mid (i, \ell) \in \mathcal{E}\}$. We assume here that $|\mathcal{V}| = n$ and the graph \mathcal{G} is simple, connected, and undirected. We define the adjacency matrix $A = [a_{ij}]$ of \mathcal{G} as $a_{ij} = 1$ if $(i, j) \in \mathcal{E}$ and $a_{ij} = 0$ otherwise. The Laplacian matrix of \mathcal{G} is defined as $L = \text{diag}(A\mathbf{1}_n) - A$. The matrix L is symmetric and positive semidefinite with eigenvalues $0 = \lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n$, where the eigenvalue $\lambda_1 = 0$ corresponds to the eigenvector $\mathbf{1}_n$. If we take the eigenvalue decomposition $L = U \text{diag}(0, \lambda_2, \dots, \lambda_n) U^\top$ where U is orthogonal, we may form the matrix $\sqrt{L} = U \text{diag}(0, \sqrt{\lambda_2}, \dots, \sqrt{\lambda_n}) U^\top$. By $\|L\|_2$ we denote the ℓ_2 matrix norm (which coincides with the spectral radius for symmetric matrices), and we denote the kernel of the matrix L as $\ker(L) = \text{span}\{\mathbf{1}_n\}$. The Kronecker product between two matrices or vectors is denoted $A \otimes B$ or $v \otimes w$, respectively. A differentiable function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ has a L_f -Lipschitz gradient if, for all $x, y \in \mathbb{R}^d$, we have

$$\|\nabla f(x) - \nabla f(y)\| \leq L_f \|x - y\|.$$

For a non-differentiable $g : \mathbb{R}^d \rightarrow \mathbb{R} \cup \{\infty\}$ with $\text{dom } g = \{x \in \mathbb{R}^d : g(x) < \infty\}$, the subdifferential set is defined as

$$\partial g(x) := \{\gamma \in \mathbb{R}^d : g(x) + \gamma^\top (y - x) \leq g(y), \forall y \in \text{dom } g\}.$$

II. PROBLEM FORMULATION

Consider a network of n agents connected via a communication graph, $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where the edge set \mathcal{E} represents

communication links between agents. Each agent has access to private cost functions $f_i : \mathbb{R}^d \rightarrow \mathbb{R}$ and $g_i : \mathbb{R}^d \rightarrow \mathbb{R} \cup \{\infty\}$. The agents seek to collectively find an optimal solution to the minimization problem

$$\underset{x \in \mathbb{R}^d}{\text{minimize}} \quad \sum_{i=1}^n \{f_i(x) + g_i(x)\}, \quad (1)$$

in a *distributed* fashion. We make the following assumptions on the functions f_i and g_i .

Assumption 1 *The functions $f_i : \mathbb{R}^d \rightarrow \mathbb{R}$ are closed, convex, proper, and have L_i -Lipschitz gradients. The functions $g_i : \mathbb{R}^d \rightarrow \mathbb{R} \cup \{\infty\}$ are proper, convex, lower semi-continuous. The proximal maps*

$$\text{prox}_{\alpha g_i}(x) := \underset{x' \in \mathbb{R}^d}{\text{argmin}} \left\{ g_i(x') + \frac{1}{2\alpha} \|x - x'\|_2^2 \right\}. \quad (2)$$

are easily computable.

Examples of functions with an easily computable proximal operator are $g(x) = \|x\|_1$, $g(x) = \|x\|_\infty$ [21], [22]. Further, if $g(x) = \mathbb{I}_{\mathcal{X}}(x)$ is the indicator function of the closed convex set $\mathcal{X} \subset \mathbb{R}^d$, the proximal operator becomes a projection operator onto \mathcal{X} . In this case, we assume that projection onto \mathcal{X} is easy to compute (e.g. box constraints).

We can reformulate problem (1) to be amenable to a distributed solution by introducing a set of variables $x_i \in \mathbb{R}^d$ for $i \in \mathcal{V}$, where x_i corresponds to a local copy of the global state variable x for agent i . Let $\mathbf{x} = (x_1^\top, \dots, x_n^\top)^\top$ be the concatenation of the local copies of the state variable. If we define a gossip matrix $\mathbf{L} = L \otimes \mathbf{I}_d$, using the Laplacian matrix L of the communication graph, we can ensure consensus (i.e. $x_i = x_j$ for all $i, j \in \mathcal{V}$) by adding the constraint $\sqrt{L}\mathbf{x} = 0$, as $\ker(\sqrt{L} \otimes \mathbf{I}_d) = \text{span}\{\mathbf{1}_n \otimes x \mid x \in \mathbb{R}^d\}$. Thus, the problem in (1) is equivalent to

$$\begin{aligned} & \underset{\mathbf{x} \in \mathbb{R}^{nd}}{\text{minimize}} && f(\mathbf{x}) + g(\mathbf{x}) \\ & \text{subject to} && \sqrt{L}\mathbf{x} = 0, \end{aligned} \quad (3)$$

where $f : \mathbb{R}^{nd} \rightarrow \mathbb{R}$ and $g : \mathbb{R}^{nd} \rightarrow \mathbb{R} \cup \{\infty\}$ are now

$$f(\mathbf{x}) := \sum_{i=1}^n f_i(x_i), \quad g(\mathbf{x}) := \sum_{i=1}^n g_i(x_i). \quad (4)$$

We form the augmented Lagrangian for (3),

$$\mathcal{L}(\mathbf{x}, \mathbf{y}) := f(\mathbf{x}) + g(\mathbf{x}) + \mathbf{y}^\top \sqrt{L}\mathbf{x} + \frac{\rho}{2} \mathbf{x}^\top L \mathbf{x}, \quad (5)$$

where $\mathbf{y} = (y_1^\top, \dots, y_n^\top)^\top$ is the stacked vector of local multipliers and $\rho \geq 0$ is the augmentation parameter. Without loss of generality, we may assume that $\mathbf{y} \in (\ker \sqrt{L})^\perp$, as for any $\tilde{\mathbf{y}} \in \ker \sqrt{L}$ we have $(\mathbf{y} + \tilde{\mathbf{y}})^\top \sqrt{L} = \mathbf{y}^\top \sqrt{L}$. Since problem (3) has linear constraints and the feasible set is nonempty, Slater's condition is satisfied and strong duality holds [23]. Hence, we can solve (3) by solving the following saddle point problem:

$$\max_{\mathbf{y} \in (\ker \sqrt{L})^\perp} \min_{\mathbf{x} \in \mathbb{R}^{nd}} \mathcal{L}(\mathbf{x}, \mathbf{y}). \quad (6)$$

A solution of the saddle point problem is a primal dual pair $(\mathbf{x}^*, \mathbf{y}^*)$ such that for all (\mathbf{x}, \mathbf{y}) ,

$$\mathcal{L}(\mathbf{x}^*, \mathbf{y}) \leq \mathcal{L}(\mathbf{x}, \mathbf{y}) \leq \mathcal{L}(\mathbf{x}, \mathbf{y}^*). \quad (7)$$

Note that $\mathcal{L}(\mathbf{x}, \mathbf{y}^*) - \mathcal{L}(\mathbf{x}^*, \mathbf{y})$ approaches 0 as (\mathbf{x}, \mathbf{y}) approaches $(\mathbf{x}^*, \mathbf{y}^*)$. The optimality condition for an optimal primal-dual pair $(\mathbf{x}^*, \mathbf{y}^*)$ is

$$0 = \nabla f(\mathbf{x}^*) + T_g(\mathbf{x}^*) + \sqrt{\mathbf{L}}\mathbf{y}^*, \quad (8a)$$

$$0 = \sqrt{\mathbf{L}}\mathbf{x}^*, \quad (8b)$$

where $T_g(\mathbf{x}^*)$ is any subgradient of g evaluated at \mathbf{x}^* . Note that, by the second equation, we have consensus, $\mathbf{x}^* = \mathbf{1}_n \otimes x^*$. Multiplying both sides of (8a) from the left by $\mathbf{1}_n^\top \otimes \mathbf{I}_d$, and noticing that $(\mathbf{1}_n^\top \otimes \mathbf{I}_d) \sqrt{\mathbf{L}} = (\mathbf{1}_n^\top \otimes \mathbf{I}_d)(\sqrt{\mathbf{L}} \otimes \mathbf{I}_d) = (\mathbf{1}_n^\top \sqrt{\mathbf{L}}) \otimes (\mathbf{I}_d \mathbf{I}_d) = \mathbf{0}_n^\top \otimes \mathbf{I}_d$, we obtain

$$\sum_{i=1}^n \{\nabla f_i(x^*) + T_{g_i}(x^*)\} = 0, \quad T_{g_i}(x^*) \in \partial g_i(x^*),$$

which is the optimality condition for the centralized problem in (1). Hence, the solution of the saddle point problem for the decentralized problem in (3) provides a solution of the centralized problem (1).

III. A DISTRIBUTED SOLUTION

A. The Primal-Dual Algorithm

We begin our exposition by presenting a primal-dual iterative algorithm proposed by Chambolle and Pock [20, Algorithm 1], which is able to solve problems of the form (6). Writing $\nabla f(\mathbf{x}) = (\nabla f_1(x_1)^\top, \dots, \nabla f_n(x_n)^\top)^\top$, the iterations of this algorithm are given by

$$\hat{\mathbf{x}}^{k+1} = \mathbf{x}^k - \alpha(\nabla f(\mathbf{x}^k) + \sqrt{\mathbf{L}}\mathbf{y}^k + \rho\mathbf{L}\mathbf{x}^k), \quad (9a)$$

$$\mathbf{x}^{k+1} = \underset{\mathbf{x}}{\operatorname{argmin}} \left\{ g(\mathbf{x}) + \frac{1}{2\alpha} \|\mathbf{x} - \hat{\mathbf{x}}^{k+1}\|_2^2 \right\}, \quad (9b)$$

$$\mathbf{y}^{k+1} = \mathbf{y}^k + \beta\sqrt{\mathbf{L}}(2\mathbf{x}^{k+1} - \mathbf{x}^k), \quad (9c)$$

where we have interpreted the augmentation term as being absorbed into the smooth part of the objective function. The parameters $\alpha, \beta > 0$ are the primal and dual step sizes, respectively. The first two updates constitute a proximal gradient step applied to the Lagrangian in the primal domain, while the third recursion is a gradient ascent-like step in the dual domain.

The following theorem shows that under appropriate selection of the step sizes, the iterates in (9) achieve an ergodic $O(1/N)$ convergence rate. An important fact for our development is that the maximum possible step sizes for convergence are inversely related to the largest eigenvalue of the Laplacian matrix, as stated below.

Theorem 1 Consider the optimization problem in (3), where f_i and g_i satisfy Assumption 1. Let the step sizes α and β for the recursions in (9) satisfy

$$\frac{1}{\beta + \rho} \left(\frac{1}{\alpha} - L_f \right) \geq \lambda_n, \quad (10)$$

where $L_f = \max_i L_i$. Let $\{(\mathbf{x}^k, \mathbf{y}^k)\}_{k=1}^N$ be the sequence of points produced by Algorithm 1 with $\mathbf{x}^0 \in \ker \sqrt{\mathbf{L}}$ and $\mathbf{y}^0 \in \ker(\sqrt{\mathbf{L}})^\perp$. Then we have

$$\mathcal{L}(\bar{\mathbf{x}}^N, \mathbf{y}^*) - \mathcal{L}(\mathbf{x}^*, \bar{\mathbf{y}}^N) \leq \frac{1}{N} \left(\frac{\|\mathbf{x}^* - \mathbf{x}^0\|_2}{2\alpha} + \frac{\|\mathbf{y}^* - \mathbf{y}^0\|_2}{2\beta} \right), \quad (11)$$

where $\bar{\mathbf{x}}^N = \frac{1}{N} \sum_{k=1}^N \mathbf{x}^k$ and $\bar{\mathbf{y}}^N = \frac{1}{N} \sum_{k=1}^N \mathbf{y}^k$.

Proof: First note that $L_f = \max\{L_1, \dots, L_n\}$ is an upper bound for the Lipschitz constant for $\nabla f(\mathbf{x})$. Indeed,

$$\begin{aligned} \|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\|_2^2 &= \sum_{i=1}^n \|\nabla f_i(x_i) - \nabla f_i(y_i)\|_2^2 \\ &\leq \sum_{i=1}^n L_i^2 \|x_i - y_i\|_2^2 \\ &\leq L_f^2 \|\mathbf{x} - \mathbf{y}\|_2^2. \end{aligned} \quad (12)$$

Again, we will interpret the augmentation term as being absorbed into the smooth term of the objective function. Thus, an upper bound for the Lipschitz constant for $\nabla(f(\mathbf{x}) + (\rho/2)\mathbf{x}^\top \mathbf{L}\mathbf{x})$ is $L_F := L_f + \rho\lambda_n$.

To prove the theorem we appeal to [10, Theorem 1.1], where the problem in question is similar to this paper, but with an additional private cone constraint for each agent. Since we have no such constraints, in our scenario the theorem states if

$$\begin{bmatrix} \left(\frac{1}{\alpha} - L_F\right) \mathbf{I}_{nd} & 0 & -\sqrt{\mathbf{L}} \\ 0 & 0 & 0 \\ -\sqrt{\mathbf{L}} & 0 & \frac{1}{\beta} \mathbf{I}_{nd} \end{bmatrix} \succeq 0, \quad (13)$$

then we have for all (\mathbf{x}, \mathbf{y})

$$\begin{aligned} \mathcal{L}(\bar{\mathbf{x}}^N, \mathbf{y}) - \mathcal{L}(\mathbf{x}, \bar{\mathbf{y}}^N) &\leq \frac{1}{N} \left(\frac{1}{2\alpha} \|\mathbf{x} - \mathbf{x}^0\|_2 + \frac{1}{2\beta} \|\mathbf{y} - \mathbf{y}^0\|_2 \right. \\ &\quad \left. + (\mathbf{y} - \mathbf{y}^0)^\top \sqrt{\mathbf{L}}(\mathbf{x} - \mathbf{x}^0) \right). \end{aligned} \quad (14)$$

From the Schur complement, (13) is equivalent to the condition that $\left(\frac{1}{\alpha} - L_F\right) \mathbf{I}_{nd} - \beta\mathbf{L} \succeq 0$, or

$$\frac{1}{\beta} \left(\frac{1}{\alpha} - L_F \right) \mathbf{I}_{nd} \succeq \mathbf{L}, \quad (15)$$

Since the \mathbf{L} and \mathbf{L} share the same eigenvalues, and the largest eigenvalue of \mathbf{L} is equal to λ_n , if

$$\frac{1}{\beta} \left(\frac{1}{\alpha} - L_F \right) \geq \lambda_n \quad (16)$$

then (15) and in turn (13) are satisfied, which by Theorem 1.1 of [10] implies (14). Recalling that $L_F = L_f + \rho\lambda_n$, a simple rearrangement of (16) gives (10).

If we choose \mathbf{x}^0 such that $\sqrt{\mathbf{L}}\mathbf{x}^0 = 0$ (i.e., initialize \mathbf{x}^0 to have consensus) and take $(\mathbf{x}, \mathbf{y}) = (\mathbf{x}^*, \mathbf{y}^*)$ in (14), then $\sqrt{\mathbf{L}}(\mathbf{x}^* - \mathbf{x}^0) = 0$, and we conclude

$$\mathcal{L}(\bar{\mathbf{x}}^N, \mathbf{y}^*) - \mathcal{L}(\mathbf{x}^*, \bar{\mathbf{y}}^N) \leq \frac{1}{N} \left(\frac{\|\mathbf{x}^* - \mathbf{x}^0\|_2}{2\alpha} + \frac{\|\mathbf{y}^* - \mathbf{y}^0\|_2}{2\beta} \right),$$

as desired. ■

Remark 1 If we do not require $\mathbf{x}_0 \in \ker \sqrt{\mathbf{L}}$, the convergence rate holds with an additional constant factor of $(\mathbf{y}^* - \mathbf{y}_0)^\top \sqrt{\mathbf{L}}(\mathbf{x}^* - \mathbf{x}_0)$ in the parenthesis in (11).

Corollary 1 If the dual step size is chosen as

$$\beta = \frac{1}{\lambda_n} \left(\frac{1}{\alpha} - L_f \right) - \rho, \quad (17)$$

then we have the following ergodic convergence rate toward consensus

$$\|\bar{\mathbf{x}}^N - \bar{\mathbf{x}}_C^N\|_2^2 \leq \frac{1}{N} \left[\frac{L_f + \rho\lambda_n}{\lambda_2} \|\mathbf{x}^0 - \mathbf{x}^*\|_2 + \frac{\lambda_n}{\lambda_2} \frac{1}{L_f + \rho\lambda_n} \|\mathbf{y}^0 - \mathbf{y}^*\|_2 \right], \quad (18)$$

where

$$\bar{\mathbf{x}}_C^N = \sum_{i=1}^d (1 \otimes e_i)^\top \bar{\mathbf{x}}^N (1 \otimes e_i),$$

is the projection of $\bar{\mathbf{x}}^N$ onto the consensus subspace.

Proof: From the definition of the augmented Lagrangian, (5), we write

$$\begin{aligned} \mathcal{L}(\bar{\mathbf{x}}^N, \mathbf{y}) - \mathcal{L}(\mathbf{x}, \bar{\mathbf{y}}^N) &= f(\bar{\mathbf{x}}^N) - f(\mathbf{x}^*) + g(\bar{\mathbf{x}}^N) - g(\mathbf{x}^*) \\ &\quad + \mathbf{y}^* \sqrt{\mathbf{L}} \bar{\mathbf{x}}^N + (\rho/2) (\bar{\mathbf{x}}^N)^\top \mathbf{L} \bar{\mathbf{x}}^N \end{aligned} \quad (19)$$

Using (8a) and (8b), we have the relation

$$\mathbf{y}^* \sqrt{\mathbf{L}} \bar{\mathbf{x}}^N = -(\nabla f(\mathbf{x}^*) + T_g(\mathbf{x}^*))^\top (\bar{\mathbf{x}}^N - \mathbf{x}^*)$$

Therefore we may write

$$\mathcal{L}(\bar{\mathbf{x}}^N, \mathbf{y}) - \mathcal{L}(\mathbf{x}, \bar{\mathbf{y}}^N) = A + B + C, \quad (20)$$

where

$$A = f(\bar{\mathbf{x}}^N) - f(\mathbf{x}^*) - \nabla f(\mathbf{x}^*)^\top (\bar{\mathbf{x}}^N - \mathbf{x}^*) \quad (21a)$$

$$B = g(\bar{\mathbf{x}}^N) - g(\mathbf{x}^*) - T_g(\mathbf{x}^*)^\top (\bar{\mathbf{x}}^N - \mathbf{x}^*) \quad (21b)$$

$$C = \frac{\rho}{2} (\bar{\mathbf{x}}^N)^\top \mathbf{L} \bar{\mathbf{x}}^N \quad (21c)$$

By the convexity of f and g , both $A, B \geq 0$. Since $\ker \mathbf{L} = \{1 \otimes x \mid x \in \mathbb{R}^d\}$, the value of $(\bar{\mathbf{x}}^N)^\top \mathbf{L} \bar{\mathbf{x}}^N$ only depends on the displacement of $\bar{\mathbf{x}}^N$ to its projection onto the consensus subspace. Since \mathbf{L} has a trivial kernel on this subspace, we may make the lower bound,

$$C = \frac{\rho}{2} (\bar{\mathbf{x}}^N)^\top \mathbf{L} \bar{\mathbf{x}}^N \geq \frac{\rho\lambda_2(\mathbf{L})}{2} \|\bar{\mathbf{x}}^N - \bar{\mathbf{x}}_C^N\|_2^2.$$

Hence we have proven

$$\mathcal{L}(\bar{\mathbf{x}}^N, \mathbf{y}) - \mathcal{L}(\mathbf{x}, \bar{\mathbf{y}}^N) \geq \frac{\rho\lambda_2(\mathbf{L})}{2} \|\bar{\mathbf{x}}^N - \bar{\mathbf{x}}_C^N\|_2^2.$$

The result now follows from combining this bound with (11). ■

Although the Laplacian matrix \mathbf{L} preserves the sparsity pattern of the communication graph, the matrix $\sqrt{\mathbf{L}}$ does not; hence, the iterations in (9) cannot be implemented in a

distributed fashion. To move to a fully distributed implementation, we make the change of variables $\boldsymbol{\nu} = \sqrt{\mathbf{L}}\mathbf{y} + \rho\mathbf{L}\mathbf{x}$. The iterations in (9) can now be written in terms of the new variable $\boldsymbol{\nu}$, as follows:

$$\hat{\mathbf{x}}^{k+1} = \mathbf{x}^k - \alpha(\nabla f(\mathbf{x}^k) + \boldsymbol{\nu}^k), \quad (22a)$$

$$\mathbf{x}^{k+1} = \underset{\mathbf{x}}{\operatorname{argmin}} \left\{ g(\mathbf{x}) + \frac{1}{2\alpha} \|\mathbf{x} - \hat{\mathbf{x}}^{k+1}\|_2^2 \right\}. \quad (22b)$$

$$\boldsymbol{\nu}^{k+1} = \boldsymbol{\nu}^k + \mathbf{L}((\rho + 2\beta)\mathbf{x}^{k+1} - (\rho + \beta)\mathbf{x}^k). \quad (22c)$$

Here, we see why $\sqrt{\mathbf{L}}$ was chosen to create the consensus constraint instead of \mathbf{L} ; we have been able to decouple the primal update from the graph structure and only the dual update must use communication between agents. The update (22c) requires agents to know the states of their neighbors and the updates (22a) and (22b) can be performed by every agent with no communication at all. Therefore, we may distribute the updates across individual agents as outlined in the following algorithm.

Algorithm 1 Distributed Primal-Dual Algorithm

Given: $f = \sum_{i=1}^n f_i$, where f has L_f -Lipschitz gradient, $g = \sum_{i=1}^n g_i$. An undirected and connected communication graph with Laplacian matrix $\mathbf{L} = [L_{ij}]$, step sizes $0 < \alpha, \beta$, augmentation parameter $\rho > 0$.

- 1: Initialize at x_i^0 such that $x_i^0 = x_j^0$ for all $i, j = 1, \dots, n$, initialize ν_i^0 such that $\sum_i \nu_i^0 = 0$
 - 2: **for** $k = 0, 1, 2, \dots$ **all agents do**
 - 3: $\hat{x}_i^{k+1} = x_i^k - \alpha(\nabla f_i(x_i^k) + \nu_i^k)$.
 - 4: $x_i^{k+1} = \underset{x}{\operatorname{argmin}} \{g_i(x) + \frac{1}{2\alpha} \|x - \hat{x}_i^{k+1}\|_2^2\}$
 - 5: $\nu_i^{k+1} = \nu_i^k + \sum_{\ell \in \mathcal{N}_i} L_{i\ell} ((\rho + 2\beta)x_\ell^{k+1} - (\rho + \beta)x_\ell^k)$.
 - 6: **end for**
-

Note that since Theorem 1 assumes we initialize \mathbf{x}^0 to have consensus, and we assume $\mathbf{y}^0 \in (\ker \sqrt{\mathbf{L}})^\perp$, we have $\boldsymbol{\nu}^0 = \sqrt{\mathbf{L}}\mathbf{y}^0 \in (\ker \sqrt{\mathbf{L}})^\perp$, or equivalently, $\sum_i \nu_i^0 = 0$.

Theorem 1 and Corollary show the dependence of the convergence rate toward optimality on the largest eigenvalue λ_n and the dependence on the convergence rate toward consensus on the condition number λ_n/λ_2 (see (10) and (18)). It is here that we draw inspiration from [19] and in the next section, apply Chebyshev polynomials to \mathbf{L} in order to increase the largest eigenvalue, reduce the condition number, and achieve a faster ergodic convergence rate.

B. The Preconditioned Primal-Dual Algorithm

The Chebyshev polynomials of the first kind are defined by the recurrence relation,

$$T_0(x) = 1, \quad T_1(x) = x, \quad T_k(x) = 2xT_{k-1}(x) - T_{k-2}(x),$$

and have the property [16],

$$\underset{p \in \mathbb{P}_k, p(\gamma)=1}{\operatorname{argmin}} \max_{t \in [\alpha, \beta]} |p(t)| = \frac{1}{T_k \left(1 + \frac{\gamma - \beta}{\beta - \alpha}\right)} T_k \left(1 + \frac{t - \beta}{\beta - \alpha}\right), \quad (23)$$

where \mathbb{P}_k is the set of monic polynomials of degree k with real coefficients. In other words, the Chebyshev polynomials can be used to construct polynomials with minimal ℓ^∞ norms over intervals. We will use a polynomial which minimizes

the maximum distance of the nonzero eigenvalues of \mathbf{L} to 1, while preserving the eigenvalue at 0,

$$\operatorname{argmin}_{p \in \mathbb{P}_k, p(0)=0} \max_{t \in [\lambda_2, \lambda_n]} |1 - p(t)|. \quad (24)$$

Such a polynomial when applied to \mathbf{L} will cluster all of the eigenvalues closer to 1, thus reducing the condition number. This approach is used in [19], where it is shown that for a scaled version of the Laplacian, $c_2 \mathbf{L}$, where $c_2 = \frac{2}{(1 + \frac{\lambda_2}{\lambda_n}) \lambda_n}$, the relation (23) allows one to find that

$$\operatorname{argmin}_{p \in \mathbb{P}_k, p(0)=0} \max_{t \in [1 - c_1^{-1}, 1 + c_1^{-1}]} |1 - p(t)| = 1 - \frac{T_K(c_1(1 - x))}{T_K(c_1)}, \quad (25)$$

where $c_1 = (1 + \frac{\lambda_2}{\lambda_n}) / (1 - \frac{\lambda_2}{\lambda_n})$ and the nonzero eigenvalues of $c_2 \mathbf{L}$ lie in $[1 - c_1^{-1}, 1 + c_1^{-1}]$. We similarly propose to precondition the Laplacian matrix with the polynomial,

$$P_k(x) = 1 - \frac{1}{T_k(c_1)} T_k(c_1(1 - x)). \quad (26)$$

We verify that the matrix $P_k(c_2 \mathbf{L})$ will still create consensus among agents. The proof is deferred to the appendix.

Proposition 1 *The preconditioned gossip matrix $P_k(c_2 \mathbf{L})$ satisfies $\ker P_k(c_2 \mathbf{L}) = \operatorname{span}\{1_n \otimes x \mid x \in \mathbb{R}^d\}$.*

Thus, we may encode the constraint of the problem (3) as $\sqrt{P_k(c_2 \mathbf{L})} \mathbf{x} = 0$ and form the associated saddle point problem. Using the recurrence relation for the Chebyshev polynomials we arrive at the following preconditioned primal dual algorithm.

Algorithm 2 Chebyshev-Accelerated Distributed Primal-Dual Algorithm

Given: $f = \sum_{i=1}^n f_i$, where f has L_f -Lipschitz gradient, $g = \sum_{i=1}^n g_i$. An undirected connected communication graph with gossip matrix $\mathbf{L} = [L_{ij}]$, Chebyshev polynomial degree K , step size $\alpha > 0$, augmentation parameter $\rho > 0$, set $\beta = \frac{1}{P_K(c_1 \lambda_n)} \left(\frac{1}{\alpha} - L_f \right) - \rho$.

- 1: Initialize at x_i^0 such that $x_i^0 = x_j^0$ for all $i, j = 1, \dots, n$, initialize ν_i^0 such that $\sum_i \nu_i^0 = 0$
 - 2: **for** $k = 0, 1, 2, \dots$ all agents **do**
 - 3: $\hat{x}_i^{k+1} = x_i^k - \alpha (\nabla f_i(x_i^k) + \nu_i^k)$.
 - 4: $x_i^{k+1} = \operatorname{argmin}_x \{g_i(x) + \frac{1}{2\alpha} \|x - \hat{x}_i^{k+1}\|_2^2\}$
 - 5: /* Initialization of auxiliary variables for accelerated gossip */
 - 6: $a_i^0 = 1, a_i^1 = c_1$.
 - 7: $\xi_i^0 = (\rho + 2\beta)x_i^{k+1} - (\rho + \beta)x_i^k$.
 - 8: $\xi_i^1 = c_1 \xi_i^0 - c_1 c_2 \sum_{\ell \in \mathcal{N}_i} L_{i\ell} \xi_\ell^0$.
 - 9: /* Accelerated gossip rounds */
 - 10: **for** $j = 1, 2, \dots, K - 1$ all agents **do**
 - 11: $a_i^{j+1} = 2c_1 a_i^j - a_i^{j-1}$
 - 12: $\xi_i^{j+1} = 2c_1 \xi_i^j - 2c_1 c_2 \sum_{\ell \in \mathcal{N}_i} L_{i\ell} \xi_\ell^j - \xi_i^{j-1}$
 - 13: **end for**
 - 14: $\nu_i^{k+1} = \nu_i^k + \left(\xi_i^0 - \frac{\xi_i^K}{a_i^K} \right)$.
 - 15: **end for**
-

Note that, for a given primal step size α , we have set the dual step size, $\beta = \frac{1}{P_K(c_1 \lambda_n)} \left(\frac{1}{\alpha} - L_f \right) - \rho$, to be as

large as possible (as given by Theorem 1) after the effect of the Chebyshev preconditioning on the Laplacian matrix. The updates for \hat{x}_i^{k+1} and x_i^{k+1} are identical to those in Algorithm 1. After these updates we must use the auxiliary variables $\{\xi_i^j\}_{j=1}^K$ and $\{a_i^j\}_{j=1}^K$ in order to compute the recurrence for the Chebyshev polynomial. The net effect of the inner loop is to use a gossip matrix with better spectral properties, at the cost of taking K extra rounds of communication per gradient update. Note that this effect is not achieved by simply performing the dual update in Algorithm 1 (the original algorithm without preconditioning) K times per primal update. Depending on the specific application, one would have to decide how many extra rounds of communication can be tolerated based on the relative costs of gradient updates and communications between agents. When the cost of agent communications is relatively cheap compared to the cost of gradient updates, the degree of the preconditioning Chebyshev polynomial can be larger without significantly increasing the time it takes for the algorithm to converge.

IV. NUMERICAL SIMULATIONS

In this section, we consider a distributed sparse signal recovery problem with an unknown signal $x_0 \in \mathbb{R}^d$ ($d = 1024$), which consists of $T = 10$ spikes with amplitude ± 1 . Consider a network of $n = 100$ agents, in which agent $i \in \{1, \dots, n\}$ measures $b_i = A_i x_0 + v_i \in \mathbb{R}^{n_i}$ ($n_i = 10$ for all i), where v_i is drawn according to the Gaussian distribution $\mathcal{N}(0, 0.01 I_{n_i})$ on \mathbb{R}^{n_i} , and $A_i \in \mathbb{R}^{n_i \times d}$ is the measurement matrix ($n_i < d$) of agent i . The aggregated measurement matrix $A = [A_1^T \dots A_n^T]^T \in \mathbb{R}^{1000 \times 1024}$ is obtained by orthogonalizing the rows of a 1000×1024 matrix whose entries are generated independently and identically according to the standard normal distribution [24]. The goal of the agents is to recover the sparsity pattern of x^* without exchanging their data (A_i, b_i) . The corresponding centralized optimization problem can be written as

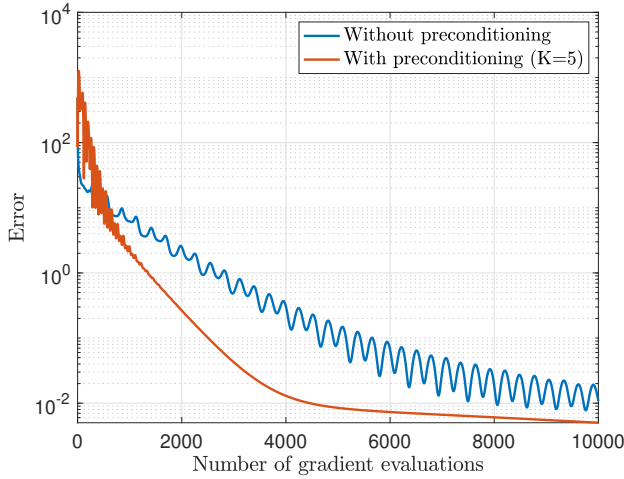
$$\operatorname{minimize}_{x \in \mathbb{R}^d} \sum_{i=1}^n \left\{ \underbrace{\frac{1}{2} \|A_i x_i - b_i\|_2^2}_{f_i(x)} + \underbrace{\alpha_i \|x\|_1}_{g_i(x)} \right\} \quad (27)$$

$$\text{subject to} \quad x_1 = \dots = x_n,$$

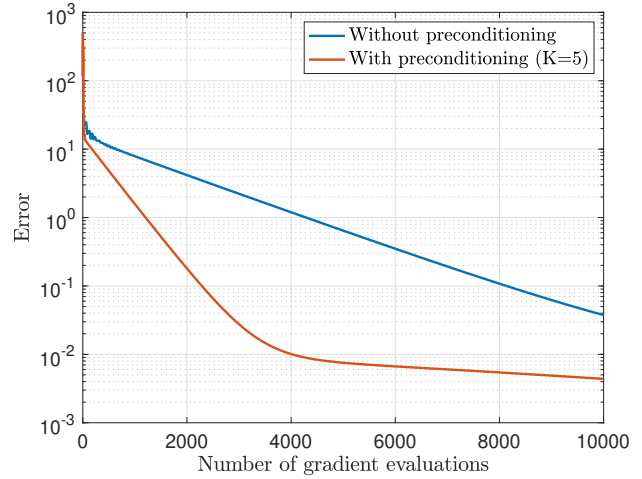
where $\alpha_i = \alpha/n$, $\alpha = 0.01$ for all i . The first term is the sum of the squared measurement errors and the second term is an ℓ_1 regularization to enhance sparsity in the decision variables. Given these numerical values, we run Algorithm 1 and 2 with an appropriate selection of the step sizes prescribed by Theorem 1. To measure the progress of the algorithm, we compute the quantities

$$\varepsilon_1^k = \frac{1}{n} \sum_{i=1}^n \{(f_i(x_i^k) - f_i(x^*)) + (g_i(x_i^k) - g_i(x^*))\}, \quad (28a)$$

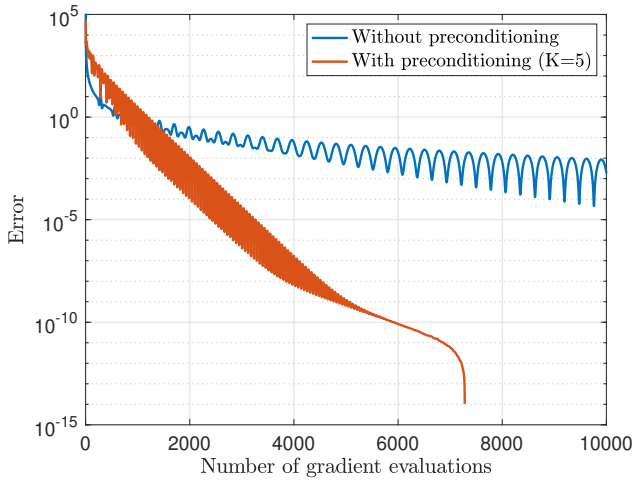
$$\varepsilon_2^k = \frac{1}{2} \sum_{i=1}^n \sum_{j \in \mathcal{N}_i} \|x_i^k - x_j^k\|_2^2. \quad (28b)$$



(a) Average suboptimality for sparse signal recovery on a chain graph with $n = 100$ nodes.

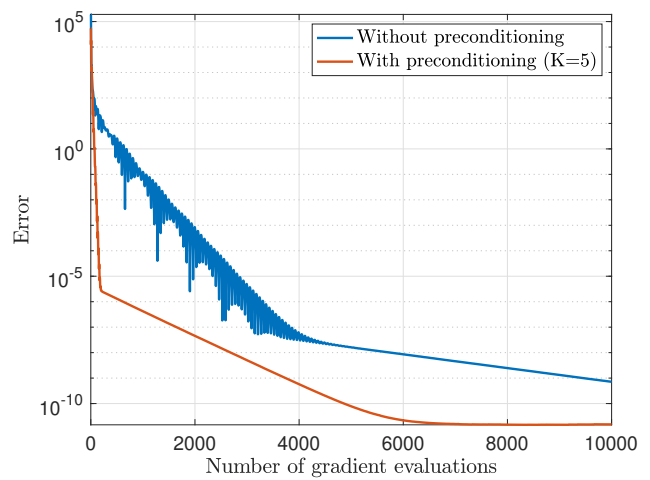


(a) Average suboptimality for sparse signal recovery on an Erdős-Rényi graph with average degree 3 and $n = 100$ nodes.



(b) Total disagreement between agents on a chain graph with $n = 100$ nodes.

Fig. 1



(b) Total disagreement between agents on an Erdős-Rényi graph with average degree 3 and $n = 100$ nodes.

Fig. 2

where x^* is an optimal solution to the centralized problem. The first term, ε_1^k , is the average suboptimality, while the second term, ε_2^k , quantifies the disagreement between the agents states.

In Figure 1a, we plot ε_1^k and ε_2^k for Algorithm 2 with $K = 1$ (no preconditioning) and $K = 5$ (preconditioning using five communication rounds per gradient update) applied to a chain network. Such a network has poor mixing properties, and we clearly observe the benefit of Chebyshev preconditioning on the empirical convergence rate. For instance, without preconditioning, the number of gradient evaluations per node to achieve an accuracy level of 0.1 is roughly 5000, while this number is roughly 2500 after preconditioning. In Figure 1b we see how far from consensus the network is as the iterations increase. The high degree of oscillations correspond to the difficulty in maintaining consensus during the iterations for a network with a high condition number,

such as a chain graph.

We also repeat the experiment for an Erdős-Rényi network with an average node degree of 3 and we plot the error in Figure 2a. Here we observe a significant increase the performance of our algorithm compared to the standard primal dual method. From Figure 2b we observe that preconditioning drives the network more quickly to consensus.

We acknowledge that the added complexity of the additional rounds of communication in the preconditioned case is hidden in these plots. However, for the case where gradients are much more expensive to compute compared to the communications these plots give a reasonable indication of relative performance of the algorithms. We emphasize that our algorithm is most useful in such scenarios.

V. CONCLUSION

We have considered a distributed optimization problem over a network of agents. To this end, our main contribution

is a distributed Chebyshev-accelerated primal-dual algorithm where we use Chebyshev matrix polynomials to generate gossip matrices whose spectral properties result in faster convergence speeds, while allowing for a fully distributed implementation. As a result, the proposed algorithm requires fewer gradient updates at the cost of additional rounds of communications between agents. We have illustrated the performance of the proposed algorithm in a distributed signal recovery problem. We have also shown how the use of Chebyshev matrix polynomials can be used to improve the convergence speed over communication networks by trading local computation by communication rounds. For future work, we will explore the possibility of using Chebyshev polynomials to gossip the gradients between updates to obtain a faster convergence. Furthermore, we will study the option of adding (properly tuned) momentum terms to the updates in order to improve the convergence speed of our algorithm.

VI. APPENDIX

A. Proof of Proposition 1

From the definition of \mathbf{L} it suffices to prove that $\ker P_k(c_2 \mathbf{L}) = \text{span}\{1_n\}$. Since 0 is an eigenvalue of \mathbf{L} we have that 0 is an eigenvalue of $P_k(c_2 \mathbf{L})$ as well, as $P_k(0) = 0$. We claim that this eigenvalue has multiplicity 1. If not, then there is some nonzero eigenvalue λ of $c_2 \mathbf{L}$, such that $P_k(\lambda) = 0$ or equivalently, that $T_k(c_1(1-\lambda)) = T_k(c_1)$.

Recall that $c_1 > 1$ and note that the nonzero eigenvalues of $c_2 \mathbf{L}$ lie in the interval $[1-c_1^{-1}, 1+c_1^{-1}]$, so λ is contained in this interval. This implies that $c_1(1-\lambda) \in [-1, 1]$. Since $|T_k(x)| > 1$ when $|x| > 1$ and $|T_k(x)| \leq 1$ otherwise, we cannot have $T_k(c_1(1-\lambda)) = T_k(c_1)$ unless $\lambda = 0$. Thus, $\lambda = 0$ has multiplicity 1 as an eigenvalue of $P_k(c_2 \mathbf{L})$.

It remains to show that for $x \in \text{span}\{1_n\}$ we have $P_k(c_2 \mathbf{L})x = 0$. From the definition of P_k , it suffices to prove that

$$T_k(c_1(\mathbf{I}_n - c_2 \mathbf{L})) = T_k(c_1)x.$$

We proceed inductively using the recurrence relation for T_k . Assuming that $T_{k-1}(c_1(\mathbf{I}_n - c_2 \mathbf{L})) = T_{k-1}(c_1)x$ and using the fact that $\mathbf{L}x = 0$, we see

$$\begin{aligned} T_k(c_1(\mathbf{I}_n - c_2 \mathbf{L}))x &= 2c_1(\mathbf{I}_n - c_2 \mathbf{L})T_{k-1}(\mathbf{I}_n - c_2 \mathbf{L})x \\ &\quad - T_{k-2}(\mathbf{I}_n - c_2 \mathbf{L})x \\ &= 2c_1(\mathbf{I}_n - c_2 \mathbf{L})T_{k-1}(c_1)x - T_{k-2}(c_1)x \\ &= (2c_1T_{k-1}(c_1) - T_{k-2}(c_1))x \\ &= T_k(c_1)x. \end{aligned}$$

Hence, $\text{span}\{1_n\} \subseteq \ker P_k(c_2 \mathbf{L})$, and since we have shown $\ker P_k(c_2 \mathbf{L})$ is one dimensional, we conclude $\ker P_k(c_2 \mathbf{L}) = \text{span}\{1_n\}$, as desired. ■

REFERENCES

[1] F. Yan, S. Sundaram, S. Vishwanathan, and Y. Qi, "Distributed autonomous online learning: Regrets and intrinsic privacy-preserving properties," *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 11, pp. 2483–2493, 2013.

[2] L. Bai, M. Ye, C. Sun, and G. Hu, "Distributed economic dispatch control via saddle point dynamics and consensus algorithms," *IEEE Transactions on Control Systems Technology*, no. 99, pp. 1–8, 2017.

[3] D. V. Dimarogonas, E. Frazzoli, and K. H. Johansson, "Distributed event-triggered control for multi-agent systems," *IEEE Transactions on Automatic Control*, vol. 57, no. 5, pp. 1291–1297, 2012.

[4] W. Shi, Q. Ling, G. Wu, and W. Yin, "Extra: An exact first-order algorithm for decentralized consensus optimization," *SIAM Journal on Optimization*, vol. 25, no. 2, pp. 944–966, 2015.

[5] K. Yuan, Q. Ling, and W. Yin, "On the convergence of decentralized gradient descent," *SIAM Journal on Optimization*, vol. 26, no. 3, pp. 1835–1854, 2016.

[6] A. Nedić and A. Ozdaglar, "On the rate of convergence of distributed subgradient methods for multi-agent optimization," *Proceedings of the IEEE Conference on Decision and Control*, vol. 54, no. 1, pp. 4711–4716, 2007.

[7] D. Feijer and F. Paganini, "Stability of primal dual gradient dynamics and applications to network optimization," *Automatica*, vol. 46, no. 12, pp. 1974–1981, 2010.

[8] S. S. Kia, J. Cortés, and S. Martínez, "Distributed convex optimization via continuous-time coordination algorithms with discrete-time communication," *Automatica*, vol. 55, pp. 254–264, 2015.

[9] J. Wang and N. Elia, "Control approach to distributed optimization," *2010 48th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, vol. 1, no. 1, pp. 557–561, 2010.

[10] N. S. Aybat and E. Y. Hamedani, "A primal-dual method for conic constrained distributed optimization problems," *Advances in Neural Information Processing Systems*, pp. 5049–5057, 2016.

[11] T. Lin, S. Ma, and S. Zhang, "On the global linear convergence of the admm with multiblock variables," *SIAM Journal on Optimization*, vol. 25, no. 3, pp. 1478–1497, 2015.

[12] M. Zargham, A. Ribeiro, A. Ozdaglar, and A. Jadbabaie, "Accelerated dual descent for network flow optimization," *IEEE Transactions on Automatic Control*, vol. 59, no. 4, pp. 905–920, 2014.

[13] R. Tutunov, H. B. Ammar, and A. Jadbabaie, "A distributed newton method for large scale consensus optimization," *arXiv preprint arXiv:1606.06593*, 2016.

[14] M. Fazlyab, S. Paternain, A. Ribeiro, and V. M. Preciado, "Distributed smooth and strongly convex optimization with inexact dual methods," in *2018 Annual American Control Conference (ACC)*, pp. 3768–3773, June 2018.

[15] P. Giselsson and S. Boyd, "Preconditioning in fast dual gradient methods," in *Decision and Control (CDC), 2014 IEEE 53rd Annual Conference on*, pp. 5040–5045, IEEE, 2014.

[16] W. Auzinger and J. Melenk, "Iterative solution of large linear systems," *Lecture notes, TU Wien*, 2011.

[17] E. Montijano, J. I. Montijano, and C. Sagues, "Fast distributed consensus with Chebyshev polynomials," *Proceedings of the 2011 American Control Conference*, pp. 5450–5455, 2011.

[18] R. L. G. Cavalcante, A. Rogers, and N. Jennings, "Consensus Acceleration in Multiagent Systems with the Chebyshev Semi-Iterative Method," *The Tenth International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, no. Aamas, pp. 165–172, 2011.

[19] K. Scaman, F. Bach, S. Bubeck, Y. T. Lee, and L. Massoulié, "Optimal algorithms for smooth and strongly convex distributed optimization in networks," in *International Conference on Machine Learning*, pp. 3027–3036, 2017.

[20] A. Chambolle and T. Pock, "On the ergodic convergence rates of a first-order primal-dual algorithm," *Mathematical Programming*, vol. 159, no. 1-2, pp. 253–287, 2016.

[21] N. Parikh, S. Boyd, et al., "Proximal algorithms," *Foundations and Trends® in Optimization*, vol. 1, no. 3, pp. 127–239, 2014.

[22] J. Duchi, S. Shalev-Shwartz, Y. Singer, and T. Chandra, "Efficient projections onto the l_1 -ball for learning in high dimensions," in *Proceedings of the 25th international conference on Machine learning*, pp. 272–279, ACM, 2008.

[23] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[24] S.-J. Kim, K. Koh, M. Lustig, S. Boyd, and D. Gorinevsky, "An interior-point method for large-scale l_1 -regularized least squares," *IEEE journal of selected topics in signal processing*, vol. 1, no. 4, pp. 606–617, 2007.