

Distributed Consensus Optimization with Consensus ALADIN

Xu Du, Jingzhe Wang*

Abstract—The paper proposes the Consensus Augmented Lagrange Alternating Direction Inexact Newton (Consensus ALADIN) algorithm, a novel approach for solving distributed consensus optimization problems (DC). Consensus ALADIN allows each agent to independently solve its own nonlinear programming problem while coordinating with other agents by solving a consensus quadratic programming (QP) problem. Building on this, we propose Broyden–Fletcher–Goldfarb–Shanno (BFGS) Consensus ALADIN, a communication-and-computation-efficient Consensus ALADIN. BFGS Consensus ALADIN improves communication efficiency through BFGS approximation techniques and enhances computational efficiency by deriving a closed form for the consensus QP problem. Additionally, by replacing the BFGS approximation with a scaled identity matrix, we develop Reduced Consensus ALADIN, a more computationally efficient variant. We establish the convergence theory for Consensus ALADIN and demonstrate its effectiveness through application to a non-convex sensor allocation problem.

Keywords: Consensus ALADIN, Distributed Consensus Optimization

I. INTRODUCTION

Distributed optimization problems with affine coupled constraints (DO) are generally formulated in the fashion of mathematical programming, where N separable objectives are linearly coupled by m equality constraints. In DO, $f_i : \mathbb{R}^{n_i} \rightarrow \mathbb{R}$ denotes the closed proper local objective function of each agent i , where $i \in \mathcal{I} = \{1, 2, \dots, N\}$. Formally, DO can be described as follows [1],

$$\begin{aligned} \min_{x_i \in \mathbb{R}^{n_i}, \forall i \in \mathcal{I}} \quad & \sum_{i=1}^N f_i(x_i) \\ \text{s.t.} \quad & \sum_{i=1}^N A_i x_i = b. \end{aligned} \quad (1)$$

Here, the coupling matrices $A_i \in \mathbb{R}^{n_i \times m}$ and the coupling parameter $b \in \mathbb{R}^m$ are given. The dimensions n_i s of local variables x_i s may differ. In recent years, DO has gained significant attention due to its successful applications to problems in signal processing [2], model predictive control [1], [3] and power grids [4], [5].

Distributed consensus optimization (DC) problems, which

are a special case of DO, are formulated as follows:

$$\begin{aligned} \min_{x_i, z \in \mathbb{R}^n, \forall i \in \mathcal{I}} \quad & \sum_{i=1}^N f_i(x_i) \\ \text{s.t.} \quad & x_i = z \mid \lambda_i. \end{aligned} \quad (2)$$

Here, $f_i : \mathbb{R}^n \rightarrow \mathbb{R}$ denotes closed and proper local objective function of agent i , and λ_i represents the dual variable associated with agent i . We introduce x and λ to stack x_i and $\lambda_i, \forall i$ as vectors respectively. The notation $(a|b)$ denotes the relation between the constraint ($a \in \mathbb{R}^n$) and the corresponding dual variable ($b \in \mathbb{R}^n$). The main difference between DO and DC problems is that DC includes a global variable z to which all private variables x_i s must converge. Applications of DC include, but are not limited to, federated learning (FL) [6] and sparse signal processing [2]. As a milestone in DC research, [7] shows that Consensus Alternating Direction Method of Multiplier (Consensus ADMM) [2] achieves a linear convergence rate for DC problems that are strongly convex under some assumptions. We refer [8] as a survey paper for more details. It is worth noting that, similar to Consensus ADMM, other algorithms such as Decentralized Gradient Descent (DGD) [9], Exact First-Order Algorithm (EXTRA) [10] provide convergence guarantees for decentralized convex problems in the study of DC. However, a detailed discussion of decentralized optimization algorithms is beyond the scope of this paper.

In this paper, we focus on a recent algorithm, named Augmented Lagrangian based Alternating Direction Inexact Newton method (ALADIN) [11]. ALADIN was originally developed for solving DO (1) problems and can be viewed as a combination of ADMM and sequential quadratic programming (SQP) [12]. ALADIN guarantees local convergence for non-convex DO problems and global convergence for convex DO problems. To date, ALADIN has inspired several elegant successors [4], [13], [1], [14] and has demonstrated effectiveness in many applications [15], [4], [13], [3]. Despite these advancements, it remains an open question whether ALADIN can solve Problem (2) efficiently. This naturally leads us to ask: *Can we propose novel variants of ALADIN to solve DC (Problem (2)) efficiently while providing formal convergence guarantees?*

Contribution: In this paper, we first propose Consensus ALADIN, a novel algorithmic structure that aligns the ALADIN approach with the structure of Problem (2). Consensus ALADIN enables each agent to solve its own nonlinear programming problem while coordinating with other agents by solving a consensus quadratic programming (QP) problem. Building on this, we pro-

* Corresponding author.

Xu Du, is with the Institute of Mathematics, Henan Academy of Science, Zhengzhou, China. E-mail: duxu@hnas.ac.cn.

Jingzhe Wang is with the Department of Informatics & Networked Systems, School of Computing and Information, University of Pittsburgh, Pittsburgh, PA, USA. E-mail: jiw148@pitt.edu.

Xu Du is supported by High-level Talent Research Start-up Project Funding of Henan Academy of Sciences (Project NO.241819034).

pose Broyden–Fletcher–Goldfarb–Shanno (BFGS) Consensus ALADIN, a communication-and-computation-efficient variant of Consensus ALADIN. In BFGS Consensus ALADIN, we reconstruct the Hessian matrix of the subproblems using the BFGS approximation technique, thereby avoiding the need to transmit the full Hessian matrices. This enhances communication efficiency compared to directly transmitting the matrix in Consensus ALADIN. Furthermore, BFGS Consensus ALADIN leverages a closed-form solution for the consensus QP, resulting in enhanced computational efficiency. Additionally, based on BFGS Consensus ALADIN, we propose Reduced Consensus ALADIN, which replaces the Hessian matrix with a scaled identity matrix to further enhance computational efficiency. We establish convergence theory of Consensus ALADIN for Problem (2). Specifically, we provide a global convergence theory for convex DC problems and a local convergence theory for non-convex DC problems. Our numerical experiments on a non-convex sensor allocation problem demonstrate the effectiveness of our proposed algorithms.

Organization: The paper is structured as follows: Section II provides the preliminaries of ALADIN and its limitation for solving (2). In Section III, we introduce Consensus ALADIN, a novel family of algorithms designed to address Problem (2). Section IV provides the convergence analysis of Consensus ALADIN. In Section V, we perform numerical experiments. Section VI concludes this paper.

Notations: In this paper, $(\cdot)^-$ denotes the value from the previous iteration, while $(\cdot)^+$ represents the value from the current iteration. For ease of expression, $(\cdot)^k$ indicates the value of (\cdot) at the k -th iteration for the given algorithms. Additionally, $\|\cdot\|_M = \sqrt{(\cdot)^\top M (\cdot)}$ when $M \succ 0$.

II. BACKGROUND AND MOTIVATION

In this section, we first review the basics of ALADIN in Section II-A, followed by a motivating example in Section II-B.

A. Preliminaries of ALADIN

At a high level, ALADIN allows each agent to independently address its nonlinear programming (NLP) problem while communicating with other agents by solving a coupled QP problem. The detailed algorithmic structure of ALADIN is given in Algorithm 1.

In Algorithm 1, the initial step involves each agent updating its local minimizer to x_i^+ . Given x_i^+ , each agent then approximates the positive definite Hessian matrices B_i s, and evaluates the gradients g_i s of f_i s, as per (4) in Step 2). Upon receiving B_i and g_i from all agents, the master solves a large-scale convex coupled QP (5). The solution to this QP (5) includes the updated primal variables $(x_i^+ + \Delta x_i)$ s and dual variable μ in Step 3). Finally, the master communicates the updated primal and dual variables back to each agent, thereby concluding the current iteration in Step 4).

The convergence behavior of ALADIN can be summarized as follows: For non-convex DO problems [11], Algorithm 1 has local convergence guarantees if the linearly independent

Algorithm 1 Standard ALADIN

Initialization: Initial guess of primal and dual variables (ϵ_i, μ) .

Repeat:

1) Parallely solve local NLP:

$$x_i^+ = \arg \min_{x_i} f_i(x_i) + \mu^\top A_i x_i + \frac{\rho}{2} \|x_i - \epsilon_i\|^2. \quad (3)$$

2) Evaluate and upload the Hessian approximation $B_i \succ 0$ and the gradient g_i at x_i^+ :

$$\begin{cases} B_i \approx \nabla^2 f_i(x_i^+), \\ g_i = \rho (\epsilon_i - x_i^+) - A_i^\top \mu. \end{cases} \quad (4)$$

3) Solve the coupled QP on master side with the uploaded x_i^+ , B_i , g_i from each agent:

$$\begin{aligned} \min_{\Delta x_i, \forall i \in \mathcal{I}} \quad & \sum_{i=1}^N \frac{1}{2} \Delta x_i^\top B_i \Delta x_i + g_i^\top \Delta x_i \\ \text{s.t.} \quad & \sum_{i=1}^N A_i (x_i^+ + \Delta x_i) = b \mid \mu^+. \end{aligned} \quad (5)$$

4) Download:

$$\begin{cases} \mu \leftarrow \mu^+, \\ \epsilon_i \leftarrow x_i^+ + \Delta x_i. \end{cases} \quad (6)$$

constraint qualification (LICQ) and second order sufficient condition (SOSC) of Problem (1) are locally satisfied. In the case of convex problems, ALADIN has global convergence guarantees without requiring the smoothness of the objectives if the strong duality of Problem (1) holds [1]. For more details, we refer interested readers to [1].

B. Limitations of ALADIN for Solving DC

In this subsection, we analyze the limitations of ALADIN when directly applied to solve DC, which motivates us to present our new algorithm family.

In general, there are two straightforward approaches that may adapt ALADIN for DC. Specifically, the **first** approach applies ALADIN directly to (2). Such an approach, however, is clearly flawed. The primary issue is that ALADIN, originally tailored for DO problems (1), requires each variable to have a strictly positive Hessian approximation matrix $B_i \succ 0$ corresponding to its independent objective function. Problem (2) presents a challenge, as the global variable z is not linked to an independent objective function, thereby violating a fundamental principle of ALADIN.

The **second** way involves a more common approach to solving DC as mentioned in [16, Section 12], which suggests reformulating (2) to align with standard DO fashion and then adopting DO algorithms. The reformulation of (2) is elaborated as follows,

$$\begin{aligned} \min_{x_i \in \mathbb{R}^n, \forall i \in \mathcal{I}} \quad & f_1(x_1) + f_2(x_2) + \cdots + f_N(x_N) \\ \text{s.t.} \quad & \begin{array}{ccccccc} x_1 & -x_2 & & & & & = 0 \\ & x_2 & -x_3 & & & & = 0 \\ & \vdots & \vdots & & \vdots & & \vdots \\ & & & & x_{N-1} & -x_N & = 0 \\ -x_1 & & & & & x_N & = 0. \end{array} \end{aligned} \quad (7)$$

The linear constraints in Problem (7) can be described as $\sum_{i=1}^N A_i x_i = 0$, where $A_1 = (I, 0, \dots, 0, -I)^\top$, $A_2 = (-I, I, 0, \dots, 0)^\top, \dots, A_N = (0, \dots, 0, -I, I)^\top$. Here I denotes the identity matrix with proper dimension. We denote the set of such vectors as $A = \{A_i\}_{i \in \mathcal{I}}$. However, this approach has two limitations. First, the selection of $\{A_i\}$ is not unique, and these linear coupling matrices A_i s can be challenging to implement in practice. Second, employing Algorithm 1 to solve (7) is inefficient in terms of both communication and computational. Specifically, the communication overhead arises from transmitting local Hessian matrices $\nabla^2 f_i(x_i^+)$ and gradients $\nabla f(x_i^+)$ of the local NLPs at each iteration, as detailed in (4). Additionally, coordination requires solving a large-scale coupled QP problem, as outlined in (5), which results in significant computational overhead.

III. CONSENSUS AUGMENTED LAGRANGE ALTERNATING DIRECTION INEXACT NEWTON METHOD

In Subsection III-A, we present the consensus QP and our proposed Consensus ALADIN. Subsection III-B details Consensus ALADIN, consisting of two variants, namely BFGS Consensus ALADIN and Reduced Consensus ALADIN.

A. Consensus ALADIN: From Coupled QP to Consensus QP

Recall from Subsection II-B that ALADIN faces challenges in solving Problem (2) due to the complexity of communication and computation in Algorithm 1, as well as the inefficient implementation of coupling equality constraints in Equation (5) and (7). Drawing inspiration from Consensus ADMM [2], our first step towards making ALADIN viable in (2) involves reconstructing Equation (5). This leads to the formulation of our convex consensus QP, formally shown as follows,

$$\begin{aligned} \min_{\Delta x_i, z \in \mathbb{R}^n, \forall i \in \mathcal{I}} \quad & \sum_{i=1}^N \left(\frac{1}{2} \Delta x_i^\top B_i \Delta x_i + g_i^\top \Delta x_i \right) \\ \text{s.t.} \quad & x_i^+ + \Delta x_i = z \mid \lambda_i. \end{aligned} \quad (8)$$

Notably, similar to Consensus ADMM, (8) introduces a global variable z and couples all x_i s to z . Note that, Equation (8) differs from the original formulation in Equation (5) by incorporating one global primal variable z and N dual variables λ_i s.

After replacing Equation (5) with Equation (8), Algorithm 1 is now capable of operating in a consensus fashion. We name this Consensus ALADIN.

B. Algorithms

1) **BFGS Consensus ALADIN:** To improve the communication and computation efficiency of Consensus ALADIN, we detail BFGS Consensus ALADIN (a variant of Consensus ALADIN) in Algorithm 2.

Similar as Algorithm 1, in the first step, each agent updates its own local variable x_i^+ with a recovered λ_i^- and uploads it

¹One can optionally update the the Hessian matrices at iteration k with $K \in \mathbb{N}^+$, if $\log_K(k) \in \mathbb{N}^+$ [17, Section III-C].

Algorithm 2 BFGS Consensus ALADIN

Initialization: choose $\rho > 0$, initial guess $(\lambda_i, z, B_i = \rho I)$.

Repeat:

- 1) Each agent optimizes its own variable x_i locally and transmits it to the master

$$x_i^+ = \arg \min_{x_i} f_i(x_i) + \lambda_i^\top x_i + \frac{\rho}{2} \|x_i - z\|^2 \quad (9)$$

with

$$\lambda_i = B_i^-(x_i^- - z) - g_i. \quad (10)$$

- 2) Operations on the master side:

- a) Recover the gradient and BFGS Hessian from each x_i^+ :

$$\begin{cases} g_i = \rho(z - x_i^+) - \lambda_i & \text{(local gradient evaluation),} \\ s_i = x_i^+ - x_i^- & \text{(difference of local variables),} \\ y_i = g_i - g_i^- & \text{(difference of local gradient).} \end{cases} \quad (11)$$

- b) Modify the local gradient with $y_i = y_i + \theta(B_i s_i - y_i)$ where

$$\theta = \frac{0.2(s_i)^\top B_i s_i - (s_i)^\top y_i}{(s_i)^\top B_i s_i - (s_i)^\top y_i}$$

if $(y_i)^\top s_i \leq \frac{1}{5}(s_i)^\top B_i s_i$.

- c) BFGS Hessian approximation evaluation (optionally)¹:

$$B_i = B_i^- - \frac{B_i^- s_i s_i^\top B_i^-}{s_i^\top B_i^- s_i} + \frac{y_i y_i^\top}{s_i^\top y_i}. \quad (12)$$

- d) The master updates the global variable z with

$$z = \left(\sum_{i=1}^N B_i \right)^{-1} \sum_{i=1}^N (B_i x_i^+ - g_i). \quad (13)$$

to the master. Later, the master recovers the agents' gradient and Hessian approximation in the second step by using the damped BFGS technique ([12, Page 537]). Finally, the master updates the global variable z^+ with the Equation (13) and broadcasts it to the agents. Repeat these steps until z converges. Here, (10) and (13) are derived from the Karush-Kuhn-Tucker (KKT) system of Problem (8).

Remark 1 Note that, similar to Step 3) in Algorithm 1, (8) also requires the primal variable $x_i^+ \in \mathbb{R}^n$, the gradient $g_i \in \mathbb{R}^n$, and the local Hessian approximation $B_i \in \mathbb{R}^{n \times n}$. In total, this requires transmitting $(2Nn + Nn^2)$ floats. Specifically, when N is fixed, transmitting g_i requires Nn floats, while transmitting B_i requires Nn^2 floats. The transmission of these two components is the main factor affecting communication efficiency during the upload phase. Therefore, Step 2) of Algorithm 2 improves upload communication efficiency by avoiding the direct transmission of the gradient and Hessian approximation (4). Taking advantage of our design, BFGS Consensus ALADIN only requires transmitting Nn floats on the upload phase.

Remark 2 Solving the KKT system of Problem (8)

finds the inversion of a KKT matrix with dimension $((2N+1)n \times (2N+1)n)$. To alleviate this issue, we design a new method that updates the global variable z directly without updating the λ_i s and Δx_i s, avoiding solving (8) directly. The optimal solution of global variable z in Equation (8) has a closed form as Equation (13) and only relies on deriving the inversion of an $n \times n$ matrix.

2) Reduced Consensus ALADIN: To further improve the computational efficiency of (13), we next introduce our proposed Reduced Consensus ALADIN, as an extension of BFGS Consensus ALADIN. Our Reduced Consensus ALADIN differs from BFGS Consensus ALADIN in that the global variable z is updated based on the following equation,

$$z = \frac{1}{N} \sum_{i=1}^N \left(x_i^+ - \frac{g_i}{\rho} \right). \quad (14)$$

For Reduced Consensus ALADIN, the updates of x_i^+ , λ_i , g_i and z in Algorithm 2 can be easily replaced with those defined by the following equation,

$$\begin{cases} x_i^+ = \arg \min_{x_i} f_i(x_i) + \lambda_i^\top x_i + \frac{\rho}{2} \|x_i - z\|^2, \\ g_i = \rho(z - x_i^+) - \lambda_i, \\ z^+ = \frac{1}{N} \sum_{i=1}^N \left(x_i^+ - \frac{g_i}{\rho} \right), \\ \lambda_i^+ = \rho(x_i^+ - z^+) - g_i. \end{cases} \quad (15)$$

Note that, both two algorithms (Algorithm 2 and Equation (15)) belong to the class of Consensus ALADIN. As a supplement, the difference between the Reduced Consensus ALADIN and Consensus ADMM can be found in the appendix.

IV. CONVERGENCE ANALYSIS

In this section, we present the convergence analysis for Consensus ALADIN. Specifically, Section IV-A develops the global convergence theory for convex DC problems, while Section IV-B provides the local convergence analysis for non-convex DC problems. Our results for Consensus ALADIN are also applicable to BFGS Consensus ALADIN and Reduced Consensus ALADIN.

A. Global Convergence Analysis

We assume that the f_i s of (2) are closed, proper, and strictly convex. In the global convergence theory of Consensus ALADIN, the sub-problems are updated as,

$$x_i^+ = \arg \min_{x_i} f_i(x_i) + \lambda_i^\top x_i + \frac{1}{2} \|x_i - z\|_{B_i}^2 \quad (16)$$

instead of (9). For ease of analysis, we temporarily assume that the matrices $B_i \succ 0$ are constant.

For establishing the global convergence theory of Consensus ALADIN, we introduce the following *energy function*

[18] (also called Lyapunov function in [2, Appendix A] and [19]) with the unique optimal solution (z^*, λ^*) ,

$$\mathcal{L}(z, \lambda) = \sum_{i=1}^N \left(\|\lambda_i - \lambda_i^*\|_{B_i^{-1}}^2 + \|z - z^*\|_{B_i}^2 \right). \quad (17)$$

Note that the choice of energy function is not unique [20]. Next, we will establish the global convergence of Consensus ALADIN by demonstrating the monotonic decrease of the energy function (17).

Theorem 1 *Let the local objective f_i s of Problem (2) be closed, proper, strictly convex. Let the strong duality of Problem (2) hold. Let matrices $B_i \succ 0$ s be constant. Let $x_i^* = z^*$ denote the primal solution and λ^* denotes the dual solution of Problem (2), then*

$$\mathcal{L}(z^+, \lambda^+) \leq \mathcal{L}(z, \lambda) - 4\Pi \left(\sum_{i=1}^N \|x_i^+ - z^*\| \right) \quad (18)$$

is satisfied with the iterations of Algorithm 2. Here, Π is a class \mathcal{K} function [20].

Proof: First, we introduce the auxiliary functions

$$\begin{cases} \tilde{f}_i(\chi_i) = f_i(\chi_i) + (\lambda_i + B_i(x_i^+ - z))^T \chi_i, \\ G(\chi) = \sum_{i=1}^N \tilde{f}_i(\chi_i) + \sum_{i=1}^N (\chi_i - z^*)^T \lambda_i^*, \end{cases} \quad (19)$$

where $\chi_i \in \mathbb{R}^n$. Due to the strict convexity of both functions in (19), the following equation holds,

$$\begin{cases} \tilde{f}_i(x_i^+) \leq \tilde{f}_i(z^*) - \tilde{\Pi}_i(\|x_i^+ - z^*\|), \\ G(z^*) \leq G(x^+) - \hat{\Pi} \left(\sum_{i=1}^N \|x_i^+ - z^*\| \right), \end{cases} \quad (20)$$

where $Z^* = [(z^*)^T, \dots, (z^*)^T]^T \in \mathbb{R}^{nN}$, $\tilde{\Pi}_i(\cdot)$ and $\hat{\Pi}(\cdot)$ are class \mathcal{K} functions. By summing up the first equation of (20), the following equation is obtained,

$$\begin{aligned} & \sum_{i=1}^N (f_i(x_i^+) - f_i(z^*)) \\ & \leq \sum_{i=1}^N (\lambda_i + B_i(x_i^+ - z))^T (z^* - x_i^+) - \sum_{i=1}^N \tilde{\Pi}_i(\|x_i^+ - z^*\|). \end{aligned} \quad (21)$$

Similarly, from the second equation of (20), we have

$$\begin{aligned} & \sum_{i=1}^N (f_i(z^*) - f_i(x_i^+)) \\ & \leq \sum_{i=1}^N (x_i^+ - z^*)^T \lambda_i - \hat{\Pi} \left(\sum_{i=1}^N \|x_i^+ - z^*\| \right). \end{aligned} \quad (22)$$

Combine Equations (21) and (22), the following inequality can be obtained,

$$\begin{aligned} & \sum_{i=1}^N (\lambda_i - \lambda_i^* + B_i(x_i^+ - z))^T (x_i^+ - z^*) \\ & \leq -\Pi \left(\sum_{i=1}^N \|x_i^+ - z^*\| \right), \end{aligned} \quad (23)$$

where

$$\begin{aligned} & \Pi \left(\sum_{i=1}^N \|x_i^+ - z^*\| \right) \\ &= \hat{\Pi} \left(\sum_{i=1}^N \|x_i^+ - z^*\| \right) + \sum_{i=1}^N \tilde{\Pi}_i (\|x_i^+ - z^*\|) \in \mathcal{K}. \end{aligned}$$

In Algorithm 2, since $\lambda_i^+ = B_i(x_i^+ - z^+) - g_i$ (Equation (10)) and $g_i = B_i(z - x_i^+) - \lambda_i$ (Equation (11)), it is easy to show

$$x_i^+ = \frac{B_i^{-1}}{2}(\lambda_i^+ - \lambda_i) + \frac{1}{2}(z + z^+). \quad (24)$$

On the other hand, from the KKT system of Problem (8),

$$\sum_{i=1}^N \lambda_i = 0 \quad (25)$$

(25) is always guaranteed.

By plugging (24) and (25) into the left hand side of (23), it can be shown that

$$\begin{aligned} (23) &= \frac{1}{4} \sum_{i=1}^N \left(-\|\lambda_i - \lambda_i^*\|_{B_i^{-1}}^2 + \|\lambda_i^+ - \lambda_i^*\|_{B_i^{-1}}^2 \right) \\ &\quad + \frac{1}{4} (\|z^+ - z^*\|_{B_i}^2 - \|z - z^*\|_{B_i}^2) \\ &= \frac{1}{4} \mathcal{L}(z^+, \lambda^+) - \frac{1}{4} \mathcal{L}(z, \lambda) \\ &\leq -\Pi \left(\sum_{i=1}^N \|x_i^+ - z^*\| \right). \end{aligned} \quad (26)$$

This completes the proof. \blacksquare

Note that, for smooth and strongly convex f_i s, we can establish the global linear convergence theory of Consensus ALADIN.

Theorem 2 *Let the local objective f_i s of Problem (2) be closed, proper, m_f strongly convex and twice continuously differentiable such that the gradients g_i s exist. Let the strong duality of Problem (2) hold. Let matrices $B_i \succ 0$ s be constant. Let (z^*, λ^*) be the unique optimal primal and dual solution of Problem (2). There exists a sufficiently small $\delta > 0$ such that*

$$\delta \mathcal{L}(z^+, \lambda^+) \leq 4m_f \sum_{i=1}^N \|x_i^+ - z^*\|^2, \quad (27)$$

then the iteration of Algorithm 2 is linearly converging to (z^*, λ^*) .

Proof: Let f_i s be twice continuously differentiable and m_f strongly convex, then the following inequality satisfies

$$\begin{aligned} m_f \sum_{i=1}^N \|x_i^+ - z^*\|^2 &\leq \sum_{i=1}^N (x_i^+ - z^*)^\top (g_i - g_i^*) \\ &\stackrel{(23)=(26)}{=} \frac{1}{4} \mathcal{L}(z, \lambda) - \frac{1}{4} \mathcal{L}(z^+, \lambda^+). \end{aligned} \quad (28)$$

Combine the result of Equation (28) and (27), we have

$$\mathcal{L}(z^+, \lambda^+) \leq \frac{1}{1+\delta} \mathcal{L}(z, \lambda).$$

Later, a serial recurrence formula can be established:

$$\begin{aligned} \sum_{i=1}^N \|z^k - z^*\|_{B_i}^2 &\leq \sum_{i=1}^N \left(\|\lambda_i^k - \lambda_i^*\|_{B_i^{-1}}^2 + \|z^k - z^*\|_{B_i}^2 \right) \\ &= \mathcal{L}(z^k, \lambda^k) \leq \left(\frac{1}{1+\delta} \right)^k \mathcal{L}(z^0, \lambda^0). \end{aligned} \quad (29)$$

Here z^0, λ^0 denote the initial primal and dual variables of the algorithm, respectively. Here k represents the iteration index. Equation (29) shows the global linear convergence rate of Consensus ALADIN. Theorem 2 is proved. \blacksquare

Note that, by setting $B_i = \rho I$, the global convergence theory of Reduced Consensus ALADIN can also be established. To the best of our knowledge, Theorem 2 provides the first global linear convergence theory for ALADIN-type algorithms in convex problems.

B. Local Convergence Analysis

The local convergence analysis of Consensus ALADIN for nonconvex problems is similar to [11, Section 7].

Theorem 3 *For Problem (2), let f_i s be twice continuously differentiable. Let x_i s be updated as (9). Let the second order sufficient optimality condition (SOSC) be satisfied at a local minimizer z^* of (2). By initializing x_i s and z in a neighborhood of z^* , the iteration of Algorithm 2 locally converges with linear rate.*

Proof: Let the minimizers of the decoupled problems (9) be regular KKT points and lie within a neighborhood of the optimal solution z^* . Under these conditions, (30) is satisfied for all the local problems (9),

$$\frac{\rho}{\sigma} \|z - z^*\| + \frac{1}{\sigma} \|\lambda_i - \lambda_i^*\| \geq \|x_i^+ - z^*\|. \quad (30)$$

Here $\|\nabla^2 f_i(x_i^+) + \rho I\| \geq \sigma > 0$.

For Consensus ALADIN, let γ denote the upper bound of distance between the Hessian approximation B_i (12) and the real Hessian, i.e. $\|B_i - \nabla^2 f_i(x_i^+)\| \leq \gamma$. From the standard SQP theory [12, Chapter 18], we have

$$\begin{cases} N \|z^+ - z^*\| \leq \gamma \sum_{i=1}^N \|x_i^+ - z^*\|, \\ \sum_{i=1}^N \|\lambda_i^+ - \lambda_i^*\| \leq \gamma \sum_{i=1}^N \|x_i^+ - z^*\|. \end{cases} \quad (31)$$

Combine (31) with (30), the following inequality can be obtained.

$$\begin{aligned} & \left(\frac{\rho N}{\sigma} \|z^+ - z^*\| + \frac{1}{\sigma} \sum_{i=1}^N \|\lambda_i^+ - \lambda_i^*\| \right) \\ & \leq \frac{(\rho+1)\gamma}{\sigma} \left(\frac{\rho N}{\sigma} \|z - z^*\| + \frac{1}{\sigma} \sum_{i=1}^N \|\lambda_i - \lambda_i^*\| \right). \end{aligned} \quad (32)$$

As long as the initial point of x_i s are sufficiently close to z^* , $\frac{(\rho+1)\gamma}{\sigma} < 1$ is satisfied. This shows a local linear convergence of Consensus ALADIN.

This completes the proof. \blacksquare

Importantly, [21] proposed a globalization strategy for enforcing Consensus ALADIN globally converge to a local minimizer of non-convex DC.

V. NUMERICAL EXAMPLE

In this section, due to space limitations, we illustrate the numerical performance of the proposed algorithms only for non-convex DC. All algorithms are implemented using Casadi-v3.5.5 with IPOPT [22].

By modifying a non-convex *sensor allocation* problem from [11], we implement the following non-convex DC problem:

$$\begin{aligned} \min_{x_i, z, \forall i \in \mathcal{I}} \quad & \sum_{i=1}^N \left(\frac{1}{2} \left(\|x_i^\alpha - \zeta_i^\alpha\|^2 + \|x_i^\beta - \zeta_i^\beta\|^2 \right) + \right. \\ & \left. \frac{1}{2} \sum_{j=1}^{10} \left(\left(x_i^\alpha[j] - x_i^\beta[j] \right)^2 - \zeta_i^\sigma[j] \right)^2 \right) \quad (33) \\ \text{s.t.} \quad & x_i = z \mid \lambda_i, \end{aligned}$$

where $x_i = [(x_i^\alpha)^\top, (x_i^\beta)^\top]^\top$, with $x_i^\alpha, x_i^\beta \in \mathbb{R}^5$ and $N = 20$. Here, $(\cdot)[j]$ denotes the j 's component of the given vector (\cdot) . All components of the measured data $(\zeta_i^\alpha, \zeta_i^\beta, \zeta_i^\sigma)$ s are drawn from a Gaussian distribution $\mathcal{N}(0, 25)$ with proper dimension. In this setting, Problem (33) has 210 primal variables and 200 dual variables. In our implementation, the update of the local primal variables x_i s relies on CasADi. Additionally, the hyper-parameter ρ is set as 10^2 for all algorithms. Note that all the initial values of primal and dual variables are set as zeros vectors in our implementation.

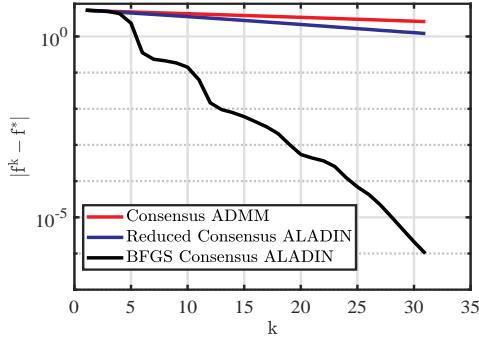


Fig. 1. Numerical comparison on a non-convex case (33).

Figure 1 shows the numerical convergence comparison among Consensus ADMM, Reduced Consensus ALADIN and BFGS Consensus ALADIN for Problem (33) as a function of iteration k . It can be observed that, even without second-order information, Reduced Consensus ALADIN still converges faster than Consensus ADMM. Additionally, BFGS Consensus ALADIN performs significantly superior performance compared to Reduced Consensus ALADIN for the given non-convex DC case.

If Problem (33) is reformulated in the manner of (7) and solved by Algorithm 1, the convergence behavior is similar to that of BFGS Consensus ALADIN. However, as explained in

Section II, Algorithm 1 is not efficient for solving (33) due to high communication and computational overhead. Therefore, the numerical performance of Algorithm 1 for (33) is then not illustrated here. Interested readers are referred to [16, Section 12] for the implementation of Algorithm 1 on Problem (7).

VI. CONCLUSION

This paper introduces a novel family of algorithms, named Consensus ALADIN, designed to efficiently solve distributed consensus optimization problems of the form (2). Building on the structure of Consensus ALADIN, we propose BFGS consensus ALADIN, a communication-and-computation-efficient variant. Furthermore, we introduce a more computationally-efficient algorithm, called Reduced Consensus ALADIN, based on BFGS Consensus ALADIN. We establish convergence theory for Consensus ALADIN and conduct numerical experiments to demonstrate its practical effectiveness.

ACKNOWLEDGEMENT

The authors wish thank Xiaohua Zhou and Shijie Zhu from ShanghaiTech University for helpful discussions.

APPENDIX

In the standard Consensus ADMM framework, there are two different cases: a) first update the dual then aggregate (update the global variable) [23], b) first aggregate then update the dual [2, Chapter 7]. In order to distinguish the Reduced Consensus ALADIN (15) from the two variant of Consensus ADMM, we use superscripts on key variables, such as λ_i^s and z , to show the difference.

A. First Update the Dual then Aggregate

By using the dual variable λ_i , the corresponding Lagrangian function of (2) can be expressed as

$$\mathcal{L}(x, z, \lambda) = \sum_{i=1}^N \left(f_i(x_i) + \lambda_i^\top (x_i - z) + \frac{\rho}{2} \|x_i - z\|^2 \right), \quad (34)$$

where ρ is a given positive penalty parameter. From (34), the main steps of updating the local primal and dual variables with Consensus ADMM [23] can be summarized as the following equation,

$$\begin{cases} x_i^+ = \arg \min_{x_i} f_i(x_i) + \left(\lambda_i^{\text{ADMM1}} \right)^\top x_i + \frac{\rho}{2} \|x_i - z^{\text{ADMM1}}\|^2, \\ \left(\lambda_i^{\text{ADMM1}} \right)^+ = \left(\lambda_i^{\text{ADMM1}} \right) + \rho \left(x_i^+ - z^{\text{ADMM1}} \right), \\ \left(z^{\text{ADMM1}} \right)^+ = \frac{1}{N} \sum_{i=1}^N \left(x_i^+ + \frac{\left(\lambda_i^{\text{ADMM1}} \right)^+}{\rho} \right). \end{cases} \quad (35)$$

From the expression of the gradients (11) in Consensus ALADIN (15), one may find

$$\left(\lambda_i^{\text{ADMM1}} \right)^+ = -g_i. \quad (36)$$

It can be noticed that the framework of Reduced Consensus ALADIN (15) is very similar to this order of (35). By setting $B_i = \rho I$, (8) boils down to

$$(z^{\text{ALADIN}})^+ = \left\{ \begin{array}{l} \arg \min_{\Delta x_i, z, \forall i \in \mathcal{I}} \sum_{i=1}^N \left(\frac{\rho}{2} \Delta x_i^\top \Delta x_i + g_i^\top \Delta x_i \right) \\ \text{s.t.} \quad \Delta x_i + x_i^+ = z \mid \lambda_i^{\text{ALADIN}} \end{array} \right\}. \quad (37)$$

It is equivalent to the same operation as that of Consensus ADMM framework (38) (if we ignore the auxiliary variables Δx_i s)

$$(z^{\text{ADMM1}})^+ = \arg \min_z \sum_{i=1}^N \left(\frac{\rho}{2} \|x_i^+ - z\|^2 - (\lambda_i^{\text{ADMM1}})^\top z \right). \quad (38)$$

In both ways of updating the global variable z , Equation (37) and (38), have the same result as Equation (14). However, the dual update in Reduced Consensus ALADIN is different from (36) which is shown as the following equation,

$$\lambda_i^{\text{ALADIN}} = \rho \left(x_i^+ - (z^{\text{ALADIN}})^+ \right) - g_i. \quad (39)$$

Importantly, in the Consensus ADMM iteration (35), $\sum_{i=1}^N (\lambda_i^{\text{ADMM1}})^* = 0$ is guaranteed only at the optimal point. On the opposite, with Reduced Consensus ALADIN (15), $\sum_{i=1}^N \lambda_i^{\text{ALADIN}} = 0$ is guaranteed in each iteration. Since the first version of Consensus ADMM can not bring the latest dual update back to each agent, the Reduced Consensus ALADIN framework can be interpreted as a more efficient way for using the consensus dual information.

B. First Aggregate then Update the Dual

As introduced in [2, Chapter 7], Consensus ADMM can be interpreted as the following equation,

$$\left\{ \begin{array}{l} x_i^+ = \arg \min_{x_i} f_i(x_i) + (\lambda_i^{\text{ADMM2}})^\top x_i + \frac{\rho}{2} \|x_i - z^{\text{ADMM2}}\|^2, \\ (z^{\text{ADMM2}})^+ = \frac{1}{N} \sum_{i=1}^N \left(x_i^+ + \frac{\lambda_i^{\text{ADMM2}}}{\rho} \right), \\ (\lambda_i^{\text{ADMM2}})^+ = (\lambda_i^{\text{ADMM2}}) + \rho \left(x_i^+ - (z^{\text{ADMM2}})^+ \right). \end{array} \right. \quad (40)$$

With (40), Consensus ADMM can also converge for convex problems with guarantees. In this form, the update of λ_i^{ADMM2} s has the same property as Reduced Consensus ALADIN that guarantees

$$\sum_{i=1}^N \lambda_i^{\text{ADMM2}} = 0 \quad (41)$$

in each iteration. In this way, the dual variables can also carry sensitivity information of the gap between the latest global variable $(z^{\text{ADMM2}})^+$ and the local variables x_i^+ . However, the second version of Consensus ADMM can not upload the latest local dual (36) back to the master because of (41). This shows that Reduced Consensus ALADIN (15) is more efficient for global variable aggregation than (40).

REFERENCES

- [1] B. Houska and Y. Jiang, "Distributed optimization and control with aladin," *Recent Advances in Model Predictive Control: Theory, Algorithms, and Applications*, pp. 135–163, 2021.
- [2] S. Boyd, N. Parikh, and E. Chu, *Distributed optimization and statistical learning via the alternating direction method of multipliers*. Now Publishers Inc, 2011.
- [3] P. Chanfreut, J. M. Maestre, D. Krishnamoorthy, and E. F. Camacho, "Aladin-based distributed model predictive control with dynamic partitioning: an application to solar parabolic trough plants," in *2023 62nd IEEE Conference on Decision and Control (CDC)*, pp. 8376–8381, IEEE, 2023.
- [4] A. Engelmann, Y. Jiang, T. Mühlpfordt, B. Houska, and T. Faulwasser, "Toward distributed OPF using ALADIN," *IEEE Transactions on Power Systems*, vol. 34, no. 1, pp. 584–594, 2019.
- [5] L. Lanza, T. Faulwasser, and K. Worthmann, "Distributed optimization for energy grids: A tutorial on admm and aladin," *arXiv preprint arXiv:2404.03946*, 2024.
- [6] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International conference on machine learning*, pp. 448–456, pmlr, 2015.
- [7] 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.
- [8] Y. Yang, X. Guan, Q.-S. Jia, L. Yu, B. Xu, and C. J. Spanos, "A survey of admm variants for distributed optimization: Problems, algorithms and features," *arXiv preprint arXiv:2208.03700*, 2022.
- [9] 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.
- [10] 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.
- [11] B. Houska, J. Frasch, and M. Diehl, "An augmented lagrangian based algorithm for distributed nonconvex optimization," *SIAM Journal on Optimization*, vol. 26, no. 2, pp. 1101–1127, 2016.
- [12] J. Nocedal and S. Wright, *Numerical optimization*. Springer Science & Business Media, New York, 2006.
- [13] X. Du, A. Engelmann, Y. Jiang, T. Faulwasser, and B. Houska, "Distributed state estimation for AC power systems using Gauss-Newton ALADIN," in *In Proceedings of the 58th IEEE Conference on Decision and Control*, pp. 1919–1924, 2019.
- [14] A. Engelmann, Y. Jiang, B. Houska, and T. Faulwasser, "Decomposition of nonconvex optimization via bi-level distributed aladin," *IEEE Transactions on Control of Network Systems*, vol. 7, no. 4, pp. 1848–1858, 2020.
- [15] Y. Jiang, J. Su., Y. Shi, and B. Houska, "Distributed optimization for massive connectivity," *IEEE Wireless Communication Letters*, vol. 9, no. 9, pp. 1412–1416, 2020.
- [16] A. Aadithya A, V. Radhakrishnan, *et al.*, "Learning (with) distributed optimization," *arXiv e-prints*, pp. arXiv–2308, 2023.
- [17] B. Houska and J. Shi, "Distributed mpc with aladin- a tutorial," in *2022 American Control Conference (ACC)*, pp. 358–363, IEEE, 2022.
- [18] Q. Ling, W. Shi, G. Wu, and A. Ribeiro, "Dlm: Decentralized linearized alternating direction method of multipliers," *IEEE Transactions on Signal Processing*, vol. 63, no. 15, pp. 4051–4064, 2015.
- [19] Y. Yang, Q.-S. Jia, Z. Xu, X. Guan, and C. J. Spanos, "Proximal admm for nonconvex and nonsmooth optimization," *Automatica*, vol. 146, p. 110551, 2022.
- [20] H. K. Khalil, *Nonlinear control*, vol. 406. Pearson New York, 2015.
- [21] X. Du, J. Wang, X. Zhou, and Y. Mao, "A bi-level globalization strategy for non-convex consensus admm and aladin," 2023.
- [22] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "CasADI – a software framework for nonlinear optimization and optimal control," *Mathematical Programming Computation*, vol. 11, no. 1, pp. 1–36, 2019.
- [23] S. Zhou and G. Y. Li, "Federated learning via inexact admm," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.