A Consensus-Based Distributed Augmented Lagrangian Method

Yan Zhang and Michael M. Zavlanos

Abstract—In this paper, we propose a distributed algorithm to solve multi-agent constrained optimization problems. Specifically, we employ the recently developed Accelerated Distributed Augmented Lagrangian (ADAL) algorithm that has been shown to exhibit faster convergence rates in practice compared to relevant distributed methods. Distributed implementation of ADAL depends on separability of the global coupling constraints. Here we extend ADAL so that it can be implemented distributedly independent of the structure of the coupling constraints. For this, we introduce local estimates of the global constraint functions and multipliers and employ a finite number of consensus steps between iterations of the algorithm to achieve agreement on these estimates. The proposed algorithm can be applied to both undirected or directed networks. Theoretical analysis shows that the algorithm converges at rate \(O(1/k)\) and has steady error that is controllable by the number of consensus steps. Our numerical simulation shows that it outperforms existing methods in practice.

I. INTRODUCTION

Distributed optimization algorithms decompose an optimization problem into smaller, more manageable subproblems that can be solved in parallel by a group of agents or processors. For this reason, they are widely used to solve large-scale problems arising in areas as diverse as wireless communications, optimal control, planning, and machine learning, to name a few. In this paper, we consider the optimization problem

\[
\min \ F(x) \triangleq \sum_{i=1}^{N} f_i(x_i) \quad \text{s.t.} \quad \sum_{i=1}^{N} A_i x_i = b, \ x_i \in \mathcal{X}_i, \forall i
\]

where \(x = [x_1^T, x_2^T, \ldots, x_N^T]^T, x_i \in \mathbb{R}^{n_i}\) is the local decision variable, \(f_i(x_i)\) is the local objective function, and \(N\) is the total number of agents. The function \(f_i(x_i)\) is convex and possibly nonsmooth. All agents are globally coupled by the equality constraint \(\sum_{i=1}^{N} A_i x_i = b\). Each agent only has access to its own objective \(f_i(x_i)\), constraint matrix \(A_i \in \mathbb{R}^{m_i \times n_i}\), and constraint set \(\mathcal{X}_i\). All agents have access to the parameters \(N\) and \(b\). We are interested in finding a distributed solution to problem (1) in which all agents communicate only with their 1-hop neighbors. Several distributed algorithms have been proposed to solve problem (1), including dual decomposition, distributed saddle point methods, and distributed Augmented Lagrangian methods.

Dual decomposition deals with the dual problem

\[
\min_{\lambda} - \sum_{i} \phi^i(\lambda) \triangleq - \sum_{i} \{\Lambda^i(\lambda) + \frac{1}{N} b^T \lambda\}, \quad (2)
\]

where \(\Lambda^i(\lambda) = \min_{x_i \in \mathcal{X}_i} f_i(x_i) + \lambda^T A_i x_i\). Problem (2) is a consensus optimization problem and can be solved using, e.g., the methods proposed in [1–4]. While these methods only solve the dual problem (2), the Consensus-Dual Decomposition method in [5] also shows convergence of the primal solution for problem (1). Similarly, Consensus-ADMM in [6,7] introduces consensus constraints and applies ADMM to solve problem (2) and proves convergence of the primal solution for problem (1). Besides dual decomposition, distributed saddle-point methods [8–12] have also been proposed to solve (1). Specifically, in [8,10,12] all agents need to have knowledge of the global constraint, while in [9,11] the agents keep local estimates of the global constraint function and multiplier and employ consensus to agree on those estimates. Since Consensus Saddle Point Dynamics can be viewed as an inexact dual method, this method suffers the same slow convergence rate of the dual method [9].

The Augmented Lagrangian method (ALM) [13] smoothes the dual function and, therefore, converges faster than the dual method. Recently, several distributed ALMs have been proposed [14–17]. Among these methods, the Accelerated Distributed Augmented Lagrangian (ADAL) method, first developed for convex optimization problems [16,17] and later extended to non-convex problems [18] and problems with noise [19], has been shown to converge faster than other distributed ALMs in practice. However, these methods either require separability of the equality constraint in (1), or require local knowledge of the global constraint function, as in the indirect method discussed in [20]. In this paper, we develop a consensus-based ADAL (C-ADAL) method that can be implemented without these requirements. By introducing local estimates of the global constraint function and multipliers and applying a finite number of consensus steps on these local estimates during every iteration, we show that C-ADAL converges at rate \(O(1/k)\) and the final primal optimality and feasibility are controllable by the number of consensus steps. In numerical experiments, C-ADAL is demonstrated to outperform existing methods.

The rest of the paper is organized as follows. In Section II, we discuss assumptions on problem (1) and formally present the C-ADAL algorithm. In Section III, we analyze the convergence of the algorithm. In Section IV, we present comparative numerical experiments on a distributed estimation problem. In Section V, we conclude the paper.

II. PROBLEM FORMULATION

In this section, we first discuss some assumptions that are common to consensus-based algorithms; see, e.g., [1,2,5].

\[\text{Department of Mechanical Engineering and Materials Science, Duke University, Durham, NC 27708, USA. yan.zhang2,michael.zavlanos@duke.edu} \]
Algorithm 1: ADAL

Require: Initialize the multiplier $\lambda^0$ and primal variable $x^0$.
Set $k = 0$.
1: Agent $i$ computes $\hat{x}^k = \arg\min_{x \in \mathcal{X}_i} \Lambda_i^k(x_i; x_{-i}^k, \lambda^k)$.
2: Agent $i$ computes $x_{i+1}^k = x_i^k + \tau(\hat{x}_i^k - x_i^k)$, Update the multiplier: $\lambda_{i+1}^k = \lambda_i^k + \tau\rho(\sum_i A_ix_{i+1}^k - b)$.
4: $k \leftarrow k + 1$, go to step 1.

Algorithm 2: C-ADAL

Require: Initialize the local multiplier $\lambda_0^0$ and primal variable $x_0^i \in \mathcal{X}_i$. Set $y_0^i = A_ix_0^i$ and $k = 0$.
1: Agent $i$ communicates with its neighbors and runs $\alpha$ consensus steps on the variables $\lambda_i^k$ and $y_i^k$, i.e.,
\[ \hat{\lambda}_i^k = \sum_j [W(k)\alpha]_{ij} \lambda_i^k, \quad \hat{y}_i^k = \sum_j [W(k)\alpha]_{ij} y_i^k. \] (4a)
2: Agent $i$ computes $\hat{x}_i^k = \arg\min_{x \in \mathcal{X}_i} \Lambda_i^k(x_i; x_{-i}^k, \hat{y}_i^k, \hat{\lambda}_i^k)$.
3: Agent $i$ computes $x_{i+1}^k = x_i^k + \tau(\hat{x}_i^k - x_i^k)$.
4: Agent $i$ updates the variables $y_i^k$ and $\lambda_i^k$ by
\[ y_{i+1}^k = y_i^k + A_ix_{i+1}^k - A_ix_i^k, \quad \lambda_{i+1}^k = \lambda_i^k + \tau\rho(Ny_{i+1}^k - b). \] (4b)
5: $k \leftarrow k + 1$, go to step 1.

A. Evolution of $\lambda_i^k$ and $y_i^k$

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ denote the network of agents, where $\mathcal{V}$ is the index set of vertices $\{1, \ldots, N\}$ and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of edges. We assume that the graph $\mathcal{G}$ is fixed and directed. An edge $(i, j) \in \mathcal{E}$ means that node $i$ can receive information from node $j$. We assign weight $W_{ij}$ to the edge $(i, j)$ so that $W_{ij} > 0$ if $(i, j) \in \mathcal{E}$ and $W_{ij} = 0$ otherwise. We also define the weight matrix $W$, where $W_{ij}$ denotes its $(i, j)$th entry. In addition, we make the following assumption:

Assumption III.1. $W$ is doubly stochastic. That is, $W_{ij} \geq 0$, for all $i, j$, $W1 = 1$ and $1^T W = 1^T$, where $1$ is a column vector with all entries equal to $1$. Furthermore, we assume that there exists a $\beta > 0$ such that $\|W - \frac{1^T}{N}\| \leq \beta < 1$.

Rather than assuming that the network is undirected and the spectral radius $\rho(W - \frac{1^T}{N}) \leq \gamma < 1$ as in [1,5], here we assume a directed graph with the spectral norm $\|W - \frac{1^T}{N}\| \leq \beta < 1$. This condition implies that $\rho(W - \frac{1^T}{N}) \leq \gamma < 1$, and suggests that $W$ is irreducible and the underlying directed graph is strongly connected, [21]. It is simple to see that under Assumption III.1, Lemma 2 in [1] still holds with $\gamma$ replaced by $\beta$ defined here and, furthermore, Lemma 3 in [1] also holds. These results are used in our analysis. Using Assumption III.1, we can show the following lemma:

Lemma III.2. Let Assumption III.1 hold. We have that
\[ \lambda_i^0 = \frac{1}{N} \sum_i \lambda_i^0 \quad \text{and} \quad y_i^0 = \frac{1}{N} \sum_i y_i^0. \] (5)

Proof. Recalling that $\lambda_i^0 = \frac{1}{N} \sum_i \lambda_i^0$ and the update in (4a), to show the first equality in (5), it suffices to show that
\[ \frac{1}{N} \sum_i \lambda_i^k = \frac{1}{N} \sum_i \lambda_i^k = \frac{1}{N} \sum_i [W(k)\alpha]_{ij} \lambda_j^k. \] (6)

that $W^α1 = W^{α-1}W1 = W^{α-1}1$. Iterating results in $W^1 = 1$. Showing that $I^T W^{α} = I^T$ is similar. Meanwhile, it is straightforward to see that since all entries in $W$ are nonnegative, all entries in $W^α$ are also nonnegative. Therefore, $W^α$ is also doubly stochastic.

We can now show (6) by switching the order of the summations on the right hand side of (6) as $\frac{1}{N} \sum_i \sum_j W(k)_{ij} \lambda_{j}^k = \frac{1}{N} \sum_j \sum_i W(k)_{ij} \lambda_{i}^k = \frac{1}{N} \sum_i \lambda_{i}^k$. The final step follows from the fact that $W^α$ is doubly stochastic. This proves the first equality in (5). The second equality in (5) can be shown in a similar way.

Next, we show a conservation property on $y_k^i$.

**Lemma III.3.** Let Assumption III.1 hold. Then, we have that at each iteration $k$, $N y_k^i = \sum_i A_i x_{k}^i$.

**Proof.** We show this lemma by mathematical induction. From the initialization of Algorithm 2, we have that $N y_0^i = \sum_i y_i^i = \sum_i A_i x_{0}^i$. Therefore, the lemma holds for $k = 0$.

Next assume that $N y_{k-1}^i = \sum_i A_i x_{k-1}^i$ holds for $k \geq 0$.

We show that $N y_k^i = \sum_i A_i x_{k}^i$. We have that

$$N y_k^i = \sum_i \sum_j W(k)_{ij} y_j^i = \sum_i \sum_j (A_i x_{k}^i - x_{k}^i) = \sum_i A_i x_{k}^i,$$

which completes the proof.

For the purpose of analysis, we introduce an augmented multiplier $\lambda^k = \lambda_0^k + \rho (1 - \tau) r(x_k^i)$, where $r(x_k^i) = \sum_i A_i x_{k}^i - b$ is the residual of the constraint. Next, we present the problem $\lambda^k$ and $\lambda^k$ evolve in C-ADAL.

**Lemma III.4.** Let Assumption III.1 hold. Then we have that

$$\lambda_{k}^{i+1} = \lambda_{k}^{i} + \tau \rho r(x_{k+1}^i) \text{ and } \bar{\lambda}_{k}^{i+1} = \bar{\lambda}_{k}^{i} + \tau \rho r(\hat{x}_k^i).$$

**Proof.** First we show that $\lambda_{k}^{i+1} = \lambda_{k}^{i} + \tau \rho r(x_{k+1}^i)$. Recalling the definition of $\lambda_{k}^{i+1}$ and the update in (4b), we obtain that

$$\lambda_{k}^{i+1} = \frac{1}{N} \sum_i [\bar{\lambda}_{k}^{i} + \tau \rho N (y_k^i - b)].$$

Meanwhile, using Lemma III.3, we have that $\frac{1}{N} \sum_i N y_k^i = \sum_i y_k^i = \sum_i A_i x_{k}^i$. Substituting this in (9) we obtain that $\lambda_{k}^{i+1} = \lambda_{k}^{i} + \tau \rho (\sum_i A_i x_{k}^i - b)$. This proves the first equality in (8). To show the second equality in (8), recall the definition of $\bar{\lambda}_k^i$ and observe that

$$\bar{\lambda}_{k}^{i+1} = \bar{\lambda}_{k}^{i} + \tau \rho r(\hat{x}_k^i).$$

The first line is due to the first equality in (8). According to the line 3 in Algorithm 2, it is simple to see that $r(x_{k+1}^i) = (1 - \tau) r(x_{k}^i) + \tau r(\hat{x}_k^i)$. Plugging this into (10), we obtain that $\lambda_{k}^{i+1} = \lambda_{k}^{i} + \rho (1 - \tau) r(x_{k}^i) + \rho \tau r(\hat{x}_k^i) = \bar{\lambda}_{k}^{i} + \tau \rho r(\hat{x}_k^i)$. This completes the proof.

**B. Consensus Error**

In this section, we show that after a finite number of consensus iterations, the disagreement errors between the local variables $y_k^i$ (respectively, $\lambda_k^i$) and the global average $y_k^0$ (respectively, $\lambda_k^0$) always remain bounded. Furthermore, this bound can be made arbitrarily small by choosing a proper number of consensus steps $\alpha$. For this, we need the following assumption on initialization of Algorithm 2.

**Assumption III.5.** Given a small positive value $\epsilon$, for all $i$, it holds that $||\lambda_k^0 - \lambda^0|| \leq \epsilon$ and $||y_k^0 - y^0|| \leq \epsilon$.

To satisfy the Assumption III.5, we can run enough consensus steps to initialize the algorithm. Since such results are well-known, we refer the reader to [21,22,23]. A consequence of Assumption III.5 is that $||\lambda_k^0 - \lambda^0|| \leq \epsilon$ and $||y_k^0 - y^0|| \leq \epsilon$. This can be easily seen by the convexity of the $|| \cdot ||$ function. This result will be useful in the following analysis.

Next we study the boundness of the disagreement errors of $y_k^i$ (or $\lambda_k^i$). The update of $y_k^i$ (or $\lambda_k^i$) can be viewed as a consensus step with local perturbation $y_k^i = A_i x_{k}^i + A_i x_{k-1}^i$ (or $\lambda_k^i = \rho (N y_{k-1}^i - b)$) at iteration $k$. According to Lemma 3 in [1], if $||\eta_k^i|| \leq B, ||\hat{y}_k^i - y_k^i|| \leq \epsilon$ for all $i$, and $\alpha \geq (log(\epsilon) - log(\sqrt{N}m(\epsilon + B))/log(\beta))$, then we have that $||y_k^0 - y_0^0|| \leq \epsilon$ for all $i$. Therefore, in order to upper bound the disagreement error, we show that $\eta_k^i$ is bounded for all $k \geq 0$ in the next lemma.

**Lemma III.6.** Let Assumptions II.1 and III.1 hold. Then, for all $i$ and $k \geq 0$, there exists a constant $B_< y_k^i$ that satisfies

$$||A_i x_{k+1}^i - A_i x_k^i||_\infty \leq B_y,$$

Meanwhile, for any $\epsilon > 0$, let $\alpha$ satisfy $\alpha \geq (log(\epsilon) - log(4\sqrt{N}m(\epsilon + B))/log(\beta))$. Then, there exists a constant $B_\lambda$ that, for all $i$ and $k \geq 0$, satisfies

$$||\tau \rho (N y_{k+1}^i - b)||_\infty \leq B_\lambda.$$  

**Proof.** First we show the bound in (11). For this, it suffices to show that $x_k^i \in \chi_i^\epsilon$ for all $k \geq 0$. Then, because $\chi_i^\epsilon$ is compact, it is straightforward to see the result. According to line 3 in Algorithm 2, we have that $x_k^{i+1} = (1 - \tau) x_k^i + \tau x_0^i$. Therefore, we need to show that $x_k^{i+1} \in \chi_i^\epsilon$ if both $x_k^i$ and $\hat{x}_k^i$ belong to $\chi_i^\epsilon$. The latter condition is satisfied by line 2 in Algorithm 2, while the former one is satisfied when $k = 0$ at initialization of Algorithm 2. Therefore, it is simple to show that $x_k^i \in \chi_i^\epsilon$ for all $k \geq 0$ using mathematical induction.

Next, we prove the bound in (12). We have that

$$||y_{k+1}^i - b||_\infty = ||N (y_k^i + A_i (x_k^i - x_0^i)) - b||_\infty \leq ||N y_k^i - b||_\infty + ||A_i (x_k^i - x_0^i)||_\infty,$$

where the inequality is due to the triangle inequality. The second term in (13) is upper bounded due to Assumption II.1. Meanwhile, we can expand the first term in (13) as $||N y_k^i - b||_\infty = ||N (\hat{y}_k^i - y_k^i) + (N y_k^0 - b)\|_\infty \leq N \|\hat{y}_k^i - y_k^0\|_\infty + \|N y_k^0 - b\|_\infty$, where the inequality is due to the triangle inequality. Choosing $\alpha$ specified in Lemma III.6, according to Lemma 3 in [1] and Lemma III.3, we can show the bound in (12). This completes the proof.
The following theorem provides upper bounds on the disagreement errors of $\hat{y}_i^k$ and $\hat{x}_i^k$.

**Theorem III.7.** Let Assumptions III.1 and III.5 hold, and define $B_p = \max\{B_p, B_3\}$. If we choose $\alpha$ to satisfy $\alpha \geq (\log(c) - \log(4N\sqrt{\mathbb{E}(e + B_p)})/\log(\beta))$, for all $i$ and $k \geq 0$, we have $\|\hat{y}_i^k - y_i^k\|, \|\hat{x}_i^k - x_i^k\| \leq \epsilon$.

**Proof.** The proof follows from combining Lemma 3 in [1] and Lemma III.6 and using Assumption III.5.

**C. Primal optimality and feasibility**

In this section, we use a Lyapunov function to obtain the convergence rate of Algorithm 2 as well as the primal optimality and feasibility of the global solution. Specifically, we define the Lyapunov function as

$$\phi^k(\lambda) = \rho \sum_{i} \|A_i(x_i^k - x_i^*)\|^2 + \frac{1}{\rho} \|\hat{x}_i^k - x_i^*\|^2. \quad (14)$$

The following lemma characterizes the decrease of the Lyapunov function at every iteration $k$.

**Lemma III.8.** Let Assumption III.1 hold and define $q = \max\{q_1, q_2, \ldots, q_m\}$, where $q_1$ is the degree of the l-th constraint, that is the number of agents involved in the l-th row of $A_i$. Moreover, let $\tau \in (1, \frac{1}{\lambda})$ and assume that $\alpha$ satisfies the condition in Theorem III.7. Then, for any $\lambda \in \mathbb{R}^m$ and $k \geq 0$, we have that

$$F(\hat{x}^k) - F(x^*) + \langle \lambda, r(\hat{x}^k) \rangle \leq \frac{1}{2\tau} (\phi^k(\lambda) - \phi^{k+1}(\lambda)) + C\epsilon,$$

where $C = N(B_3B_2(1 + \rho N))$. Adding and subtracting $\sum_{j \neq i} A_j\hat{x}_j^k$ to the term $A_i\hat{x}_i^k + \sum_{j \neq i} A_jx_j^k - b$ on the left hand side of (17), adding $\langle \lambda, r(\hat{x}^k) \rangle$ to both sides of this inequality, recalling that $\sum A_ix_i^* = b$ and $r(x) = \sum A_ix_i - b$, and rearranging terms, we have that

$$\langle \lambda - \lambda^*_a, r(\hat{x}^k) \rangle - \rho\|r(\hat{x}^k)\|^2 + \rho \sum_j \langle A_j\hat{x}_j^k - A_j\hat{x}_j^k, A_i(x_i^k - \hat{x}_i^k) \rangle + C\epsilon \geq F(\hat{x}^k) - F(x^*) + \langle \lambda, r(\hat{x}^k) \rangle. \quad (18)$$

Therefore, to show (15), it suffices to prove that the left hand side in (18) is upper bounded by $\frac{1}{2\tau} (\phi^k(\lambda) - \phi^{k+1}(\lambda)) + C\epsilon$. To show this, expanding $\phi^k(\lambda) - \phi^{k+1}(\lambda)$, applying Lemma III.4, and dividing both sides by $2\tau$, we have that

$$\frac{1}{2\tau} (\phi^k(\lambda) - \phi^{k+1}(\lambda)) = \rho \sum_{i} \langle A_i(x_i^k - x_i^*), A_i(x_i^k - \hat{x}_i^k) \rangle - \langle \lambda - \hat{x}_i^k, r(\hat{x}^k) \rangle - \frac{\tau\epsilon}{2} \rho \sum_{i} \|A_i(x_i^k - \hat{x}_i^k)\|^2 + \|r(\hat{x})\|^2. \quad (19)$$

Next we manipulate the left hand side in (18) so that it can be related to $\frac{1}{2\tau} (\phi^k(\lambda) - \phi^{k+1}(\lambda))$ in (19). Specifically, adding and subtracting $-\|r(x^*)\|\rho(\hat{x}^k, r(x^*))$ to the term $(\lambda - \lambda^*_a, r(\hat{x}^k))$, we have that $\langle \lambda - \lambda^*_a, r(\hat{x}^k) \rangle$ can be replaced by $\langle \lambda - \lambda^*_a, r(x^*) \rangle + (1 - \tau)\rho(r(x^*), r(\hat{x}^k))$. 

Adding and subtracting $\rho \sum_i \langle A_i(x_i^k - x_i^*), A_i(x_i^k - \hat{x}_i^k) \rangle$ to the third term of (18) and recalling that $r(x) = \sum_i A_ix_i - b$, we can replace the third term in (18) with

$$-\rho \|r(x^*) - r(\hat{x}^k), r(\hat{x}^k)\| - \rho \sum_{i} \langle A_i(x_i^k - x_i^*), A_i(x_i^k - \hat{x}_i^k) \rangle + \rho \sum_i \|A_i(x_i^k - x_i^*), A_i(x_i^k - \hat{x}_i^k)\|^2. \quad (20)$$

Substituting (20) and (21) into (18) and rearranging terms, we have that

$$\langle \lambda - \lambda^*_a, r(x^*) \rangle - \tau\rho(r(x^*), r(\hat{x}^k)) + (1 - \tau)\rho\|r(\hat{x}^k)\|^2 + \rho \sum_{i} \langle A_i(x_i^k - x_i^*), A_i(x_i^k - \hat{x}_i^k) \rangle - \rho \sum_{i} \|A_i(x_i^k - \hat{x}_i^k)\|^2 - \rho\|r(\hat{x}^k)\|^2 + C\epsilon \geq F(\hat{x}^k) - F(x^*) + \langle \lambda, r(\hat{x}^k) \rangle. \quad (22)$$
Theorem III.9. Let Assumptions II.1, II.2, III.1 and III.5 hold and let $\alpha$ satisfy the condition in Theorem III.7. Then, we have the following bound on primal optimality of $\hat{x}^K$:

$$-(\frac{1}{2\tau K} \phi^0(2\alpha^*)+C\epsilon) \leq F(\hat{x}^K) - F(x^*) \leq \frac{1}{2\tau K} \phi^0(0)+C\epsilon.$$  

(23)

Moreover, we have the following bound on primal feasibility:

$$\|r(\hat{x}^K)\| \leq \frac{1}{2\tau K} \left[ \rho \sum_i \|A_i(x^0_i - x^*_i)\|^2 \right.$$

$$+ \frac{2}{\rho} (\|\lambda^0 - \lambda^*\|^2 + 1) + C\epsilon.$$  

(24)

\textbf{Proof.} Summing both sides of (15) in Lemma III.8 for $k = 0, 1, \ldots, K-1$ and dividing both sides by $K$, we obtain

$$\frac{1}{K} \sum_k F(\hat{x}^k) - F(x^*) + \langle \lambda, r(\hat{x}^k) \rangle \leq \frac{1}{2\tau K} \phi^0(0)+C\epsilon. $$

After rearranging terms, we get

$$F(\hat{x}^k) - F(x^*) + \langle \lambda, r(\hat{x}^k) \rangle \leq \frac{1}{2\tau K} \phi^0(\lambda) + C\epsilon.$$  

(25)

Next we use (25) to show the bound (23) and (24). To show the upper bound in (23), since $\lambda \in \mathbb{R}^m$ in (25), we select $\lambda = 0$ and obtain $F(\hat{x}^k) - F(x^*) \leq \frac{1}{2\tau K} \phi^0(0)+C\epsilon$. To prove the lower bound in (23), we recall that $(x^*, \lambda^*)$ is a saddle point that satisfies $F(x^*) \leq F(\hat{x}^k) + \langle \lambda^*, r(\hat{x}^k) \rangle$. Adding $\langle \lambda^*, r(\hat{x}^k) \rangle$ to both sides of the saddle point inequality and rearranging terms, we get

$$\langle \lambda^*, r(\hat{x}^k) \rangle \leq F(\hat{x}^k) - F(x^*) + \langle 2\lambda^*, r(\hat{x}^k) \rangle.$$  

(26)

The right hand side in (26) can be upper bounded by selecting $\lambda = 2\alpha^*$ in (25). Specifically, we have that $F(x^k) - F(\hat{x}^k) \leq \langle \lambda^*, r(\hat{x}^k) \rangle \leq \frac{1}{2\tau K} \phi^0(2\alpha^*) + C\epsilon$, which completes the proof of (23).

Next we show the bound (24) on primal feasibility. Letting $\lambda = \lambda^* + \frac{r(\hat{x}^k)}{\|r(\hat{x}^k)\|}$ in (25), we have that

$$F(\hat{x}^k) - F(x^*) + \langle \lambda^*, r(\hat{x}^k) \rangle + \|r(\hat{x}^k)\| \leq \frac{1}{2\tau K} \phi^0(\lambda^* + \frac{r(\hat{x}^k)}{\|r(\hat{x}^k)\|}) + C\epsilon.$$  

(27)

Since $(x^*, \lambda^*)$ is the saddle point, we have that $F(\hat{x}^k) - F(x^*) + \langle \lambda^*, r(\hat{x}^k) \rangle \geq 0$, and, therefore, these terms can be neglected. This gives $\|r(\hat{x}^k)\| \leq \frac{1}{2\tau K} \phi^0(\lambda^* + \frac{r(\hat{x}^k)}{\|r(\hat{x}^k)\|}) + C\epsilon$. Plugging the expression of $\phi^0(\lambda)$ in (14) into this upper bound completes the proof. \qed

The error terms in Theorem III.9 are due to the disagreement errors $\|\hat{\lambda}_i^k - \lambda^*_i\|$ and $\|\hat{y}_i^k - y^*_i\|$. Similar to the results in Lemma 3 in [1], as long as the perturbation in (4b) is uniformly bounded, it is straightforward to make the error term in Theorem III.9 uniformly bounded by an arbitrarily small value by choosing $\alpha$ large enough. Exponential convergence of consensus under time-varying digraphs with doubly stochastic weight matrices has been proved in [2,22]. Therefore, our results can be easily modified to fit into the time-varying network topology in [2,22].

In this section, we conduct numerical experiments to verify our theoretical analysis and compare the performance of C-ADAL to other existing methods. Specifically, we consider a network of $N$ agents and apply C-ADAL to solve the following distributed estimation problem:

$$\min \sum_i \|M_i x_i - y_i \|^2$$

s.t. $\sum_i A_i x_i = b$ and $l_i \leq x_i \leq u_i$, for all $i$,  

(28)

where $M_i \in \mathbb{R}^{n_i \times p}$ is the design matrix, $y_i \in \mathbb{R}^{n_i}$ is the observation vector, $x_i \in \mathbb{R}^p$, $A_i \in \mathbb{R}^{m \times p}$ and $b \in \mathbb{R}^m$.

To demonstrate the performance of C-ADAL, we compare it with consensus Dual Decomposition (C-DD) in [5], consensus Saddle Point Dynamics (C-SPD) in [9], consensus ADMM (C-ADMM) in [6], and an indirect version of ADMM (P-ADMM) in [15]. C-ADMM and P-ADMM can only be implemented on undirected graphs, while C-ADAL, C-DD, and C-SPD can be applied to directed graph. We conduct simulations on an undirected graph, so that C-ADAL can be compared with C-ADMM and P-ADMM. We study the case where we have $N = 10$ agents, $n_i = 5$, $p = 10$, and $m = 20$. The problem data are randomly generated. A chain graph is applied and the weight matrix $W$ is determined as in [21]. According to Theorem III.7, if $\epsilon = 0.1$, then $\alpha \approx 300$. However, in numerical experiments we have observed that C-ADAL behaves well for much smaller values of $\alpha$ for most randomly generated problems. Therefore, we select $\alpha = 10$.
Moreover, we provided numerical simulations showing that our algorithm outperforms existing methods in practice.

REFERENCES


for C-ADAL, C-DD, and C-SPD. The other parameters in these methods are optimized by trials. The primal optimality and feasibility for the direct iterate (or running average) are presented in Figures 1 (or Figure 2).

In Figure 1, the direct iterate of C-DD oscillates because this algorithm applies a subgradient method to maximize a nonsmooth dual function of (28). C-ADAL and C-ADMM outperform other methods in terms of primal optimality, while C-ADAL outperforms other methods in terms of primal feasibility. In Figure 2, we see that the running average of all methods converge. Specifically, in Figure 2(a), we observe that C-DD and C-SPD have relatively large errors due to the constant stepsize, [1,2,5]. C-ADAL outperforms other methods both in terms of primal optimality and feasibility.

V. CONCLUSIONS

In this paper, we proposed a distributed algorithm to solve multi-agent constrained optimization problems. Specifically, we employed the recently developed Accelerated Distributed Augmented Lagrangian (ADAL) algorithm that has been shown to exhibit faster convergence rates in practice compared to relevant distributed methods. Distributed implementation of ADAL depends on separability of the global coupling constraints. Here we extended ADAL so that it can be implemented distributedly independent of the structure of the coupling constraints. Our proposed algorithm can be applied to both undirected and directed networks. We showed that our algorithm converges at rate $O(1/k)$ and has steady error that is controllable by the number of consensus steps.