Scalable Distributed Optimization with Separable Variables in Multi-Agent Networks

Olaoluwa Shorinwa[†], Trevor Halsted[†], Mac Schwager[‡]

***Abstract*— Robotics, signal processing, and other disciplines involve distributed data collection and storage for state estimation, control, and predictive modeling using optimization. We consider large-scale optimization problems in which multiple agents with limited resources communicate over a network to obtain the optimal variables of the centralized problem. In this work, we present the Separable Optimization Variable ADMM (SOVA) method where each agent optimizes only over a subset of the optimization variables relevant to its data or role, avoiding unnecessary optimization over all the problem variables. We demonstrate superior convergence rates of the SOVA method compared to previous distributed ADMM methods. Further, we show applications of the SOVA method to robotics and data modeling.**

I. Introduction

Advances in computing and sensing have spurred the deployment of decentralized data collection and storage systems. In robotics, signal processing, and process control, increasingly autonomous systems rely on distributed data collection for improved estimation, localization, planning, and control. In other disciplines, the resulting data is employed in building predictive models for autonomous financial systems in algorithmic trading and improved personalization services in media. The collected data can be aggregated centrally to extract the model variables through optimization. However, agents are often sparsely located, and the collated data is often large in size, creating communication, storage and computing challenges. To address these issues, distributed optimization methods have been applied to these problems to eliminate the need for data transfer and storage at a central location.

The dual decomposition method has been applied to distributed minimization of differentiable convex functions using dual ascent and extended to problems with non-differentiable convex objectives using subgradients [1], [2], [3], [4] (see [5] for a discussion on the use of dual decomposition and subgradient methods in distributed optimization). More recently, these approaches have been combined with consensus methods for fully distributing problems with separable objective functions. These methods employ consensus with a subgradient projection step for distributed optimization [6], [7].

[†] O. Shorinwa and T. Halsted are with the Department of Mechanical Engineering, Stanford University. Email: {shorinwa, halsted}@stanford.edu

[‡] M. Schwager is with the Department of Aeronautics and Astronautics, Stanford University. Email: schwager@stanford.edu

This work was funded in part by DARPA YFA grant D18AP00064, NSF NRI grant 1830402, and ONR grant N00014-18-1-2830. The second author was supported on an NDSEG fellowship. We are grateful for this support.

Convergence in dual decomposition methods is guaranteed for strictly convex or finite objective functions, prompting the development of augmented Lagrangian methods such as the method of multipliers for improved convergence in problems with non-finite objectives. The alternating direction method of multipliers (ADMM) blends the dual decomposition method with the method of multipliers, providing better convergence properties than the dual decomposition method [8], [9]. The method belongs to a sub-class of other distributed optimization methods derived from the proximal point algorithm and often requires reformulating the original problem for decomposability of the problem objective [10], [11], [12]. ADMM combines a dual variable update through dual ascent with a minimization step for the primal variables and requires the existence of a central node for maintaining and communicating the dual variable with other agents (see [13], [14] for a recent survey of ADMM and its update procedures). Other augmented Lagrangian methods make similar assumptions on the existence of a central node.

In many applications, no single agent can communicate with all other agents due to technical constraints such as limited communication ranges. Addressing this challenge, some distributed approaches formulate the ADMM method on a communication graph with no requirement for a central node [16]. This ADMM method uses consensus to ensure each agent obtains the same solution and is referred to as consensus ADMM in literature. Other approaches solve distributed optimization problems with a centralized linear sharing constraint using multiple consensus steps at every update iteration, with increased computation and communication costs [15]. In contrast, we do not employ any consensus steps within our optimization method.

In consensus ADMM methods, each agent optimizes over all the problem variables at each update step. In many applications, only a small subset of these variables depend on data stored by each agent while in other scenarios, each agent requires a subset of the variables for its assigned roles. Such applications arise in signal processing where receivers operate within a limited frequency band or collect different signals and in sensor networks and distributed robotics where sensors or robotic agents are situated physically in different locations and observe different areas of their environment. Optimizing over all problem variables creates communication, storage, and computing overhead which can be limiting in resource-constrained systems . Previous work has examined using approximate primal updates within the ADMM method to ease its computational complexity such as [17], [18], [19]. The inexact update steps improve computational

cost at the expense of slower convergence rates and increased communication costs.

In this paper, we present the Separable Optimization variable ADMM (SOVA) method, a distributed optimization algorithm for these applications which does not require a central node for implementation. With the SOVA method, each agent optimizes over a subset of the problem variables, eliminating the overhead from all agents performing unnecessary optimization over all the variables. SOVA allows for arbitrary mappings between elements of each agent's optimization variable to the centralized optimization variable. We prove its convergence to the centralized optimal solution under typical convexity and regularity conditions on the objective function and constraints. In addition, we show in numerical simulations that SOVA converges more than two orders of magnitude faster than consensus ADMM in three prevalent problems in different domains.

The paper is organized as follows. In Section II, we formulate the general optimization problem and introduce notation for distributed optimization. In Section III, we describe the consensus ADMM method and derive the SOVA method in Section IV. In Section V, we demonstrate the superior convergence, communication, and computational properties of SOVA in sensor networks and data modeling problems in statistics, finance, and medicine. We provide concluding remarks in Section VI.

II. Problem Formulation

In this section, we formulate the general distributed optimization problem with separable objectives among N agents. Consider the optimization problem

$$\underset{z}{\text{minimize}} \ f(z) \tag{1}$$

where $f(z) : \mathbf{R}^n \to \mathbf{R}$ denotes the problem objective, and $z \in \mathbf{R}^n$ is the optimization variable. The optimal variable $z^\star$ in (1), depends on each agent's local data. We consider unconstrained optimization; however, our method can be readily extended to optimization problems with convex constraints. In distributed optimization, we solve (1) through local operations performed by each agent instead of centrally. Given a group of N agents and a separable objective function, Problem (1) can be expressed in optimization variables $x_i \in \mathbf{R}^n \ i = 1, \cdots, N$ which are local copies of z. Agent i updates its optimization variable x_i to recover the optimal solution of (1) using its local data. The corresponding optimization statement is given by

$$\begin{array}{ll} \underset{x_1,\cdots,x_N}{\text{minimize}} & \displaystyle\sum_{i=1}^{N} f_i(x_i) \\ \text{subject to} & x_i = z \quad i = 1, \cdots, N \end{array} \tag{2}$$

where $f_i(x_i)$ encodes the optimization objective for agent i. x_i denotes the optimization variable of agent i, and z consists of all the unknown parameters of the centralized problem. The consensus constraints in (2) ensure each agent obtains the same solution.

Communication Graph

We represent the agents as nodes in an undirected graph $\mathcal{G}$ with a set of vertices $\mathcal{V} = \{1, \cdots, N\}$ and edges $\mathcal{E} = \{\cdots, (i,j), \cdots\}$ given there is a communication link between agents i and j. The neighbor set of agent i represents the set of agents sharing a communication link with agent i which we denote as $\mathcal{N}_i = \{j \mid (i,j) \in \mathcal{E}\}$.

Notation: $|\mathcal{N}_i|$ denotes the cardinality of the neighbor set of agent i. $\mathbf{I}_n \in \mathbf{R}^{n \times n}$ denotes the identity matrix. We denote the Mahalonobis norm (the weighted l_2 norm with matrix M) as $\|\cdot\|_M$.

III. Consensus ADMM

In consensus ADMM methods, each agent maintains a local copy of the optimization variable z (shown in Figure 1). The method introduces consensus constraints between pairs of nodes sharing a communication link, ensuring each agent converges to the same values with the assumption that the communication graph $\mathcal{G}$ is connected.

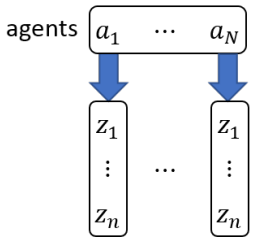

Fig. 1. Each agent maintains a local copy of the entire optimization variable in consensus ADMM.

The consensus ADMM method reformulates Problem (2) as

$$\begin{array}{ll} \underset{x_1,\cdots,x_N}{\text{minimize}} & \displaystyle\sum_{i=1}^{N} f_i(x_i) \\ \text{subject to} & x_i = x_j \quad \forall j \in \mathcal{N}_i \quad i = 1, \cdots, N \end{array} \tag{3}$$

and minimizes the augmented Lagrangian of Problem (3) with respect to x_i while updating the dual variables through gradient ascent [16]. The update steps for x_i and the dual variable $p_i \in \mathbf{R}^n$ at iteration $k+1$ are

$$x_i^{k+1} = \underset{x_i}{\text{argminimize}} \Bigg(f_i(x_i) + p_i^{kT} x_i + \rho \sum_{j \in \mathcal{N}_i} \left\| x_i - \frac{x_i^k + x_j^k}{2} \right\|_2^2 \Bigg) \tag{4}$$

$$p_i^{k+1} = p_i^k + \rho \sum_{j \in \mathcal{N}_i} \left(x_i^k - x_j^k \right) \tag{5}$$

where ρ is a penalty term on the agreement between the optimization variables of an agent and its neighbors. In many problems with non-differentiable objective functions, solving the update step in (4) is computationally expensive. Inexact consensus ADMM methods replace $f_i(x_i)$ in (4) with its first-order approximations to yield more tractable update steps [17], [18].

IV. Separable Optimization Variable ADMM

In consensus ADMM methods, all agents store and solve for all parameters in z which is superfluous in many applications where each agent's data relates to only a subset of the parameters. Moreover, each agent often requires only a subset of these parameters for its role. This situation arises in estimation problems, signal processing, and data fitting where data collection sources and modes differ. For example, robots are often equipped with different sensing modalities including active sensors such as lidar and radar and passive vision-based sensors while receivers in signal processing operate within different frequency bands. In computing, medicine, and other disciplines, data collection modes differ across devices and locations.

Despite this structure, each agent optimizes over all parameters in z irrespective of the relevance of its data to each individual parameter. This strategy is inefficient considering the limited storage, computing, and communication resources available to each agent. Hence, we develop a distributed optimization algorithm for problems with separable optimization variables in which each agent optimizes over a subset of the optimization variable depending on the relevance of these unknown parameters to its data or role. The optimization variable is distributed among the agents as depicted in Figure 2.

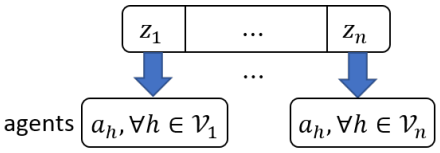

Fig. 2. The optimization variable is separated among agents in the SOVA method.

Optimization Variable Separation Subgraph

The separation of the optimization variable among the agents induces a set of subgraphs of the communication graph $\mathcal{G}$ denoted by $\mathcal{G}_s = \{\mathcal{G}_s^i, \quad i = 1, \cdots, n\}$. Subgraph $\mathcal{G}_s^i$ is described by a set of vertices $\mathcal{V}_i$ consisting of all agents optimizing over parameter z_i and a set of edges $\mathcal{E}_i$. An edge (j, k) exists in $\mathcal{E}_i$ given agents j and k optimize over z_i (j and $k \in \mathcal{V}_i$) and agents j and k share a communication link in $\mathcal{G}$. We can construct these subgraphs from the relevance of an agent to each parameter in z, depending on its role or data.

Optimization Procedure

The SOVA method is not limited to applications in which each agent's optimization variable consists of parameters in z but also applies to cases where combinations of elements within each agent's optimization variable map to the parameters in z. We make the following assumptions on the graph topology and the objective functions.

Assumption 1. *The subgraphs in $\mathcal{G}_s$ induced by distributing the parameters in z are connected i.e. a set of edges exists which connect any pair of nodes in $\mathcal{G}_s^i, i = 1, \cdots, n$.*

Assumption 2. *The objective function $f_i(x_i)$ is closed, proper, and convex $\forall i \in \mathcal{V}$.*

In many problems, agents optimizing over the same parameter in z perform similar roles and share close proximity. Consequently, these agents can communicate with each other, satisfying Assumption 1. For example, in estimation problems where the agents collect data on a given event such as temperature and concentration in process control or amplitudes in signal processing, agents optimize over the same variable when these agents are in close proximity to one another. We introduce consistency constraints on equivalent elements within each agent's optimization variable in Problem (2).

$$\begin{aligned} \underset{x_1,\cdots,x_N}{\text{minimize}} \quad & \sum_{i=1}^{N} f_i(x_i) \\ \text{subject to} \quad & A_{ij}x_i = c_{ij} \\ & B_{ij}x_j = d_{ij} \\ & c_{ij} = d_{ij} \quad j \in \mathcal{N}_i, \quad i = 1, \cdots, N \end{aligned} \tag{6}$$

where $A_{ij} \in \mathbf{R}^{m_i \times n_i}$ and $B_{ij} \in \mathbf{R}^{m_j \times n_j}$ define mappings between elements in x_i and x_j to each parameter in z, and agent i optimizes over n_i variables and shares m_i consistency constraints with agent j. Agent i shares a consistency constraint with agent j if both agents optimize over the same parameter p in z, described by the subgraph G_s^p. Mappings between elements in the consistency constraints are not limited to individual elements. The consistency constraints can be enforced on arbitrary combinations of elements within each agent's optimization variable. We have introduced the slack variables c_{ij} and d_{ij} into the consistency constraints in (6), ensuring that each agent maintains a local slack variable. Note that agent i maintains c_{ij} while agent j maintains d_{ij}. We denote the set of optimization variables $\{x_1, \cdots, x_N\}$ as $x_{1:N}$ and derive the augmented Lagrangian $\mathcal{L}_a$ of Problem (6).

$$\begin{aligned} \mathcal{L}_a(x_{1:N}, c, d, v, w) = & \sum_{i=1}^{N} f_i(x_i) \\ & + \sum_{i=1}^{N} \sum_{j \in \mathcal{N}_i} \Big(v_{ij}^T (A_{ij}x_i - c_{ij}) \\ & \qquad + w_{ij}^T (B_{ij}x_j - d_{ij}) \Big) \\ & + \frac{\rho}{2} \sum_{i=1}^{N} \sum_{j \in \mathcal{N}_i} \Big(\|A_{ij}x_i - c_{ij}\|_2^2 \\ & \qquad + \|B_{ij}x_j - d_{ij}\|_2^2 \Big) \end{aligned} \tag{7}$$

$v_{ij} \in \mathbf{R}^{m_i}$ and $w_{ij} \in \mathbf{R}^{m_j}$ are dual variables on the consistency constraints.

At each iteration, x_i, c_{ij}, and d_{ij} are updated by minimizing $\mathcal{L}_a$ with respect to these variables using the dual variables from the previous iterations. The dual variables are updated through gradient ascent on $\mathcal{L}_a$. Initializing $v_{ij}^0 = -w_{ij}^0 =$

$-v_{ji}^0 = 0$, the update equations for c_{ij} and d_{ij} are

$$c_{ij}^{k+1} = d_{ij}^{k+1} = \frac{1}{2}\big(A_{ij}x_i^{k+1} + B_{ij}x_j^{k+1}\big) \quad j \in \mathcal{N}_i \tag{8}$$

Likewise, the dual update equation reduces to

$$v_{ij}^{k+1} = v_{ij}^k + \frac{\rho}{2}\big(A_{ij}x_i^{k+1} - B_{ij}x_j^{k+1}\big) \tag{9}$$

Combining the update equations for c_{ij} and d_{ij}, the update step for x_i is

$$\begin{aligned} x_i^{k+1} = \underset{x_i}{\text{argminimize}} \Big\{ & f_i(x_i) + q_i^{kT}x_i \\ & + \rho \sum_{j\in\mathcal{N}_i} \|A_{ij}x_i - \frac{1}{2}\big(A_{ij}x_i^k + B_{ij}x_j^k\big)\|_2^2 \Big\} \end{aligned} \tag{10}$$

where $q_i = 2\sum_{j\in\mathcal{N}_i}\big(A_{ij}^T v_{ij}\big)$. We have introduced q_i to combine the updates for v_{ij} and w_{ij}. The corresponding update step for q_i^{k+1} from Equation (9) is given by

$$q_i^{k+1} = q_i^k + \rho \sum_{j\in\mathcal{N}_i} \Big(A_{ij}^T\big(A_{ij}x_i^{k+1} - B_{ij}x_j^{k+1}\big)\Big) \tag{11}$$

For the distributed optimization problem in (6), agent i maintains the variables x_i and q_i. Note that the sizes of these variables will vary with each agent. Ideally, the size of each variable relates to the storage, computing, and communication resources available to the agent. At each iteration, each agent performs two updates given by Equations (10) and (11). Subsequently, the agents communicate the updated variables to their neighbors. Agent i does not need to maintain the slack variables c_{ij}, d_{ij} and dual variables v_{ij}, w_{ij}.

Algorithm 1 describes the Separable Optimization Variable ADMM algorithm.

Theorem 1. *The parameters optimized by agents in subgraph $\mathcal{G}_s^i$ converge to the optimal parameter $z_i^\star$.*

Proof. we can express Problem (6) as

$$\begin{aligned} \underset{x,z}{\text{minimize}} \quad & F(x) \\ \text{subject to} \quad & Gx = Hz \end{aligned} \tag{12}$$

where

$$F(x) = \sum_{i=1}^N f_i(x_i)$$
$$x = [x_1, \cdots, x_N]$$

The centralized optimization variable z consists of all the unknown parameters in Problem (1), and G defines the mappings from the combinations of the elements in $x_i\ \forall i \in \mathcal{V}$ to z. From Assumption 1, each subgraph in $\mathcal{G}_s$ is connected. Hence, the consistency constraint is enforced on all the agents' optimization variables within each subgraph. Likewise, the objective $F(x)$ is convex from Assumption 2. Thus, the residual $r^k = Gx - Hz$ decays to zero as the algorithm progresses as in [13]. In addition, the objective converges to the optimal value of (6). □

Theorem 2. *If the elements in each agent's optimization variable map directly to the parameters in z and all parameters in z are distributed among the agents, Problem (6) is equivalent to Problem (1). Moreover, the union of all the agents' optimization variables $X = \bigcup_{i=1}^N x_i \setminus \bigcap_{i=1}^N x_i$ converges to $z^\star$ the optimal solution of (1).*

Proof. From Assumption 1, all subgraphs in $\mathcal{G}_s$ are connected. Thus, equivalent elements in $x_i\ \forall i \in \mathcal{V}$ can be replaced with the same variables. Consequently, the union of all the agents' optimization variables X is equivalent to z in (1). Likewise, the objective in (6) can be expressed in terms of z, and Problem (6) is equivalent to Problem (1). From Theorem 1, $x_i\ \forall i \in \mathcal{V}$ converges to the optimal solution of (6) and thus, X converges to $z^\star$. □

Algorithm 1 Separable Optimization Variable ADMM

Parameter Separation
  Construct $\mathcal{G}_s^j\ j = 1, \cdots, n$
  Initialize x_i^0 and set $q_i^0 = 0\ \forall i \in \mathcal{V}$.
  $k = 0$
do *in parallel* **for** $i = 1, \cdots, N$
  $x_i^{k+1} \leftarrow$ Equation (10)
  $q_i^{k+1} \leftarrow$ Equation (11)
  $k \leftarrow k + 1$
while convergence or stopping criterion is not satisfied

V. Simulations

We examine the convergence properties and computational complexity of the SOVA method in contrast to consensus ADMM which requires each agent to optimize over a full copy of the problem variables. First, we consider a general least-squares problem with quadratic objective and investigate the rate of convergence of elements within each agent's optimization variable to the centralized optimal value. We also present applications of the SOVA algorithm to prevalent problems in sensor networks and data modeling.

A. Least Squares: Quadratic Objective Optimization

The least squares problem is prevalent in many disciplines such as statistics, finance, medicine, and engineering for parameter estimation and regression. These problems involve objective terms assigning quadratic penalties to the problem residuals. As such, we examine distributed optimization problems with quadratic objectives. The general linear least-squares problem is given by

$$\underset{x}{\text{minimize}} \quad \|y - Ax\|_2^2 \tag{13}$$

where $y \in \mathbf{R}^m$ and $A \in \mathbf{R}^{m\times n}$ represent the problem data. We seek to find the optimal parameters $x \in \mathbf{R}^n$ that minimize the estimation error or best explain the problem data.

Consider a network of N agents with each agent having a subset of the problem data. In addition, a subset of the

parameters in x influences each agent's data. We can express Problem (13) as a distributed optimization problem given by

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & \sum_{i=1}^{N} \|y_i - A_i x_{p,i}\|_2^2 \\ \text{subject to} \quad & B_{ij} x_{p,i} = C_{ij} x_{p,j} \quad \forall j \in \mathcal{N}_i,\ i = 1, \cdots, N \end{aligned} \tag{14}$$

where $y_i \in \mathbf{R}^{m_i}$ and $A_i \in \mathbf{R}^{m_i \times n_i}$ denote the problem data accessible to agent i. $B_{ij} \in \mathbf{R}^{n_{ij} \times n_i}$ and $C_{ij} \in \mathbf{R}^{n_{ij} \times n_j}$ define mappings between the parameters estimated by agents i and j. n_{ij} denotes the number of parameters jointly estimated by agents i and j. Agent i's optimization variable $x_{p,i}$ consists of a subset of parameters in x which relate to the problem data available to agent i. We assume agent i is able to estimate $x_{p,i}$ uniquely i.e $m_i \geq n_i$ and $\text{rank}(A_i) \geq n_i$.

With consensus ADMM methods, each agent optimizes over all the parameters in x. Thus, applying consensus ADMM methods requires reformulating Problem (14) as

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & \sum_{i=1}^{N} \|y_i - A_{\text{aug},i} x_i\|_2^2 \\ \text{subject to} \quad & x_i = x_j \quad \forall j \in \mathcal{N}_i,\ i = 1, \cdots, N \end{aligned} \tag{15}$$

$A_{\text{aug},i} \in \mathbf{R}^{m_i \times n}$ is obtained by augmenting A_i with zeros in corresponding locations. $x_i \in \mathbf{R}^n$ denotes agent's i copy of x. The optimal variable $x^\star$ is obtained following the procedure in Section III. At each iteration, the update procedure for x_i^{k+1} can be expressed in closed-form as

$$\begin{aligned} x_i^{k+1} = \quad & \left(A_{\text{aug},i}^T A_{\text{aug},i} + \rho |\mathcal{N}_i| \mathbf{I}_n\right)^{-1} \\ & \left(A_{\text{aug},i}^T y_i - \frac{1}{2} p_i^k + \frac{\rho}{2} \sum_{j \in \mathcal{N}_i} \left(x_i^k + x_j^k\right)\right) \end{aligned} \tag{16}$$

with the dual update given by

$$p_i^{k+1} = p_i^k + \rho \sum_{j \in \mathcal{N}_i} \left(x_i^k - x_j^k\right) \tag{17}$$

The SOVA method enables direct optimization of Problem (14) without any reformulations. The primal and dual update procedures are obtained following the procedure outlined in Algorithm 1:

$$\begin{aligned} x_{p,i}^{k+1} = & \left(A_i^T A_i + \rho \sum_{j \in \mathcal{N}_i} B_{ij}^T B_{ij}\right)^{-1} \\ & \left(A_i^T y_i - \frac{1}{2} q_{p,i}^k + \frac{\rho}{2} \sum_{j \in \mathcal{N}_i} B_{ij}^T \left(B_{ij} x_{p,i}^k + C_{ij} x_{p,j}^k\right)\right) \end{aligned} \tag{18}$$

$$q_{p,i}^{k+1} = q_{p,i}^k + \rho \sum_{j \in \mathcal{N}_i} B_{ij}^T \left(B_{ij} x_{p,i}^k - C_{ij} x_{p,j}^k\right) \tag{19}$$

with $q_{p,i}^k \in \mathbf{R}^{n_i}$.

Computational Complexity

We examine the computational complexity of each update step in consensus ADMM and SOVA methods. Each primal variable update requires solving a system of linear equations which can be solved efficiently using matrix factorization and back-solve steps. To simplify notation, we define the following matrices and vectors

$$\begin{aligned} M_f &= A_{\text{aug},i}^T A_{\text{aug},i} + \rho |\mathcal{N}_i| \mathbf{I}_n \\ h_f &= A_{\text{aug},i}^T y_i - \frac{1}{2} p_i^k + \frac{\rho}{2} \sum_{j \in \mathcal{N}_i} \left(x_i^k + x_j^k\right) \\ M_p &= A_i^T A_i + \rho \sum_{j \in \mathcal{N}_i} B_{ij}^T B_{ij} \\ h_p &= A_i^T y_i - \frac{1}{2} q_{p,i}^k + \frac{\rho}{2} \sum_{j \in \mathcal{N}_i} B_{ij}^T \left(B_{ij} x_{p,i}^k + C_{ij} x_{p,j}^k\right) \end{aligned}$$

The primal update for x_i in the consensus ADMM method involves forming $M_f \in \mathbf{R}^{n \times n}$ which requires $O(n^2 m_i)$ flops (floating-point operations). Factoring M_f costs $O(n^3)$ flops with a subsequent cost of $O(n^2)$ flops for the back-solve step. Hence, consensus ADMM requires a total computational cost of $O(n^3 + n^2 m_i)$ flops. We have retained the most significant cost terms. The dual update for p_i requires $O(n)$ flops.

In the SOVA method, each primal update step involves forming $M_p \in \mathbf{R}^{n_i \times n_i}$ at a cost of $O(n_i^2 m_i)$ and factoring M_p with $O(n_i^3)$ flops. The back-solve step requires $O(n_i^2)$ flops. The resulting computational cost of each primal update is $O(n_i^3 + n_i^2 m_i)$. The dual update on $q_{p,i}$ incurs $O(n_i)$ flops.

The computation cost of each primal and dual update in consensus ADMM forms an upper bound on the computational complexity of the primal and dual updates in the SOVA method. In large networks of resource-constrained agents with each agent estimating smaller set of parameters in x ($n_i << n$), the SOVA method substantially reduces the computational complexity of each update step performed by each agent, enabling distributed optimization with less compute resources.

Consider applications where the optimization variable x grows with the number of agents, which arises in signal estimation, robotics, medicine, and finance. In these problems, we can express the size of x as $n = \alpha N$ where N denotes the number of agents and α depends on each agent's contribution to x. When applied to these problems, consensus ADMM has a computational complexity of $O(\alpha^3 N^3 + \alpha^2 N^2 m_i)$. Meanwhile, SOVA requires significantly less computational complexity of $O(\alpha^3 + \alpha^2 m_i)$, a factor of N^3 improvement in complexity, if each agent optimizes only over α variables related to its contribution to x. Moreover, SOVA attains a computational complexity independent of the number of agents.

Data Storage and Communication Cost

In the SOVA method, each agent maintains $q_{p,i} \in \mathbf{R}^{n_i}$ and $x_{p,i} \in \mathbf{R}^{n_i}$ consisting of a subset of parameters in x, and thus requires $16 n_i$ bytes for storage with a 64-bit floating-point representation. In consensus ADMM, each

agent maintains a copy of all elements in x, requiring $16n$ bytes for storage. Consequently, the SOVA method uses a smaller storage footprint, efficient for agents with limited storage resources.

In the consensus ADMM method, each agent shares the entire parameter set with its neighbors at each iteration, incurring a cost of n floating-point samples. The update steps in SOVA require communicating $n_i \leq n$ floating-point samples. Consequently, the SOVA method incurs significantly less communication costs in large optimization problems with greater reductions as the precision of the parameters increases. In applications where n grows with N, SOVA requires constant data storage and communication costs even as the number of agents scales.

Results

We consider the distributed optimization problem with quadratic objective in (14) with $x \in \mathbf{R}^{600}$. We randomly generated the problem data $A \in \mathbf{R}^{10309\times 600}$ and $y \in \mathbf{R}^{10309}$ distributed among 150 agents and examine convergence of the consensus ADMM and SOVA algorithms to the optimal parameter $x^\star$ obtained from the centralized problem.

Figure 3 shows the magnitude of the residual $r = \|x^\star - x\|_2$ on a fully-connected graph with $\rho = 100$, and on a chain graph (the least-connected acyclic graph with each agent having at most two neighbors) with $y \in \mathbf{R}^{5365}$ and $\rho = 1000$. SOVA requires two orders of magnitude fewer iterations for convergence compared to the consensus ADMM method. On a chain graph, the SOVA algorithm converges more than three orders of magnitude faster than the consensus ADMM method which requires substantially more iterations to reach the same magnitude of the residual. In effect, the SOVA method provides increasing computation and communication benefits in networks of agents with significantly limited resources.

Figure 4 shows the convergence rates of the consensus ADMM and SOVA algorithms on randomly connected graphs as the number of agents increases. The consensus ADMM method requires orders of magnitude more iterations to converge with increasing number of agents. In contrast, the SOVA method converges in relatively the same number of iterations even as the network size increases.

The above analysis demonstrates the superior convergence properties of the SOVA method with limited communication and computation resources. Next, we discuss applications of the SOVA method to distributed optimization problems arising in robotics and data modeling. We consider chain communication graphs and show convergence to the optimal parameters of the optimization problem.

B. Sensor Networks

We consider a network of 1000 robots equipped with radio receivers operating within different frequency bands. These scenarios arise in telecommunications and search and rescue applications where robots are deployed to recover noisy signals transmitted by electronic devices and emergency beacons. The scale of these applications often require the

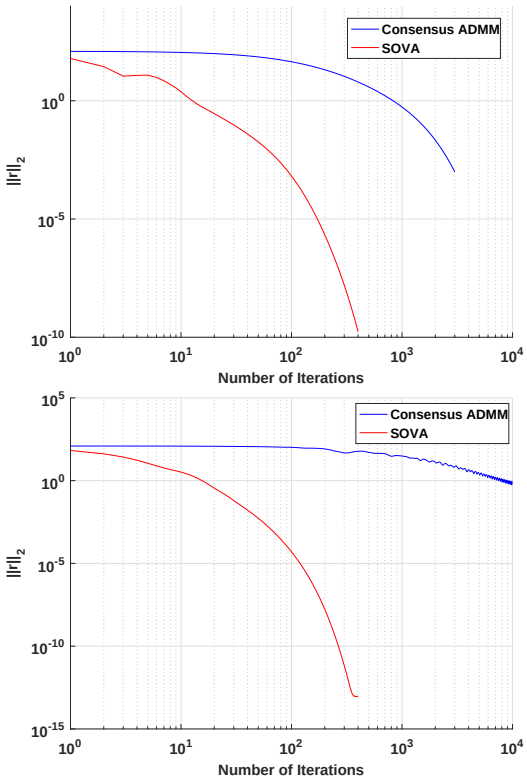

Fig. 3. Convergence rates of the consensus ADMM and Separable Optimization variable ADMM (SOVA) methods on (top) fully-connected and (bottom) chain graphs. The SOVA algorithm provides orders of magnitude increase in convergence rate especially on sparsely connected graphs such as chain graphs.

deployment of large numbers of robots with limited communication and storage resources to cover large areas which are difficult to access. The measured signals are corrupted with noise, and thus have to be de-noised. In many applications, recovering the transmitted signal from noisy measurements is posed as an optimization problem given by

$$\underset{x}{\text{minimize}} \sum_{i=1}^{N} \|y_i - A_i x_i\|_2^2 + \alpha \|G_i x_i\|_p^2 \qquad (20)$$

$y_i \in \mathbf{R}^{m_i}$ represents the measured signals at robot i. $A_i \in \mathbf{R}^{m_i \times n_i}$ denotes the problem data available to robot i. Each robot recovers $x_i \in \mathbf{R}^{n_i}$. G_i defines the regularization on x_i, measuring the smoothness or size of x_i. The l_1 norm on $G_i x_i$ arises in basis pursuit to recover sparse signals and in total variation de-noising. We consider the l_2 norm case where $G_i \in \mathbf{R}^{(n_i-2)\times n_i}$ gives a measure of the curvature of the parameter which arises in smoothing applications:

$$G_{i,jk} = \begin{cases} 1 & j = k \\ -2 & j = k+1 \\ 1 & j = k+2 \\ 0 & \text{otherwise} \end{cases}$$

j and k denote indices of G_i. We randomly generated the problem data $A \in \mathbf{R}^{6506\times 2002}$, $y \in \mathbf{R}^{6506}$ and $x \in \mathbf{R}^{2002}$. Figure 5 shows the recovered signals with $\alpha = 0$ and

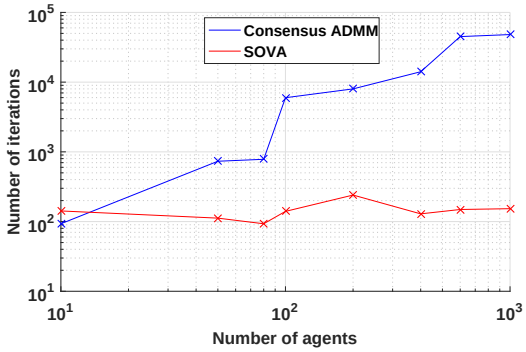

Fig. 4. Convergence rates of the consensus ADMM and Separable Optimization Variable ADMM (SOVA) methods as the number of agents grows. The convergence rate of the SOVA algorithm remains relatively the same while the consensus ADMM method requires substantially more iterations as the number of agents increases.

$\alpha = 1000$. The recovered signal appears noisy without any smoothing. At $\alpha = 1000$, the signal retains its overall shape and shows much less variation. Figure 6 shows the convergence rates of the SOVA method to $x^\star$. Convergence of the SOVA method requires three orders of magnitude less communication rounds in comparison to the consensus ADMM method, resulting in less communication and computation by each agent.

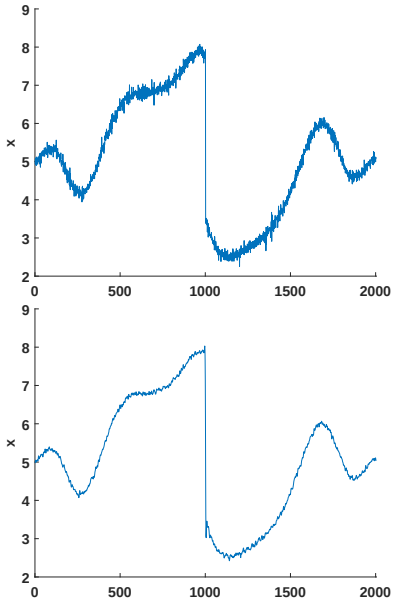

Fig. 5. Signal reconstruction with a network of 1000 robots using the Separable Optimization Variable ADMM (SOVA) algorithm. (top) The recovered signal with no smoothing $\alpha = 0$. (bottom) The smoothed recovered signal with $\alpha = 100$.

C. Data Modeling and Statistics

Medical, finance, and Internet media disciplines involve building predictive models through optimization from data collected in different modes from user devices and clinical trials. In many cases, the distributed systems require low complexity optimization methods given limited computation,

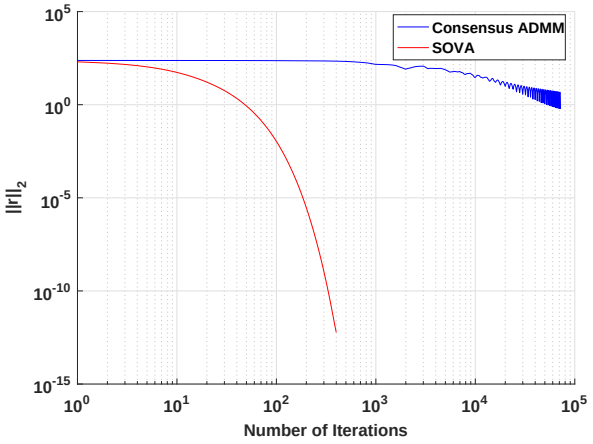

Fig. 6. Convergence rates of the consensus ADMM and Separable Optimization Variable ADMM (SOVA) methods on the distributed signal estimation problem among 1000 robots. SOVA requires orders of magnitude fewer iterations to converge.

communication, and data storage resources. The model parameters $x \in \mathbf{R}^n$ are selected to maximize the likelihood of the observed data (Maximum Likelihood Estimation) which arises often in machine learning or the posterior distribution of the data (maximum a posteriori estimation) which arises in Bayesian estimation. We can pose the problem of finding the explanatory variables as an optimization problem. We consider maximum a posteriori (MAP) estimation with the data model

$$y = f(x, z) + v \tag{21}$$

where $y \in \mathbf{R}^m$ and $z \in \mathbf{R}^m$ represent problem-specific data and $v \in \mathbf{R}^m$ is the model noise. We assume v is independent and normally distributed with zero mean and covariance $P \in \mathbf{R}^{m\times m}$. $x \in \mathbf{R}^n$ denotes the model parameters. In many scenarios, we have some prior knowledge on x. Here, we assume x comes from a normal distribution parameterized by mean $\mu \in \mathbf{R}^n$ and covariance $\Sigma \in \mathbf{R}^{n\times n}$. With a data model linear in its parameters, the optimization problem is given by

$$\underset{x}{\text{minimize}} \quad \|x - \mu\|^2_{\Sigma^{-1}} + \|y - Ax\|^2_{P^{-1}} \tag{22}$$

$y \in \mathbf{R}^m$ denotes the observed data, and $A \in \mathbf{R}^{m\times n}$ represents the problem data. We randomly generated the problem data and show convergence rate of the SOVA method with $x \in \mathbf{R}^{2010}$, $A \in \mathbf{R}^{5371\times 2010}$, and $y \in \mathbf{R}^{2010}$ in Figure 7. SOVA converges more than three orders of magnitude faster than the consensus ADMM algorithm, despite the large number of agents. Rapid convergence in SOVA enables efficient re-optimization of model parameters for deploying predictive models in finance, medicine, and other applications.

VI. Conclusions

In previous ADMM methods, distributed agents have to optimize unnecessarily over all problem variables which are neither related to their data nor required for their role. We present the Separable Optimization Variable ADMM

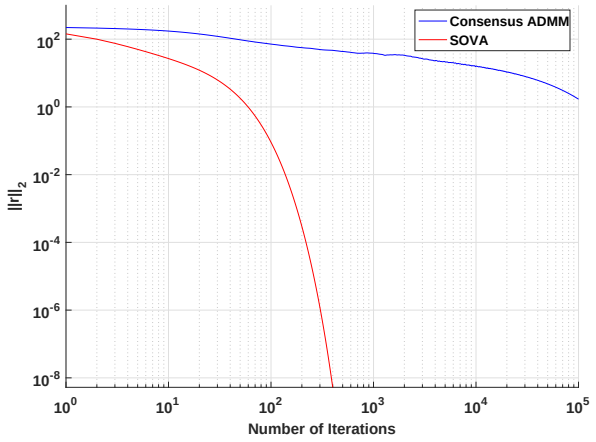

Fig. 7. Convergence rate of the consensus ADMM and Separable Optimization Variable ADMM (SOVA) methods on a maximum a posteriori (MAP) estimation problem with data distributed among 1000 agents. SOVA converges more than three orders of magnitudes faster than consensus ADMM.

(SOVA) method where each agents optimizes over a smaller subset of variables, resulting in highly efficient computation, communication, and data storage. We show superior convergence properties of the SOVA method to the centralized optimal values compared to consensus ADMM methods with different communication constraints. Future work will establish analytical bounds on the convergence rates of the SOVA method and apply SOVA to non-convex problems.

REFERENCES

[1] G. B. Dantzig and P. Wolfe, "Decomposition principle for linear programs," *Operations research*, vol. 8, no. 1, pp. 101–111, 1960.

[2] N. Z. Shor, *Minimization methods for non-differentiable functions*. Springer Science & Business Media, 2012, vol. 3.

[3] A. Nedic and D. P. Bertsekas, "Incremental subgradient methods for nondifferentiable optimization," *SIAM Journal on Optimization*, vol. 12, no. 1, pp. 109–138, 2001.

[4] A. Nedic and A. Ozdaglar, "Distributed subgradient methods for multi-agent optimization," *IEEE Transactions on Automatic Control*, vol. 54, no. 1, p. 48, 2009.

[5] A. Nedi and A. Ozdaglar, *Cooperative distributed multi-agent optimization*. Cambridge University Press, 2009, p. 340386.

[6] A. Agarwal, M. J. Wainwright, and J. C. Duchi, "Distributed dual averaging in networks," in *Advances in Neural Information Processing Systems*, 2010, pp. 550–558.

[7] A. Nedic, A. Ozdaglar, and P. A. Parrilo, "Constrained consensus and optimization in multi-agent networks," *IEEE Transactions on Automatic Control*, vol. 55, no. 4, pp. 922–938, 2010.

[8] W. Shi, Q. Ling, K. Yuan, G. Wu, and W. Yin, "On the linear convergence of the admm in decentralized consensus optimization," *IEEE Transactions on Signal Processing*, vol. 62, no. 7, pp. 1750–1761, 2014.

[9] R. T. Rockafellar, "Augmented lagrangians and applications of the proximal point algorithm in convex programming," *Mathematics of operations research*, vol. 1, no. 2, pp. 97–116, 1976.

[10] ——, "Monotone operators and the proximal point algorithm," *SIAM journal on control and optimization*, vol. 14, no. 5, pp. 877–898, 1976.

[11] G. Chen and M. Teboulle, "A proximal-based decomposition method for convex minimization problems," *Mathematical Programming*, vol. 64, no. 1-3, pp. 81–101, 1994.

[12] J. Eckstein and M. Fukushima, "Some reformulations and applications of the alternating direction method of multipliers," in *Large scale optimization*. Springer, 1994, pp. 115–134.

[13] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein *et al.*, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends® in Machine learning*, vol. 3, no. 1, pp. 1–122, 2011.

[14] F. Iutzeler, P. Bianchi, P. Ciblat, and W. Hachem, "Asynchronous distributed optimization using a randomized alternating direction method of multipliers," in *52nd IEEE conference on decision and control*. IEEE, 2013, pp. 3671–3676.

[15] Y. Zhang and M. M. Zavlanos, "A consensus-based distributed augmented lagrangian method," in *2018 IEEE Conference on Decision and Control (CDC)*. IEEE, 2018, pp. 1763–1768.

[16] G. Mateos, J. A. Bazerque, and G. B. Giannakis, "Distributed sparse linear regression," *IEEE Transactions on Signal Processing*, vol. 58, no. 10, pp. 5262–5276, 2010.

[17] T.-H. Chang, M. Hong, and X. Wang, "Multi-agent distributed optimization via inexact consensus admm," *IEEE Transactions on Signal Processing*, vol. 63, no. 2, pp. 482–497, 2014.

[18] S. Ma, "Alternating proximal gradient method for convex minimization," *Journal of Scientific Computing*, vol. 68, no. 2, pp. 546–572, 2016.

[19] B. He, L.-Z. Liao, D. Han, and H. Yang, "A new inexact alternating directions method for monotone variational inequalities," *Mathematical Programming*, vol. 92, no. 1, pp. 103–118, 2002.