Communication Efficient Curvature Aided
Primal-dual Algorithms for Distributed
Optimization

Yichuan Li, Student Member, IEEE, Petros G. Voulgaris, Fellow, IEEE,
Dušan M. Stipanović, Senior Member, IEEE, and Nikolaos M. Freris Senior
Member, IEEE

Abstract

This paper presents a family of primal-dual algorithms for distributed convex composite problems. We consider the setting of a network of agents that cooperatively minimize a global objective function composed of a sum of local functions plus a regularizer. Through use of intermediate consensus variables, we remove the need for inner-communication loops between agents when computing curvature guided updates. A general scheme is presented which unifies the analysis for a plethora of computing choices, including gradient descent, Newton updates, and BFGS updates. Our analysis establishes sublinear convergence to the optimal solution under convex objective functions with Lipschitz continuous gradient, as well as linear convergence rate when the local functions are assumed to be strongly convex. Moreover, we explicitly characterize the acceleration due to curvature information. Last but not least, we present an asynchronous implementation for the proposed algorithms, which removes the need for a central clock, with linear convergence rate established in expectation under strongly convex objectives. We ascertain the effectiveness of the proposed methods with numerical experiments on benchmark datasets.

Yichuan Li is with the Coordinated Science Laboratory and the Department of Mechanical Science and Engineering, University of Illinois Urbana-Champaign, IL 61820 USA (email: yli129@illinois.edu).

Petros G. Voulgaris is with the Department of Mechanical Engineering, University of Nevada, Reno, NV 89557, USA (email: pvoulgaris@unr.edu).

Dušan M. Stipanović is with the Coordinated Science Laboratory and the Department of Industrial and Enterprise Systems Engineering, University of Illinois Urbana-Champaign, IL 61820 USA (email: dusan@illinois.edu).

Nikolaos M. Freris is with the School of Computer Science, University of Science and Technology of China, Hefei, Anhui, 230027, China (email: nfr@ustc.edu.cn).
Index Terms

Asynchronous algorithms, distributed optimization, primal-dual algorithms, network analysis and control.

I. INTRODUCTION

The proliferation of mobile devices with computation and communication capabilities has fueled the surge of applications of distributed optimization in various fields. Examples include distributed control [1], wireless sensor networks [2], power grid management [3], and large-scale machine learning [4]. A canonical problem in distributed optimization assumes a network of agents collaboratively optimizing a global objective function through message passing with immediate neighbors. In specific, we consider the following optimization problem:

\[
\minimize_{\hat{x} \in \mathbb{R}^d} \sum_{i=1}^{m} f_i(\hat{x}) + g(\hat{x}),
\]

where each \( f_i(\cdot) : \mathbb{R}^d \to \mathbb{R} \) is a convex and smooth function accessible only by agent \( i \) while \( g(\cdot) : \mathbb{R}^d \to \mathbb{R} \) is a convex (possibly nonsmooth) regularizer function. The inclusion of the regularizer is multi-faceted, e.g., it serves for promoting desired structures in the decision vector, such as sparsity in the controller design using the \( \ell_1 \)-norm [5], preventing overfitting in machine learning problems using the squared \( \ell_2 \)-norm [6], and enforcing constraints using indicator functions of convex sets.

First order methods [7]–[13] using (sub)-gradient information constitute popular choices for solving (1) due to their economical computational costs and simple implementation. In first order methods, agents compute updates by using local gradients combined with averaged information from their neighbors. For the case with no regularizer (\( g(\cdot) \equiv 0 \)), [8], [10], [12] exploit the history of gradient and iterate values to achieve linear convergence rate for strongly convex objectives. [13] provides a unified framework for designing various first order schemes for the general problem (1). When nonsmooth regularizers are present, existing work almost exclusively applies proximal gradient type of updates: each agent first performs gradient descent on the smooth part of the objective function and then invokes the proximal operator associated with the nonsmooth regularizer \( g(\cdot) \). Nonetheless, using only first order information suffers from slow convergence speed and thus requires large number of total iterations to reach a prescribed accuracy. This constitutes a key limitation for first order methods, that is most pronounced in
applications where high accuracy solutions are pursued in few rounds of iterations, for example due to high communication costs.

A natural option for accelerating the convergence is to use second order information for local updates. Most second order methods [14]–[17] for solving (1) focus on cases where the objective function is smooth, i.e., \( g(\cdot) = 0 \). One reason is that even when proximal gradient steps are efficiently computable, proximal Newton steps require significantly more computational resources due to Hessian scaling in the evaluation of the proximal operator. Another challenge in designing second order methods lies in constructing distributively computable Newton updates. Computing curvature guided updates requires solving a linear system that, in general, involves global information, whence a direct application of Newton method is not feasible. Moreover, the standard Newton method requires backtracking line search to select appropriate step sizes for ensuring global convergence [18]. Such operations incur heavy communication burdens in the form of collecting all local objective function values in the network; this necessitates extensive message passing between agents or the presence of a centralized coordinator. Authors in [15] propose to use matrix splitting techniques in the dual problem, so that the Hessian inverse admits a distributively computable Taylor expansion. By truncating the Taylor series to \( K \) terms, agents may compute local updates with an additional \( K \) rounds of communication loops with their neighbors. With \( g(\cdot) = 0 \), [16] and [17] use similar matrix splitting techniques to solve a penalized version of (1) where the former presents a synchronous scheme and the latter extends it to asynchronous settings. We note that [16] and [17] are effectively solving a different problem (penalized version) compared to (1) and therefore do not converge to the exact solution.

Another popular line of algorithmic design for solving (1) is based on primal-dual methods, such as the Generalized Method of Multipliers, the Augmented Lagrangian Method, and the Alternating Direction Method of Multipliers (ADMM) [19]–[21]. In the setting of distributed primal-dual algorithms [22]–[27], agents solve a sub-optimization problem at each iteration, which often involves multiple inner loops and thus induces heavy computation burden. Several approximation schemes [28]–[33] were proposed to replace the exact minimization step with one or multiple update steps using approximated models of the augmented Lagrangian. It has been shown that by appropriately choosing the mixing matrices and the augmented Lagrangian model, primal-dual algorithms can recover several accelerated primal-only algorithms using gradient and iterate tracking techniques [34]. Further acceleration can be achieved by resorting to Newton or quasi-Newton primal updates [31]–[33]. However, all of them are synchronous algorithms
considering the smooth problem \((g(\cdot) = 0)\) and \([32], [33]\) require multiple inner communication loops at each iteration of the algorithm. In such scenarios, despite improving the convergence speed, it is not clear whether the overall communication costs can be reduced due to the additional communication rounds per iteration. In emerging applications such as multi-agent Cyberphysical Systems \([35]\) and Federated Learning \([36]\), high responsiveness and reducing communication costs are of primordial importance. This motivates the development of methods with accelerated convergence as well as with guaranteed low communication costs, which is the focus of this paper.

**Contributions:**

- We introduce a framework for designing distributed primal-dual algorithms for (1) with nonsmooth regularization function. Through the use of intermediate consensus variables, we decouple the primal subproblem pertaining to an agent from those of its neighbors. As a result, we obtain a block-diagonal Hessian that allows to incorporate curvature information in local updates *without additional communication*. This is in contradistinction with the state-of-the-art, where multiple communication inner loops are required to compute (quasi) Newton updates.

- Using this framework, we propose DistRibuted cUrvature aided prImal Dual algorithms (DRUID), a family of algorithms that offer flexible choices of updating schemes, including gradient, Newton, and BFGS type of updates. Furthermore, we present a unified analysis framework for this class of algorithms, which not only establishes \(O(\frac{1}{T})\) convergence rate to the optimum under convex objectives, but also theoretically reveals the discrepancies among them. When strong convexity is further assumed, we establish linear convergence rate for this class of algorithms, and once again quantify the acceleration.

- We devise an asynchronous extension for this class of algorithms, and establish linear convergence rate in expectation, under strong convexity. This setting removes the need for a central clock in the network, and further allows for arbitrary number of agents being active at each iteration. We demonstrate the merits of the proposed framework through case studies of machine learning problems using real-life datasets.

**Notation:** We represent column vectors \(x \in \mathbb{R}^d\) with lower case letters, matrices \(A \in \mathbb{R}^{n \times m}\) with capital letters, and matrix transpose as \(A^\top\). We also use \([A, B]\) and \([A; B]\) to respectively denote row and column stacking (for matrices with equal numbers of rows or columns, respectively). Superscript denotes the sequence index while subscript denotes the vector component.
For example, $x^t_i$ represents the vector component held by agent $i$ at iteration $t$. Moreover, $[A]_{ij}$ denotes the $ij$-th entry of matrix $A$. If a norm specification is not provided, $\|x\|$ and $\|A\|$ represent the vector Euclidean norm and the induced matrix norm, respectively. For a positive definite matrix $P \succ 0$, we define $\|x\|_P := \sqrt{x^\top P x}$. The set $\{1, \ldots, m\}$ is abbreviated as $[m]$ and the proximal mapping associated with a function $g(\cdot) : \mathbb{R}^d \to \mathbb{R}$ is defined as $\text{prox}_{g/\mu}(v) := \arg\min_{\theta \in \mathbb{R}^d} \{g(\theta) + \frac{\mu}{2} \|\theta - v\|^2\}$. We further denote the identity matrix of dimension $d$ as $I_d$ and the Kronecker product between two matrices of arbitrary dimension $A, B$ as $A \otimes B$.

II. Preliminaries

In this section, we begin with reformulating problem (1) to the consensus setting that is used for our development in Section II-A. We provide some background on using ADMM to solve the reformulated problem in Section II-B and an introduction to quasi-Newton methods in Section II-C.

A. Problem formulation

We capture the network topology by an undirected graph $G = \{V, E\}$ where $V := [m]$ denotes the vertex set and the edge set $E \subseteq V \times V$ contains the pair $(i, j)$ if and only if agent $i$ can communicate with agent $j$. We do not consider self loops, i.e., $(i, i) \notin E$ for any $i \in [m]$. For notational convenience, we enumerate the edge set (arbitrary order) and use $E_k$ to denote the $k$-th edge, $k \in [n]$, where $n := |E|$ is the number of edges. Moreover, the set of neighbors of agent $i$ is defined as $N_i := \{j \in V : (i, j) \in E\}$. Using the above definitions, we reformulate problem (1) to the following consensus formulation, by introducing local decision variables $x_i$ at corresponding agent $i$, as well as edge variables $z_{ij}$ for $(i, j) \in E$. The consensus formulation is given by:

$$\begin{align*}
\text{minimize} & \quad \sum_{i=1}^m f_i(x_i) + g(\theta), \\
\text{s.t.} & \quad x_i = z_{ij} = x_j, \ \forall \ i \in [m] \ \text{and} \ j \in N_i, \\
& \quad x_l = \theta, \ \text{for some} \ l \in [m].
\end{align*}$$

(2)

Note that we have also introduced $\theta$ to separate the argument of the smooth and nonsmooth functions and only enforce the equality constraint for $\theta$ at the $l$-th agent as $x_l = \theta$, where $l$ can be arbitrarily selected. We emphasize that this agent is not a central coordinator, but rather the agent whose local updates factor in the nonsmooth regularizer. This is without loss of generality.
and induces minimal computational overhead from evaluating proximal mappings. Assuming $G$ is connected, it is easy to check that (2) is equivalent to (1) since their optima coincide, i.e., $\hat{x}^* = x_i^* = z_{ij}^* = \theta^*, \forall i \in [m]$ and $j \in \mathcal{N}_i$. This is achieved by satisfying the consensus constraints in (2). We note that consensus can be enforced by simply letting $x_i = x_j$, i.e., without intermediate consensus variables $\{z_{ij}\}$. However, the introduction of intermediate variables is key to our design: the purpose of $\{z_{ij}\}$ is to decouple $x_i$ from its neighbors so that we achieve a block-diagonal Hessian for the augmented Lagrangian. A block-diagonal Hessian allows agents to compute the (quasi) Newton steps without additional communication with their neighbors.

We provide further discussion on this choice in Section III.

We proceed to define the source and destination matrices $\hat{A}_s, \hat{A}_d \in \mathbb{R}^{n \times m}$. Each row of $\hat{A}_s$ and $\hat{A}_d$ corresponds to an edge $E_k$ in the graph, $k \in [n]$; $[\hat{A}_s]_{ki} = [\hat{A}_d]_{kj} = 1$ if and only if $E_k = (i, j)$, and 0 otherwise. Problem (2) can then be compactly expressed using the concatenated column vectors $x := [x_1^T, \ldots, x_m]^T$, $z := [z_1^T, \ldots, z_u]^T$ as:

$$\min_{x \in \mathbb{R}^{md}, \theta \in \mathbb{R}^d, z \in \mathbb{R}^{nd}} F(x) + g(\theta),$$

$$Ax = \begin{bmatrix} \hat{A}_s \otimes I_d \\ \hat{A}_d \otimes I_d \end{bmatrix} x = \begin{bmatrix} I_{nd} \\ I_{nd} \end{bmatrix} z = Bz,$$

$$S^T x = \theta,$$

where $F(x) := \sum_{i=1}^m f_i(x_i)$ and matrices $A$ and $B$ are obtained by stacking the matrices as shown in (3). We further define $S := (s_l \otimes I_d) \in \mathbb{R}^{md \times d}$ where $s_l \in \mathbb{R}^m$ is an all-zero vector except for the $l$-th entry being one. In other words, the $S^T$ matrix serves to select the $l$-th component of $x$ held by the agent $l$, i.e., $S^T x = x_l$. We proceed to present some identities that associate source and destination matrices to the incidence and Laplacian matrices corresponding to the graph topology in the following.

$$\hat{E}_s = \hat{A}_s - \hat{A}_d, \quad \hat{E}_u = \hat{A}_s + \hat{A}_d,$$  

(4a)

$$\hat{L}_s = \hat{E}_s^T \hat{E}_s, \quad \hat{L}_u = \hat{E}_u^T \hat{E}_u,$$  

(4b)

$$\hat{D} \equiv \frac{1}{2} (\hat{L}_s + \hat{L}_u) = \hat{A}_s^T \hat{A}_s + \hat{A}_d^T \hat{A}_d,$$  

(4c)

where $\hat{E}_s, \hat{E}_u \in \mathbb{R}^{n \times m}$ are signed and unsigned graph incidence matrices and $\hat{L}_s, \hat{L}_u \in \mathbb{R}^{m \times m}$ are signed and unsigned graph Laplacian matrices respectively. The diagonal matrix $\hat{D} \in \mathbb{R}^{m \times m}$ denotes the graph degree matrix with entries $D_{ii} = |\mathcal{N}_i|$. We further introduce the block extensions to the dimension $d$, that is $E_s := \hat{E}_s \otimes I_d$ and similarly for $E_u, L_s, L_u, \text{ and } D$. 


B. Background on ADMM

We begin by defining the augmented Lagrangian for problem (3):

\[
\mathcal{L}(x, \theta, z; y, \lambda) := F(x) + g(\theta) + y^\top (Ax - Bz) + \lambda^\top (S^\top x - \theta) + \frac{\mu_z}{2} \|Ax - Bz\|^2 + \frac{\mu_\theta}{2} \|S^\top x - \theta\|^2,
\]

(5)

where \(y \in \mathbb{R}^{2nd}, \lambda \in \mathbb{R}^d\) are Lagrange multipliers associated with the constraints \(Ax = Bz\) and \(S^\top x = \theta\), respectively. Note that since penalty coefficients of the quadratic terms are closely related to dual step sizes, we have separated them into \(\mu_z\) and \(\mu_\theta\) to offer broader choices of selection. ADMM solves (3), equivalently (2) and (1), by sequentially minimizing the augmented Lagrangian (5) over each of the primal variables \((x, \theta, z)\), and then performs gradient ascent on the dual variables \((y, \lambda)\):

\[
\begin{align*}
  x^{t+1} &= \arg\min_x \mathcal{L}(x; \theta^t, z^t; y^t, \lambda^t), \\
  \theta^{t+1} &= \arg\min_{\theta} \mathcal{L}(x^{t+1}, \theta, z^t; y^t, \lambda^t), \\
  z^{t+1} &= \arg\min_z \mathcal{L}(x^{t+1}, \theta^{t+1}, z; y^t, \lambda^t), \\
  y^{t+1} &= y^t + \mu_z (Ax^{t+1} - Bz^{t+1}), \\
  \lambda^{t+1} &= \lambda^t + \mu_\theta (S^\top x^{t+1} - \theta^{t+1}).
\end{align*}
\]

(6a) (6b) (6c) (6d) (6e)

The above iterations fall into the category of 3-block ADMM which is not guaranteed to converge for arbitrary \(\mu_z, \mu_\theta > 0\) [37]. Step (6a) requires a solution to a sub-optimization problem which often involves multiple inner-loops for general objective functions, and therefore becomes the most expensive step in ADMM. Executing step (6b) bears the complexity of computing the proximal mapping of the regularization function \(g(\cdot)\). For commonly used \(g(\cdot)\), such as the \(\ell_1\)-norm, a closed form solution exists. Step (6c) results from the use of consensus variables \(\{z_{ij}\}\), and it does not require explicit evaluation as we demonstrate in the Section III.

C. Introduction to quasi-Newton methods

Quasi-Newton methods [18] constitute a class of methods that aim to accelerate convergence using curvature information of the objective function without solving a linear system as in the
Newton method. Specifically, the update direction $u^t \in \mathbb{R}^d$ in quasi-Newton methods is given by:

$$u^t = -(H^t)^{-1}\nabla f(x^t),$$

where $(H^t)^{-1} \succ 0$ is some matrix (the inverse is just notation for ease of exposition, and no inversion is needed) that approximates the Hessian inverse. One of the main advantages of quasi-Newton methods lies in the fact that $(H^t)^{-1}$ is explicitly available so computing $u^t$ amounts to performing matrix multiplication at the cost of $O(d^2)$ for general problems, as compared to solving a linear system with computational costs $O(d^3)$ in Newton method. Many schemes exist for estimating $(H^t)^{-1}$ and in subsequent discussions, we focus on the one proposed by Broyden, Fletcher, Goldfarb, and Shanno (BFGS) [38]–[41], which is proven to be the most effective in terms of acceleration and self-correcting capabilities [18].

We define the consecutive iterate and gradient differences as:

$$s^t = x^{t+1} - x^t, \text{ and } q^t = \nabla f(x^{t+1}) - \nabla f(x^t). \quad (7)$$

BFGS requires the updated Hessian inverse approximation $(H^{t+1})^{-1}$ to satisfy the following secant condition:

$$(H^{t+1})^{-1} q^t = s^t, \quad (8)$$

which is motivated by the fact that the exact Hessian inverse satisfies [8] as $x^{t+1}$ tends to $x^t$. However, the secant condition alone is not enough to specify $(H^{t+1})^{-1}$. BFGS proposes to select $(H^{t+1})^{-1}$ by further requiring the updated estimate to be close to the previous one in the following sense:

$$\min_{H^{-1}} \|H^{-1} - (H^t)^{-1}\|_W \quad (9)$$

s.t. $H^{-1} = (H^{-1})^\top$, $H^{-1} q^t = s^t$,

where $\|A\|_W := \left\|W^{\frac{1}{2}} A W^{\frac{1}{2}}\right\|_F$ denotes the weighted Frobenius norm with $W$ being the average Hessian [18]. Problem (9) admits a closed form solution which gives rise to the update formula for $(H^{t+1})^{-1}$ in BFGS:

$$(H^{t+1})^{-1} = (I - \rho^t s^t(q^t)^\top)(H^t)^{-1} (I - \rho^t q^t(s^t)^\top) + \rho^t s^t(s^t)^\top, \quad (10)$$

where $\rho^t = 1/(q^t)^\top s^t$. By using only gradient information as in first order methods, BFGS iteratively constructs a Hessian estimate of the objective function as in (10) that is accurate
enough to achieve superlinear convergence rate. However, direct application of BFGS does not admit a distributed implementation as can be seen in the formula (10) where computing $s^t(q^t)^\top$ involves global operations and message passing among agents. In the following section, we introduce BFGS updates in the framework of primal-dual algorithms that are not only distributedly computable, but also retain the same communication costs as their first order counterparts.

### III. Algorithmic Development

An approximated augmented Lagrangian $\hat{L}(.)$ can be obtained using second order expansion as follows:

$$
\hat{L}(x, \theta^t, z^t; y^t, \lambda^t) = L^t(x^t) + (x - x^t)^\top \nabla_x L^t + \frac{1}{2} \|x - x^t\|_{H^t}^2,
$$

where we abbreviated the $L(x^t, \theta^t, z^t; y^t, \lambda^t)$ as $L^t(x^t)$, and the selection of $H^t$ is a means for designing a range of methods as will be subsequently elaborated in this section. We obtain a closed form solution when minimizing $\hat{L}(\cdot)$ over $x$ and replace step (6a) with the following one-step update:

$$
x^{t+1} = x^t - (H^t)^{-1} \nabla_x L(x^t, \theta^t, z^t; y^t, \lambda^t).
$$

By completion of squares, step (6b) admits an analytical expression through the proximal mapping:

$$
\theta^{t+1} = \text{prox}_{y/\mu} (S^\top x^{t+1} + \frac{1}{\mu} \lambda^t).
$$

Moreover, since the augmented Lagrangian is quadratic with respect to $z$, it follows that $z^{t+1}$ of step (6c) can be computed by solving the following linear system of equations:

$$
B^\top y^t + \mu z B^\top (Ax^{t+1} - Bz^{t+1}) = 0.
$$

Dual variables are updated in verbatim as in steps (6d) and (6e). We note that dual updates can be performed in parallel once primal updates are completed. Before we explicate the choice for $H^t$, we present a lemma that allows for efficient implementation of (11)–(13) and (6d)–(6e) under appropriate initialization.

**Lemma 1.** Recall the identities in (4a)–(4c) and the definitions thereafter. We express the dual variable as $y^t = [\alpha^t; \beta^t]$, $\alpha^t, \beta^t \in \mathbb{R}^{nd}$. If $y^0$ and $z^0$ are initialized so that $\alpha^0 + \beta^0 = 0$ and...
$z^0 = \frac{1}{2}E_u x^0$, then $\alpha^t + \beta^t = 0$ and $z^t = \frac{1}{2}E_u x^t$ for all $t \geq 0$. Moreover, defining $\phi^t = E_s^\top \alpha^t$, we equivalently express the updates (11)–(13), (6d)–(6e) as:

\begin{align*}
  x^{t+1} &= x^t - (H^t)^{-1} \left[ \nabla F(x^t) + \phi^t + S \lambda^t + \frac{\mu_z}{2} L_s x^t + \mu_\theta S (S^\top x^t - \theta^t) \right], \\
  \theta^{t+1} &= \text{prox}_{\gamma/\mu_\theta} (S^\top x^{t+1} + \frac{1}{\mu_\theta} \lambda^t), \\
  \phi^{t+1} &= \phi^t + \frac{\mu_z}{2} L_s x^{t+1}, \\
  \lambda^{t+1} &= \lambda^t + \mu_\theta (S^\top x^{t+1} - \theta^{t+1}).
\end{align*}

(14a)–(14d)

**Proof:** See Appendix A.

**Remark 1.** The above proof was similarly derived in [22] and [25] for the case $g(\cdot) = 0$ and the suboptimization problem (6a) was solved exactly. Note that to satisfy the requirement of Lemma 1, zero initialization for all variables suffices. Lemma 1 establishes that updates (14a)–(14d) are equivalent to (11)–(13) and (6d)–(6e) under appropriate initialization. This has a twofold implication: (i) we have achieved transforming a 3-block ADMM to a 2-block ADMM, which allows for a broader range of algorithm parameters $\mu_z, \mu_\theta$ that guarantee convergence; (ii) it is not required to explicitly store and update $z^t$ since it evolves on a linear manifold parameterized by $x^t$, i.e., $z = \frac{1}{2}E_u x$. Besides, only half of $y^t$ needs to be stored since $y^t = [\alpha^t; -\alpha^t]$. This further reduces associated storage and communication costs.

Using the equivalent while more efficient updates (14a)–(14d), we proceed to develop a family of algorithms by explicating different choices of $J^t$ used in the construction of the approximate Hessian of the augmented Lagrangian as:

$$H^t = J^t + \mu_z D + \mu_\theta SS^\top + \epsilon I_{md},$$

(15)

where we have introduced $\epsilon > 0$ to provide additional robustness for our approximation. Notice that $\mu_z D + \mu_\theta SS^\top + \epsilon I_{md}$ is a diagonal matrix, whence $H^t$ is block-diagonal when $J^t$ is. When $H^t$ is *block-diagonal*, each component of the update direction, $(H^t)^{-1} \nabla_x L^t$ in (11), can be computed individually by agents. More precisely, agent $i$ computes the update $u^t_i$ by solving the following linear system:

$$H^t_{ii} u^t_i = \nabla_x L^t_i.$$  

(16)

Therefore, once the right hand side of (16) is obtained by $i$-th agent, no additional communication is needed to solve for $u^t_i$. This is made possible by using the intermediate consensus variables $\{z_{ij}\}$ which decouple $x_i$ from $x_j$. 

Remark 2. If consensus constraints are enforced directly as $x_i = x_j$, e.g., $E_s x = 0$, then the approximated Hessian will not be block-diagonal, but rather have a structure compatible with the graph:

$$H^t = \nabla^2 F(x^t) + \mu_z L_s + \mu_\theta S S^\top + \epsilon I_{md}.$$  

Due to the presence of the signed graph Laplacian matrix $L_s$, the $ij$-th block will be nonzero if $(i, j) \in \mathcal{E}$. In such scenarios, computing $u^t_i$ either requires the presence of a fusion center which gathers all $\nabla_x \mathcal{L}^t_i$ for centralized processing, or a distributed implementation can be pursued by computing inexact (quasi) Newton-updates by truncating the Taylor expansion of the Hessian inverse with $K$ terms [16], [17], [32], [33], [42]–[44]. However, the truncation approach incurs $K$ additional communication rounds among agents and their neighbors, per iteration. This not only induces large communication overhead, but also demands stringent synchronization among agents. In contrast, all our proposed methods feature minimal communication complexity, i.e., only one round of distributed communication per iteration. Different choices of $J^t$ in (15) affect the local computational cost and convergence rate as we elaborate next.

A. Gradient updates

By choosing $J^t_{\text{Gradient}} \equiv 0$, it follows that $H^t$ is diagonal. Therefore, (14a) is equivalent to performing diagonally preconditioned gradient descent on the augmented Lagrangian, where step sizes are controlled by setting $\epsilon$. We note that the proposed algorithm recovers Decentralized Linearized ADMM (DLM) [29] with $g(\cdot) = 0$ as a special case of (3). Specifically, agent $i$ computes $u^t_i$ from (16) as:

$$u^t_i = (\mu_z |\mathcal{N}_i| + \delta_{il} \mu_\theta + \epsilon)^{-1} \nabla_x \mathcal{L}^t_i,$$  

where $\delta_{il} = 1$ if $i = l$ and 0 otherwise. The above shows that the step size of the gradient descent at agent $i$ is related to the number of its neighbors and can be adjusted by tuning $\epsilon$. In other words, agents who are densely connected, i.e., having a large $|\mathcal{N}_i|$, have a larger step sizes. Computing updates for agents using (17) involves $O(d)$ computational costs, and we proceed to specify how curvature information is incorporated with nonzero $J^t$ in the following.
B. Newton updates

By setting $J_{Newton}^t = \nabla^2 F(x^t)$ in (15), we obtain the Hessian of the augmented Lagrangian plus $\epsilon I_{md}$:

$$H_{Newton}^t = \nabla^2 F(x^t) + \mu_z D + \mu_\theta SS^\top + \epsilon I_{md}. \quad (18)$$

We note that since $F(x^t) = \sum_{i=1}^m f_i(x^t_i)$, $\nabla^2 F(x^t)$ is a block diagonal matrix with the $i$-th block being $\nabla^2 f_i(x^t_i)$. As discussed previously, this induces a block diagonal $H^t$ and the update direction $u_i^t$ can be obtained by solving (16) by each agent at the cost of $O(d^3)$ for general objective functions, without additional communication among agents.

C. Quasi-Newton updates

In this section, we introduce a distributedly implementable BFGS scheme that harnesses curvature information without inner communication loops. Some insights can be gained by investigating the target Hessian of the augmented Lagrangian in (18). We note that the only time varying part of $H_{Newton}^t$ is $\nabla^2 F(x^t)$, while the remaining part is constant (the graph structure is assumed time-invariant in this paper). In [33], authors propose to estimate $\nabla^2 F(x^t)$ using BFGS formula with each nodes’ local information and then compute the $K$-th order Taylor expansion of $(H^t)^{-1}$. For a distributed implementation, $K$ additional communication rounds are needed due to direct coupling between $x_i$ and $x_j$ in [33]. We note that such schemes not only incur higher communication costs per round, but also induce $O(d^3)$ computational costs since linear systems have to be solved by agents.

In contrast, we exploit the block-diagonal structure of the Hessian (18), and propose the following scheme for approximating $(H^t)^{-1}$ using no additional communication (i.e., by means of local computation with information already available at the agents). In specific, each agent $i$ constructs the Hessian inverse model directly using the pairs $\{q_i^t, s_i^t\}_{i=1}^m$ defined as:

$$q_i^t := \nabla f_i(x_i^{t+1}) - \nabla f_i(x_i^t) + (\mu_z |N_i| + \delta_{ii} \mu_\theta + \epsilon) s_i^t,$$

$$s_i^t := x_i^{t+1} - x_i^t. \quad (19)$$

In other words, instead of approximating the Hessian inverse of the local objective $(\nabla^2 f_i(x_i^t))^{-1}$, we are directly constructing a model for $(\nabla^2 \mathcal{L}_i^t)^{-1}$. The $i$-th block of the approximated Hessian inverse $(H_{ii}^t)^{-1}$ can be recursively updated using (10) for the $\{q, s\}$ pairs defined in (19). We emphasize that it is not needed to explicitly form $H_{ii}^t$ and solve for the update direction as in
Instead, computing $u^t_i$ is tantamount to performing matrix multiplication $(H^t_i)_{ii}^{-1} \nabla_x L^t_i$. In summary, the proposed algorithm is advantageous compared to existing methods over the following aspects: (i) no additional communication loops are needed after each gradient evaluation and (ii) the computation costs for each agent is reduced from $O(d^3)$ to $O(d^2)$. For the sake of comparison with the gradient and Newton updates, we define:

$$J^t_{\text{BFGS}} := H^t_{\text{BFGS}} - \mu_x D - \mu_\theta S S^T - \epsilon I_{md},$$

where $H^t_{\text{BFGS}}$ is obtained by the BFGS formula with $\{q^t_i, s^t_i\}_{i=1}^m$ pairs defined in (19). We proceed to describe the distributed implementation of the proposed algorithms.

### D. Distributed Implementation

The proposed algorithms admit the exact same distributed implementation with variable computing choices corresponding to the selection of $J^t$ in (15), so as to incorporate curvature information or not. The unified description is detailed in Algorithm 1. We let the $i$-th agent hold $(x^t_i, \phi^t_i)$ while the $l$-th agent additionally holds the pair $(\theta^t, \lambda^t)$ pertaining to the nonsmooth regularization function $g(\cdot)$. The gradient of the augmented Lagrangian pertaining to agent $i$, $\nabla_x L^t_i$, is expressed as:

$$h^t_i = \nabla f_i(x^t_i) + \phi^t_i + \frac{\mu_x}{2} \sum_{j \in N_i} (x^t_i - x^t_j) + \delta_{il} \mu_\theta (x^t_i - \theta^t) + \frac{1}{\mu_\theta} \lambda^t.$$  

(21)

At the beginning of each round, each agent estimates its local curvature as in steps 3-4. For the BFGS computing scheme, no computation is required at these steps. Agents then carry primal updates by first computing $h^t_i$ expressed in (21), regardless of the choice of computing scheme. We emphasize that $h^t_i$ can be computed without communication, since each agent $i$ already has access to the variables of its neighbors, $\{x^t_j | j \in N_i\}$, from the previous round. It thus computes $h^t_i$ locally in step 5. If the $i$-th agent opts to use gradient or Newton updates, $u^t_i$ is obtained by solving the linear system (for gradient descent $u^t_i$ can be trivially solved since $H^t_{ii}$ is a constant scalar). For the BFGS scheme, $u^t_i$ is computed by performing matrix-vector multiplication $(H^t_{ii})^{-1} h^t_i$. Once $u^t_i$ is obtained, agents update their $x^{t+1}_i$ in step 7. We note that the only communication round occurs at step 8 where agents broadcast $x^{t+1}_i$ to their neighbors. Dual updates are executed in step 9. We require agents to store $\{x^{t+1}_j, j \in N_i\}$, to execute step 5 in the next iteration. In addition to the primal-dual variables $(x_i, \phi_i)$, the $l$-th agent further holds $(\theta, \lambda)$ associated with the regularization term $g(\cdot)$, which are updated in steps 11-12. Finally, if
Algorithm 1 Synchronous DRUID

Initialization: zero initialization for all variables.

1: for \( t = 0, 1, \ldots \) do
2:    for \( i \in [m] \) do
3:        Compute the local curvature \( H_{ii}^t \):
4:            \( J_{ii}^t = \begin{cases} 0 & \text{Gradient updates} \\ \nabla^2 f_i(x_i^t) & \text{Newton updates} \end{cases} \)
5:            \( (H^t)_{ii} \leftarrow J_{ii}^t + (\mu_z |N_i| + \delta_{ii} \mu_\theta + \epsilon) I_d \)
6:        Primal update:
7:            compute \( h_i^t \) as in (21)
8:            \( u_i^t = (H_{ii}^t)^{-1} h_i^t \) BFGS updates
9:            \( x_i^{t+1} = x_i^t - u_i^t \)
10:       Communication:
11:          Broadcast \( x_i^{t+1} \) to neighbors and retrieve \( x_j^{t+1} \)
12:       Dual update:
13:          \( \phi_i^{t+1} = \phi_i^t + \frac{\mu_\theta}{2} \sum_{j \in N_i} (x_i^{t+1} - x_j^{t+1}) \)
14:          Updates pertaining to the regularization function:
15:           if \( i = l \) then
16:               \( \theta^{t+1} = \text{prox}_{\alpha/\mu_\theta}(x_i^{t+1} + \frac{1}{\mu_\theta} \lambda^t) \)
17:               \( \lambda^{t+1} = \lambda^t + \mu_\theta (x_i^{t+1} - \theta^{t+1}) \)
18:           end if
19:       Curvature estimation update (BFGS only):
20:          Update \( (H_{ii}^{t+1})^{-1} \) using \( \{q_i^t, s_i^t\} \) in (19) and the formula in (10)
21: end for
22: end for

BFGS is opted as the updating scheme, agents update their local curvature estimation \( (H_{ii}^{t+1})^{-1} \) in step 14.
IV. ASYNCHRONOUS IMPLEMENTATION

The proposed algorithms in the previous section fall into the category of synchronous algorithms, where at each iteration, all agents communicate with their neighbors and participate in computing in a coordinated and deterministic fashion. Such settings are appropriate when abundant communication bandwidth is available and the network is homogeneous in the sense that agents are able to finish local computations in adjacent time windows. In heterogeneous networks, where agents have different hardware conditions and different volumes of data, the progress of synchronous algorithms is limited by the slowest agent in the network at each iteration (also known as the straggler problem). Moreover, the requirement of a central coordinator becomes less practical when the size of the network grows and the availability of agents becomes unpredictable.

Asynchronous algorithms [20] remove the need for a central clock by letting a subset of agents update in a randomized fashion at each iteration. Asynchronous methods can be further classified into totally asynchronous algorithms and partially asynchronous. In the former setting, agents are able to tolerate arbitrarily large delays between updates while in the latter, a maximum delay constraint is imposed to guarantee convergence. In this section, we extend DRUID to the \textit{totally asynchronous} setting that further broadens its applicability.

Recall the synchronous updates defined in (14). With any choice of computing scheme (gradient descent, Newton, or BFGS), we compactly express the synchronous algorithms by defining the operator $T: \mathbb{R}^{(2m+2)d} \to \mathbb{R}^{(2m+2)d}$ as follows:

$$v^{t+1} = Tv^t,$$

where $v \in \mathbb{R}^{(2m+2)d}$ is a concatenation of $[x; \phi; \theta; \lambda]$, and the operator $T$ maps $[x^t; \phi^t; \theta^t; \lambda^t]$ to $[x^{t+1}; \phi^{t+1}; \theta^{t+1}; \lambda^{t+1}]$ according to (14). We proceed to define the following activation matrix:

$$\Omega^t := \begin{bmatrix} X^t & 0 & 0 & 0 \\ 0 & X^t & 0 & 0 \\ 0 & 0 & X_{ii}^t & 0 \\ 0 & 0 & 0 & X_{ii}^t \end{bmatrix},$$

where $X^t \in \mathbb{R}^{md \times md}$ is a diagonal random matrix with sub-blocks $X_{ii}^t \in \mathbb{R}^{d \times d}$, $i \in [m]$, being random sub-matrices taking values as the identity matrix $I_d$ or a zero matrix. Using the definition (22) and (23), the proposed asynchronous algorithms are expressed as:

$$v^{t+1} = v^t + \Omega^{t+1}(Tv^t - v^t).$$
The above construction corresponds to activating agents, i.e., the \( i \)-th agent only updates the corresponding pair \((x^t_i, \phi^t_i)\) (additionally \((\theta^t_i, \lambda^t_i)\) if \( i = l \)) if and only if \( X^t_{ii} = I_d \). To implement the asynchronous algorithms, we equip each agent with a buffer so that even if agents are not active, they can still receive information from their neighbors. We let each agent hold a Poisson clock, which ticks according to a Poisson process, independent of each other. We assume that clocks tick at a slower time scale than agents’ computing processes, and only one clock ticks at each iteration. These assumptions are standard in asynchronous algorithms, see also in [17], [45]–[47]. In this setting, only the active agent performs updates at each iteration, i.e., step 2 of Algorithm 1 is replaced by “for \( i \) such that \( X^t_{ii} = I_d \)”. When the \( i \)-th agent becomes active, it executes steps 3–4 using only local information and then retrieves the most recent \( x^t_j \) from its buffer for computing \( h^t_i \) in step 5. Once \( u^t_i \) is computed and \( x^t_{i+1} \) is updated in steps 6 and 7, respectively, the active agent \( i \) broadcasts \( x^t_{i+1} \) to its neighbors, whose buffers store the updated \( x^t_{i+1} \). Since only one agent is active during each iteration, \( x^t_{j+1} = x^t_j \) for all \( j \neq i \). Therefore, the active agent can directly proceed to dual updates and finish its computing as in steps 9–14.

V. Analysis

In this section, we present a unified framework for analyzing the proposed algorithms with gradient, Newton, and BFGS updates. Throughout this section, we assume that the initialization requirement in Lemma 1 is satisfied. We recall the concatenated vector \( v = [x; \phi; \theta; \lambda] \in \mathbb{R}^{(2m+2)d} \) introduced in (22), and we similarly define \( v_\alpha = [x; z; \alpha; \theta; \lambda] \in \mathbb{R}^{(m+2n+2)d} \). We use \( v \) for implementation as in Algorithm 1 while analyze convergence using \( v_\alpha \) for technical convenience. We note that their equivalence is established by Lemma 1 and the definition \( \phi = E^T_s \alpha \). We first establish sublinear convergence rate of the synchronous DRUID (Algorithm 1) under the assumption that the local objective functions are convex. By further assuming strong convexity, we establish global linear convergence rate for both the synchronous and the asynchronous settings.

A. Preliminaries

We begin by stating the assumptions for our analysis.

**Assumption 1.** (Existence of solutions) The solution set \( \mathcal{X}^* \) of problem (1) is nonempty, i.e., \( \mathcal{X}^* \neq \emptyset \).
Assumption 2. The local costs functions $f_i(\cdot)$ and the regularizer function satisfy the following conditions:

(i) Each $f_i : \mathbb{R}^d \to \mathbb{R}$ is twice continuously differentiable, $m_f$–strongly convex and $M_f$–smooth, i.e., $\forall i \in [m], x_i \in \mathbb{R}^d$:

$$m_f I_d \preceq \nabla^2 f_i(x_i) \preceq M_f I_d,$$

where $0 \leq m_f \leq M_f < \infty$.

(ii) The regularizer function $g(\cdot) : \mathbb{R}^d \to \mathbb{R}$ is proper, closed, and convex, i.e., $\forall x, y \in \mathbb{R}^d$,

$$(x - y)^\top (\partial g(x) - \partial g(y)) \geq 0,$$

where $\partial g(\cdot)$ denotes the subdifferential set.

Assumption 3. The Hessians of the local objective functions are Lipschitz continuous with constant $L_f$, i.e., $\forall i \in [m], x, y \in \mathbb{R}^d$,

$$\|\nabla^2 f_i(x) - \nabla^2 f_i(y)\| \leq L_f \|x - y\|.$$

Note that we allow the case $m_f = 0$ (convex but not strongly convex), and we will analyze separately for the cases $m_f = 0$ and $m_f > 0$ to establish sublinear and linear rates, respectively.

Assumptions 1-2 are standard for analyzing distributed algorithms while Assumption 3 is standard for analyzing second order methods [48].

Assumption 4. The Hessian estimate obtained by the BFGS is uniformly upper bounded, i.e., for any $t \geq 0$, there exists a constant $\psi > 0$ such that:

$$H^t_{BFGS} \preceq \psi I_{md}.$$

Remark 3. Assumption 4 applies only for BFGS updates and is, in general, not standard. However, many techniques can be used to satisfy (27). For example, adding small regularization when computing the Hessian inverse approximations, i.e., $(H^t_{BFGS})^{-1} = (\hat{H}^t_{BFGS})^{-1} + \frac{1}{\psi} I_{md}$, where $(\hat{H}^t_{BFGS})^{-1}$ is obtained through (10). Other means for ascertaining the upper bound in (27) include using regularized BFGS updates and invoking L-BFGS estimation by using a finite prescribed number of $\{q^t_i, s^t_i\}$ copies. In brief, we make this assumption for convenience and without serious loss in generality; see also [33] and [49].

When local functions are assumed to be only convex ($m_f = 0$), there might be multiple optimal primal solutions $x^*$. For each optimal primal solution $x^*$, there are multiple optimal
dual solutions $\alpha^*$. However, there exists a unique dual pair that lies in the column space of $C := \begin{bmatrix} E_s \\ S^T \end{bmatrix}$. We formalize this in the following lemma.

**Lemma 2.** The tuple $(x^*, z^*, \alpha^*, \theta^*, \lambda^*)$ solves (3), and equivalently (1), if and only if the following holds:

\[
\nabla F(x^*) + E_s^T \alpha^* + S \lambda^* = 0, \quad \text{KKTa}
\]
\[
\partial g(\theta^*) - \lambda^* \geq 0, \quad \text{KKTb}
\]
\[
E_s x^* = 0, \quad \text{KKTc}
\]
\[
E_u x^* = 2z^*, \quad \text{KKTd}
\]
\[
S^T x^* = \theta^*. \quad \text{KKTe}
\]

Moreover, there exists a unique dual optimal pair $[\alpha^*; \lambda^*] \in \mathbb{R}^{(n+1)d}$ that lies in the column space of $C := \begin{bmatrix} E_s \\ S^T \end{bmatrix} \in \mathbb{R}^{(n+1)d \times md}$.

**Proof:** See Appendix A

The above conditions can be used to characterize the suboptimality of iterates $(x^t, z^t, \alpha^t, \theta^t, \lambda^t)$ generated by (14), where KKTa-b can be interpreted as the optimality condition of minimizing the objective function, while KKTc-e measure consensus violation among agents. We proceed to establish a lemma that characterizes the suboptimality of the iterates when we replace the exact sub-optimization problem (6a) with a one-step update as in (14a).

**Lemma 3.** Consider the iterates generated by (14). The following holds:

\[
e^t + \nabla F(x^{t+1}) - \nabla F(x^*) + \epsilon(x^{t+1} - x^t) + E_s^T (\alpha^{t+1} - \alpha^*) + \mu_z E_u^T (z^{t+1} - z^t)
\]
\[
+ S (\lambda^{t+1} - \lambda^* + \mu_\theta (\theta^{t+1} - \theta^t)) = 0
\]

where the error term is:

\[
e^t = \nabla F(x^t) - \nabla F(x^{t+1}) + J^t(x^{t+1} - x^t).
\]

**Proof:** See Appendix A
B. Sublinear Convergence

We recall $J_t$ in (15) and the concatenated vector $v_\alpha \in \mathbb{R}^{(m+2n+2)d}$. We further define $\tilde{J}^t = J^t + \epsilon I$, and the scaling matrix $G^t$ as follows:

$$v_\alpha = \begin{bmatrix} x \\ z \\ \alpha \\ \theta \\ \lambda \end{bmatrix}, \quad G^t = \begin{bmatrix} J^t & 0 & 0 & 0 \\ 0 & 2\mu_z & 0 & 0 \\ 0 & 0 & \frac{2}{\mu_z} & 0 \\ 0 & 0 & 0 & \mu_\theta \\ 0 & 0 & 0 & \frac{1}{\mu_\theta} \end{bmatrix}. \quad (29)$$

**Theorem 1.** Recall the definition in (29). Consider the iterates generated by (14). We denote the smallest and the biggest eigenvalue of $L_u$ and $L_s$ as $\sigma_{\min}^{L_u}$ and $\sigma_{\max}^{L_s}$ respectively. Under Assumptions 1-4, $(m_f = 0)$, and we select $\mu_z$ and $\epsilon$ such that: $\epsilon > \frac{M_f}{2}, \mu_z \epsilon < \psi^2$. Then the following holds:

$$\frac{1}{T} \sum_{t=1}^{T} \|x^t\|^2_{L_s} + \frac{\mu_\theta}{T} \|x^t - \theta^t\|^2 + \frac{1}{T} \sum_{t=1}^{T} \|v_{\alpha,t+1} - v_{\alpha,t}\|^2_{G^t} \geq \frac{1}{T} \sum_{t=1}^{T} \left\{ \frac{\mu_z}{2} \left\| x^{t+1}_{L_s} - \theta^{t+1} \right\|^2 + \frac{\mu_\theta}{T} \left\| \nabla F(x^t) + E^s_{\alpha} \alpha^t + S\lambda^t \right\|^2 + \mu_\theta \left\| \theta^{t+1} - \theta^t \right\|^2 \\
+ 2\mu_z \left\| z^{t+1} - z^t \right\|^2 + \left( \frac{\mu_z}{2} - \frac{c_0^2}{2M^2} \right) \left\| x^t \right\|^2_{L_s} + \left( \mu_\theta - \frac{2c_\mu_\theta}{M^2(\rho-1)} \right) \left\| S^T x^t - \theta^t \right\|^2 \right\},$$

where we denote $d_{\max} = \max_i |N_i|$, $\rho > \max \left\{ \frac{2c_\mu_\theta}{M^2}, \sigma_{\max}^{L_s} \right\} + 1$, $M_{\text{Gradient}} = \mu_z d_{\max} + \epsilon + \mu_\theta$, $M_{\text{Newton}} = M_f + \mu_z d_{\max} + \epsilon + \mu_\theta$, $M_{\text{BFGS}} = \psi$.

**Proof:** See Appendix B

**Remark 4.** It is not hard to verify $z^t = \frac{1}{2} E_u x^t$ (Remark 1) and $\lambda^t \in \partial g(\theta^t)$ holds along the convergence path and establishing convergence amounts to satisfying KKTa,c,e. We proceed to explicate the convergence rate of these terms in the following.

**Corollary 1.** The running-average suboptimality residual and consensus errors converge as follows:

$$\frac{1}{T} \sum_{t=1}^{T} \left\| \nabla F(x^t) + E^s_{\alpha} \alpha^t + S\lambda^t \right\|^2 = O\left( \frac{1}{T} \right), \quad (31a)$$

$$\frac{1}{T} \sum_{t=1}^{T} \left\| x^t \right\|^2_{L_s} = O\left( \frac{1}{T} \right), \quad (31b)$$

$$\frac{1}{T} \sum_{t=1}^{T} \left\| S^T x^t - \theta^t \right\|^2 = O\left( \frac{1}{T} \right). \quad (31c)$$
Proof: See Appendix B

C. Linear Convergence

By further assuming strongly convex $f_i(\cdot)$ ($m_f > 0$), we establish linear convergence rate of DRUID. We show that the iterates converge to the unique $[x^*; z^*; \alpha^*; \theta^*; \lambda^*]$, where the dual pair $[\alpha^*; \lambda^*]$ lies in the column space of $C$ as shown in Lemma 2. We first bound the error in (28).

Lemma 4. Recall the error term defined in (28). The following holds:

$$
\|e^t\| \leq \tau^t \|x^{t+1} - x^t\|,
$$

where

$$
\tau^t_{\text{Gradient}} = M_f,
$$

$$
\tau^t_{\text{Newton}} = \min \left\{ 2M_f, \frac{L_f}{2} \|x^{t+1} - x^t\| \right\},
$$

$$
\tau^t_{\text{BFGS}} = \|H^t - H^{t+1}\| \leq 2\psi.
$$

Proof: See Appendix C

The above lemma complements the result presented in [29] and [31]. By upper bounding the error induced when we replace the exact suboptimization step (6a) with a one-step update (14a), we reveal the differences when using different computing schemes. Since the algorithm converges, as established in the previous subsection, the $\frac{L_f}{2} \|x^{t+1} - x^t\|$ term will eventually become smaller than the $2M_f$ term in (33b). In other words, the error term eventually diminishes quadratically with respect to $\|x^{t+1} - x^t\|$ in Newton updates. On the other hand, since $\|H^t - H^{t+1}\| \to 0$, we conclude that the error term in BFGS diminishes superlinearly with respect to $\|x^{t+1} - x^t\|$. In summary, as $t \to \infty$,

$$
e^t_{\text{Gradient}} = O(\|x^{t+1} - x^t\|),
$$

$$
e^t_{\text{Newton}} = O(\|x^{t+1} - x^t\|^2),
$$

$$
e^t_{\text{BFGS}} = o(\|x^{t+1} - x^t\|).
$$


Before establishing the linear convergence rate of DRUID, we recall $v_\alpha$ and introduce the following scaling matrix:

$$
v_\alpha = \begin{bmatrix} x \\ z \\ \alpha \\ \theta \\ \lambda \end{bmatrix}, \quad H = \begin{bmatrix} \epsilon & 0 & 0 & 0 \\ 0 & 2\mu_z & 0 & 0 \\ 0 & 0 & \frac{2}{\mu_z} & 0 \\ 0 & 0 & 0 & \mu_\theta \\ 0 & 0 & 0 & \frac{1}{\mu_\theta} \end{bmatrix}. \tag{35}
$$

**Theorem 2.** Recall the definition in (35). Under Assumptions 1–4, we denote the maximum and minimum eigenvalue of $L_u$ as $\sigma_{L_{\text{max}}}^u$ and $\sigma_{L_{\text{min}}}^u$ respectively. Let $\sigma_{\text{min}}^+$ be the smallest positive eigenvalue of $CC^\top$, where $C := \begin{bmatrix} E_s \\ S^\top \end{bmatrix}$, and $c_{\text{max}} = \max\{M_f, \psi\}$. By selecting $\mu_z = 2\mu_\theta$, $\epsilon > \frac{c_{\text{max}}^2(m_f+M_f)}{2m_fM_f}$, and arbitrary constant $\zeta \in \left(\frac{m_f+M_f}{2}, \frac{\epsilon}{(\tau/t)^2}\right)$, the iterates generated by (14) satisfy:

$$
\|v_{\alpha}^{t+1} - v_\alpha^*\|_H^2 \leq \frac{1}{1+\eta} \|v_\alpha^t - v_\alpha^*\|_H^2,
$$

where $\eta$ satisfies:

$$
\eta = \min \left\{ \left(\frac{2m_fM_f}{m_f+M_f} - 1 \right) \cdot \frac{1}{\epsilon + \mu_\theta(\sigma_{\text{max}}^u+2)} \cdot \frac{1}{2} \cdot \frac{\mu_\theta \sigma_{\text{min}}^+}{\sigma_{\text{max}}^+} \cdot \frac{\mu_\theta \sigma_{\text{min}}^+}{\sigma_{\text{max}}^+} \cdot 5 \cdot \frac{\sigma_{\text{min}}^+}{5 \max\{1, \sigma_{L_{\text{max}}}^u\}} \right\}. \tag{36}
$$

**Proof:** See Appendix C.

**Remark 5.** To shed some light on the convergence rate, we first consider the case when the sub-optimization problem (6a) is solved exactly. When an exact solution is obtained from the step (6a), Lemma 3 holds with $e^t = 0$, and therefore $\tau^t = 0$ in Lemma 4. Having $\tau^t = 0$ allows us to choose $\epsilon = 0$ and $\zeta \gg 1$, which gives the following rate:

$$
\eta_{\text{exact}} = \min \left\{ \left(\frac{2m_fM_f}{m_f+M_f} - 1 \right) \cdot \frac{1}{\epsilon + \mu_\theta(\sigma_{\text{max}}^u+2)} \cdot \frac{1}{2} \cdot \frac{\mu_\theta \sigma_{\text{min}}^+}{\sigma_{\text{max}}^+} \cdot \frac{\mu_\theta \sigma_{\text{min}}^+}{\sigma_{\text{max}}^+} \cdot 5 \cdot \frac{\sigma_{\text{min}}^+}{5 \max\{1, \sigma_{L_{\text{max}}}^u\}} \right\}. \tag{37}
$$

Denoting $\kappa = M_f/m_f$ and choosing $\mu_\theta = m_f\sqrt{\kappa}$, we obtain an iteration complexity of $O\left(\sqrt{\kappa} \log(1/\epsilon)\right)$ from (37). Moreover, since $\sigma_{\text{min}}^+$ is related to the smallest positive eigenvalue of $L_s$, i.e., the algebraic connectivity of the graph, (37) implies that a more connected graph (larger $\sigma_{\text{min}}^+$) gives rises to a larger $\eta_{\text{exact}}$, and faster convergence rates. On the other hand, the rate $\eta$ established in (36) is no larger than $\eta_{\text{exact}}$ in (37): this is due to the fact that we have replaced the exact optimization step with the one step update (14a). Characterizing the gap between $\eta$ and $\eta_{\text{exact}}$ serves to reveal the differences between using gradient, Newton, and BFGS updates.
This is achieved by comparing the upper bound for the error term $\tau^t$ in Lemma 4 as well as how $e^t$ evolves in (34). As established in Section V-B, $\lim_{t \to \infty} \| x^{t+1} - x^t \| = 0$, the error bound $\tau_{\text{Newton/BFGS}}^t \to 0$ from inspecting (33b) and (33c). In other words, we can recover the convergence rate in (37) only if we use curvature aided updates.

We recall that the asynchronous implementation in (24) is defined using $\{ \phi_i \}_{i=1}^m$ and $v = [x; \phi; \theta; \lambda]$, for the most efficient and economical deployment. In the rest of this section, we first characterize the condition for $v_\alpha$ to converge under random activation, and then show that the implementation (24) satisfies this condition. We first define the following activation matrix corresponding to $v_\alpha = [x; z; \alpha; \theta; \lambda] \in \mathbb{R}^{(m+2n+2)d}$:

$$\Omega^t_\alpha := \begin{bmatrix} X^t & 0 & 0 & 0 & 0 \\ 0 & Y^t & 0 & 0 & 0 \\ 0 & 0 & Y^t & 0 & 0 \\ 0 & 0 & 0 & X^t_{ll} & 0 \\ 0 & 0 & 0 & 0 & X^t_{ll} \end{bmatrix}. \tag{38}$$

The activation matrix $\Omega^t_\alpha$ differs from $\Omega^t$ in (23) as we allow $(z^t, \alpha^t)$ to be updated independently from $x^t$, captured by the random matrix $Y^t \in \mathbb{R}^{d \times nd}$. We can similarly develop an asynchronous algorithm as:

$$v^{t+1}_\alpha = v^t_\alpha + \Omega^t_\alpha (T_\alpha v^t_\alpha - v^t_\alpha), \tag{39}$$

where the operator $T_\alpha : \mathbb{R}^{(m+2n+2)d} \to \mathbb{R}^{(m+2n+2)d}$ is equivalent to the synchronous updates (14). The update (39) captures a wider range of random activation schemes than the update in (24), but it is more costly to implement. Therefore, we only use (39) as a guideline for analysis. We proceed to introduce the following notation $\mathbb{E}^t[\cdot] := \mathbb{E}[\cdot|\mathcal{F}^t]$, where $\mathcal{F}^t$ is the filtration generated by $(X^t, \ldots, X^t)$ and $(Y^t, \ldots, Y^t)$.

**Theorem 3.** Consider the iterates generated by the asynchronous algorithm (39). Under the same setting as in the Theorem 2 and any activation scheme such that $\mathbb{E}^t[\Omega^t_\alpha] = \Omega_\alpha \succ 0$, then the following holds:

$$\mathbb{E}^t \left[ \| v^{t+1}_\alpha - v^*_\alpha \|^2_{H\Omega^{-1}_\alpha} \right] \leq \left( 1 - \frac{p^\min}{1 + \eta} \right) \| v^t_\alpha - v^*_\alpha \|^2_{H\Omega^{-1}_\alpha},$$

where for $i \in [m], j \in [n]$, we denote $\mathbb{E}^t[X^t_{ii}] = p^X_i, \mathbb{E}^t[Y^t_{kk}] = p^Y_k, p^\min := \min_{i \in [m], k \in [n]} \{ p^X_i, p^Y_k \}$, and $\eta$ is given by (36).

**Proof:** See Appendix C.
We note that the activation of the asynchronous scheme using \((\Omega_{\alpha}^{t+1}, v_{\alpha}^{t+1})\) described by (39) amounts to specifying the random matrix \(X^{t+1}\) and \(Y^{t+1}\), which can be chosen independently from each other. Theorem 3 shows that as long as \(E_t[\Omega_{\alpha}^{t+1}] \succ 0\), iterates \(v_{\alpha}^{t}\) converge linearly in expectation. This translates to each \(x_i, i \in [m]\), and \((z_k, \alpha_k), k \in [n]\), being updated infinitely often. On the other hand, the more practical implementation using \((\Omega^{t+1}, v^{t+1})\) described by (24), only needs to specify \(X^{t+1}\), i.e., activating agents. Note that irrespective of the activation scheme, dual updates are incorporated in the form of \(\phi^t = E_s \alpha^t\). The difference of the two activation schemes lies in the fact that (39) first updates a subset of \(\alpha_k\), then computes \(\phi = E_s^T \alpha\), while (24) directly updates a subset of \(\phi_i\). In the following corollary, we show that using the activation scheme described by (24), the induced iterates \(v_{\alpha}^{t}\) converge linearly in expectation.

**Corollary 2.** Consider activation matrices \(\Omega^{t+1}\) in (23) and \(\Omega_{\alpha}^{t+1}\) in (38). Under the same \(X^{t+1}\) and updating scheme (24), if \(E_t[X^{t+1}] \succ 0\), then it holds that \(E_t[\Omega_{\alpha}^{t+1}] \succ 0\).

**Proof:** See Appendix C.

Since \(E_t[\Omega_{\alpha}^{t+1}] \succ 0\), we conclude that the implementation using \((\Omega^t, v^t)\) induces an equivalent sequence of \((\Omega_{\alpha}^t, v_{\alpha}^t)\) that converges linearly in expectation using Theorem 3.

**VI. Numerical experiments**

In this section, we present a comparative experimental validation of the proposed methods with existing state-of-the-art methods, namely PG-EXTRA [9], P2D2 [11], and ESOM [32]. Note that ESOM and other existing (quasi) Newton methods [17], [31], [33], [43] do not support nonsmooth regularization functions and therefore ESOM is not compared in the Fig 1 and Fig 2 where the regularization function is nondifferentiable. We consider the following distributed optimization problem:

\[
\min_{x \in \mathbb{R}^d} G(x) = \left\{ \sum_{i=1}^{m} f_i(x) + g(x) \right\},
\]

(40)

where \(f_i(\cdot)\) and \(g(\cdot)\) are to be specified according to the application. All experiments are conducted using real-life data sets from the LIBSVM\(^1\) and UCI Machine Learning Repository\(^2\). We generate connected random graphs with \(m\) agents by repetitively drawing edges between agents according to a Bernoulli(p) distribution. We ensure connectedness by redrawing the graph.

---

\(^1\)https://www.csie.ntu.edu.tw/~cjlin/libsvm/
\(^2\)https://archive.ics.uci.edu/ml/index.php
Figure 1: Performance comparison on the CCPP dataset. We plot the iteration number versus the relative cost error (left) \( \frac{G(x_t) - G(x^*)}{G(x_0) - G(x^*)} \) and the relative distance error (right) \( \frac{\|x_t - x^*\|}{\|x_0 - x^*\|} \) on a randomly generated graph consisting of \( m = 20 \) agents.

if necessary. The mixing matrix of P2D2 and ESOM are generated using the Metropolis rule while the mixing matrix of PG-EXTRA is generated by the Laplacian based constant weight matrix, respectively.

A. Distributed LASSO

The distributed LASSO problem considers solving (40) with \( g(x) = \gamma \|x\|_1 \) and \( f_i(\cdot) : \mathbb{R}^d \rightarrow \mathbb{R} \) defined as:

\[
f_i(x) = \frac{1}{2} \sum_{i=1}^{m_i} \left( a_i^\top x - b_i \right)^2. \tag{41}
\]

Each \( \{a_i, b_i\} \in \mathbb{R}^d \times \mathbb{R} \) is a given data point and \( m_i \) denotes the total number of data points held by the \( i \)-th agent. The purpose of the regularization function \( \gamma \|x\|_1 \) is to promote a sparse solution vector. We consider the combined cycle power plant (CCPP) dataset from the UCI Machine Learning Repository, consisting of 9,000 data points of dimension \( d = 4 \) collected from a combined cycle power plant over 6 years. The goal is to predict net hourly electrical energy output of the plant given 4 features of the plant. We evenly distribute the data points among \( m = 20 \) agents in a randomly generated graph with \( p = 0.2 \) and \( \gamma = 0.002 \).

We note that in Figure 1, all algorithms have the same communication costs per iteration while first order methods have lower computational costs. However, first order methods suffer from slower convergence speed. Significant reduction of iteration numbers for prescribed accuracy
can be achieved by using (quasi) Newton methods. Moreover, (quasi) Newton methods become necessary when high accuracy solution is desired as first order methods tend to stagnate at error levels $10^{-2} \sim 10^{-4}$. We note that using BFGS estimated curvature information requires less computational resources compared to using the Newton method but the convergence of DRUID-BFGS is inferior to that of the DRUID-Newton, as expected. Therefore, DRUID-BFGS strikes a fine balance between the proposed algorithms with first order and second order updates, in terms of computational costs, convergence speed, and achieved solution accuracy.

B. Distributed Logistic Regression

The distributed logistic regression solves \cite{40} with $g(x) = \gamma \|x\|_1$ and $f_i(\cdot) : \mathbb{R}^d \to \mathbb{R}$ defined as:

$$f_i(x_i) = \sum_{j=1}^{m_i} \left[ \ln \left( 1 + e^{-w_j^T x_i} \right) + (1 - y_j)w_j^T x_i \right],$$

where $m_i$ is the number of data points accessible by the $i$-th agent. We denote the local training data set as $\{w_j, y_j\}_{j=1}^{m_i} \subset \mathbb{R}^d \times \{0, 1\}$, where $w_j$ are feature vectors and $y_j$ are known labels. In the case of logistic regression, we consider 2,000 data points from the ijcnn1 dataset with dimensions $d = 22$. We explore the effect of the graph topology by varying the size of the network $m$. As shown in Figures 2 algorithms with curvature aided information consistently
Figure 3: Performance comparison using the space ga dataset. We plot the total communication rounds versus the relative cost error $\frac{G(x^t) - G(x^*)}{G(x^0) - G(x^*)}$ on random graphs, with number of agents and probability of generating an edge equal to $m = 20, p = 0.2$ (left) and $m = 40, p = 0.8$ (right).

outperform first order algorithms. Moreover, we observe that DRUID is insensitive to networks with different sizes, but with fixed $p$. This is consistent with our analysis where the convergence rate is affected by algebraic connectivity, but not system size $m$.

C. Distributed Ridge Regression

Since existing second order methods only support differentiable regularization functions, we consider the problem of distributed ridge regression, whose $f_i(\cdot)$ is same as in (41) but with $g(x) = \gamma \|x\|^2$. We compare DRUID with ESOM-$K$, where $K$ denotes the number of inner communication loops. The total number of communication rounds between each agent is used as the metric for comparing, and we note that for ESOM-0 (no inner communication loops), the communication costs per round is same as DRUID. In the case of ESOM-$K$, a more accurate Hessian estimation can be obtained by increasing $K$, at the cost of more communication rounds. On the other hand, we emphasize that through use of consensus variables $\{z_{ij}\}$, DRUID-Newton utilizes the exact Hessian without inducing inner loops, and thus achieves the highest communication efficiency as shown in the Figure 3.

VII. Conclusions

We have proposed a family of distributed primal-dual algorithms for solving convex composite optimization problems. Various computing choices, including gradient, Newton, and BFGS
updates, are proposed to achieve a balance between economical computational costs, solution accuracy, and convergence speeds. By use of intermediate consensus variables, we achieve a block-diagonal Hessian that allows us to harness the curvature information without additional communication rounds after each gradient evaluation. Thus, all methods feature the same minimal communication complexity. An asynchronous extension of the proposed algorithms is also presented that further removes the need for global synchronization [50]. We establish a unified analytical framework for the proposed algorithms that theoretically reveals the difference between using different updating schemes.

APPENDIX A

Proof of Lemma 1: We first write \( y^t = [\alpha^t; \beta^t] \) and recall the dual update for \( y^{t+1} \) in (6d):

\[
y^{t+1} = y^t + \mu_z(Ax^{t+1} - Bz^{t+1}).
\]

Using (13) and premultiplying (6d) with \( B^\top \) on both sides, we obtain \( B^\top y^{t+1} = 0 \) for all \( t \geq 0 \). Since \( B^\top = [I_{nd}, I_{nd}] \), it holds that \( \alpha^{t+1} + \beta^{t+1} = 0 \) for all \( t \geq 0 \). By further assuming \( \alpha^0 = -\beta^0 \), we obtain \( \alpha^t = -\beta^t \) for all \( t \geq 0 \). Recall \( A_s := \hat{A}_s \otimes I_d \) and \( A_d := \hat{A}_d \otimes I_d \), as well as the definition of \( A \) in (3), the dual update (6d) can be rewritten as:

\[
\alpha^{t+1} = \alpha^t + \mu_z(A_s x^{t+1} - z^{t+1}),
\]

(42)

\[
-\alpha^{t+1} = -\alpha^t + \mu_z(A_d x^{t+1} - z^{t+1}).
\]

(43)

Recall that \( E_s = A_s - A_d \) and \( E_u = A_s + A_d \). By taking the sum and difference of (42) and (43), we obtain for \( t \geq 0 \),

\[
z^{t+1} = \frac{\mu_z}{2} E_u x^{t+1},
\]

(44)

\[
\alpha^{t+1} = \alpha^t + \frac{\mu_z}{2} E_s x^{t+1}.
\]

(45)

This establishes that the dual update (6d) for \( y^{t+1} \) can be replaced by (45). Using the definition of \( \phi^t = E_s^\top \alpha^t \) and premultiplying (45) with \( E_s^\top \), we obtain (14c). By initializing \( z^0 = \frac{1}{2} E_u x^0 \), we have that \( z^t = \frac{1}{2} E_u x^t \) for \( t \geq 0 \). Therefore, the update (13) for \( z^{t+1} \) is not necessary since \( z^t \) can be obtained by computing \( \frac{1}{2} E_u x^t \). It remains to show the equivalence between (11) and (14a). Using (5), it follows that update (11) is given by:

\[
x^{t+1} = x^t - (H^t)^{-1} [\nabla F(x^t) + A^\top y^t + SL^t + \mu_z A^\top (Ax^t - Bz^t) + \mu_\theta S(S^\top x^t - \theta^t)].
\]

(12)
Since \( y^t = [\alpha^t; -\alpha^t] \) and \( z^t = \frac{1}{2} E_u x^t \), we obtain:
\[
A^\top y^t = \begin{bmatrix} A^\top_s & A^\top_d \end{bmatrix} \begin{bmatrix} \alpha^t \\ -\alpha^t \end{bmatrix} = E^\top_s \alpha^t = \phi^t, \tag{46}
\]
\[
\mu z^\top A (Ax^t - Bz^t) = \frac{\mu}{2} (2D - L_u) x^t = \frac{\mu}{2} L_u x^t, \tag{47}
\]
where we have used the identity \( A^\top_s - A^\top_d = E^\top_s \), \( D = A^\top A \), and \( \mu z^\top Bz^t = \frac{\mu}{2} E^\top_u E_u x^t = \frac{\mu}{2} L_u x^t \). After substituting (46) and (47) into (11), we obtain the desired. \( \blacksquare \)

**Proof of Lemma 2**: The KKT conditions for (3) are given by:
\[
\nabla F(x^*) + A^\top y^* + S\lambda^* = 0, \tag{48a}
\]
\[
B^\top y^* = 0, \tag{48b}
\]
\[
\partial g(\theta^*) - \lambda^* \ni 0, \tag{48c}
\]
\[
Ax^* = Bz^*, \tag{48d}
\]
\[
S^\top x^* = \theta^*. \tag{48e}
\]
Since the objective function is convex with linear constraints, strong duality holds. Recall the definition \( B = \begin{bmatrix} I_{md} & I_{md} \end{bmatrix} \in \mathbb{R}^{2md \times md} \). The condition (48b) implies that for any dual optimal \( y^* = [\alpha^*; \beta^*] \), it holds that \( \alpha^* = -\beta^* \). Since \( A = [A_s; A_d] \) and \( E_s = A_s - A_d \), the condition (48a) can be rewritten as:
\[
\nabla F(x^*) + E^\top_s \alpha^* + S\lambda^* = 0. \tag{49}
\]
Note that since \( E^\top_s \) has a nontrivial kernel for any network with agent number \( m > 1 \), there exist multiple \( \alpha^* \) that satisfy (49). We proceed to show that there exists a unique dual optimal \( [\alpha^*; \lambda^*] \) that lies in the column space of \( C = \begin{bmatrix} E_s \\ S^\top \end{bmatrix} \). To show existence, let \( \xi^0 := \begin{bmatrix} \alpha^0 \\ \lambda^0 \end{bmatrix} \) be any dual optimal that satisfies (49) and (48c). We denote its projection to the column space of \( C \) as \( \xi^* := [\alpha^*; \lambda^*] \). By the property that \( C^\top (\xi^0 - \xi^*) = 0 \), we conclude that \( \nabla F(x^*) + C^\top \xi^* = 0 \). Moreover, since \( \text{col}(E^\top_s) \cap \text{col}(S) = 0 \) and \( \text{ker}(S) = 0 \), it holds that \( \lambda^0 = \lambda^* \). We prove the uniqueness of \( \xi^* \) by contradiction. Suppose there exist \( \xi^1 = Cr^1 \) and \( \xi^2 = Cr^2, \ r^1 \neq r^2 \), that satisfy:
\[
\nabla F(x^*) + C^\top Cr^1 = 0,
\]
\[
\nabla F(x^*) + C^\top Cr^2 = 0.
\]
After taking the difference of the above, we obtain $C^\top C(r_1 - r_2) = 0$. Note that $C^\top C = E_s^\top E_s + SS^\top = L_s + SS^\top$. Since both $L_s$ and $SS^\top$ are positive semidefinite, $C^\top C(r_1 - r_2) = 0$ if and only if $L_s(r_1 - r_2) = SS^\top(r_1 - r_2) = 0$. Moreover, since the graph is connected, the kernel of $L_s$ is a subspace spanned by $1 \otimes I_d$, where $1 \in \mathbb{R}^m$ is the all one vector, and the kernel of $SS^\top$ is spanned by vectors with the $t$-th entry being 0. Therefore, $L_s(r_1 - r_2) = SS^\top(r_1 - r_2) = 0$ if and only if $r_1 - r_2 = 0$, which contradicts with the assumption $r_1 \neq r_2$.

Proof of Lemma 3: Recall the primal update (14a) and the identity $\phi^t = E_s^\top \alpha^t$. After rearranging, we obtain:

$$\nabla F(x^t) + E_s^\top \alpha^t + S\lambda^t + \frac{\mu_s}{2} L_s x^t + \mu_\theta S(S^\top x^t - \theta^t) + H^t(x^{t+1} - x^t) = 0 \quad (50)$$

From the dual update (14c), we obtain:

$$E_s^\top \alpha^t + \frac{\mu_s}{2} L_s x^t = E_s^\top \alpha^{t+1} - \frac{\mu_s}{2} L_s(x^{t+1} - x^t). \quad (51)$$

Similarly, from the dual update (14d), it holds that:

$$S\lambda^t + \mu_\theta S(S^\top x^t - \theta^t) = S\lambda^{t+1} - \mu_\theta S(S^\top x^{t+1} - x^t) - (\theta^{t+1} - \theta^t). \quad (52)$$

After substituting (51) and (52) into (50), we obtain:

$$\nabla F(x^t) + E_s^\top \alpha^{t+1} - \frac{\mu_s}{2} L_s(x^{t+1} - x^t) + S\lambda^{t+1} - \mu_\theta S(S^\top x^{t+1} - x^t) + \mu_\theta (\theta^{t+1} - \theta^t) + H^t(x^{t+1} - x^t) = 0 \quad (53)$$

Recall $2D = L_s + L_u$ from (4c). After adding and subtracting $(\mu_s D + \mu_\theta SS^\top + \epsilon I)(x^{t+1} - x^t)$ from (53), we obtain:

$$\nabla F(x^t) + E_s^\top \alpha^{t+1} - \frac{\mu_s}{2} L_u(x^{t+1} - x^t) + S\lambda^{t+1} + \mu_\theta S(\theta^{t+1} - \theta^t) + \epsilon(x^{t+1} - x^t) + (H^t - \mu_s D - \mu_\theta SS^\top - \epsilon I)(x^{t+1} - x^t) = 0.$$  

Moreover, $\frac{\mu_s}{2} L_u(x^{t+1} - x^t) = \frac{\mu_s}{2} E_u^\top E_u(x^{t+1} - x^t) = \mu_\theta E_u^\top(z^{t+1} - z^t)$ by (44). Recall the construction of $H^t$ in (15), as well as (20) for the BFGS case. After subtracting KKTa and substituting the definition of $e^t$ in (28) and the expression for $\frac{\mu_s}{2} L_u(x^{t+1} - x^t)$ into the above, we obtain the desired.
APPENDIX B

Proof of Theorem 1: We begin with proving the following two technical inequalities:

\[(\lambda^{t+1} - \lambda^t) \trans (\theta^{t+1} - \theta^t) \geq 0, \quad (54)\]
\[(\lambda^{t+1} - \lambda^*) \trans (\theta^{t+1} - \theta^*) \geq 0. \quad (55)\]

From the definition of the proximal operator, it holds that:

\[\theta^{t+1} = \arg\min_{\theta} \left\{ g(\theta) + \frac{\mu_\theta}{2} \left\| S^\trans x^{t+1} + \frac{1}{\mu_\theta} \lambda^t - \theta \right\|^2 \right\}. \quad (56)\]

By the optimality condition of (56), we obtain: \(0 \in \partial g(\theta^{t+1}) - \mu_\theta (\frac{1}{\mu_\theta} \lambda^t + S^\trans x^{t+1} - \theta^{t+1}) = \partial g(\theta^{t+1}) - \lambda^{t+1}\), where the last equality follows from the dual update (14d). Therefore, it holds that: \((\lambda^{t+1} - \lambda^t) \trans (\theta^{t+1} - \theta^t) \in (\partial g(\theta^{t+1}) - \partial g(\theta^t)) \trans (\theta^{t+1} - \theta^t) \geq 0\), where the inequality follows from the convexity of \(g(\cdot)\). Moreover, \((\lambda^{t+1} - \lambda^*) \trans (\theta^{t+1} - \theta^*) \in (\partial g(\theta^{t+1}) - \partial g(\theta^*) \trans (\theta^{t+1} - \theta^*) \geq 0\), where the inclusion follows from KKTb. The rest of the proof is constituted by the following:

- (i) Establishing the convergence of \(\|v^t_\alpha - v^*_\alpha\|_G^2\) to 0.
- (ii) Establishing the running average upper bound in (30).

Part (i): Since \(F(\cdot)\) is convex with the gradient being Lipschitz continuous with parameter \(M_f\), the following holds:

\[\frac{1}{M_f} \|\nabla F(x^t) - \nabla F(x^*)\|^2 \leq (x^t - x^*) \trans (\nabla F(x^t) - \nabla F(x^*)) = (x^{t+1} - x^*) \trans (\nabla F(x^t) - \nabla F(x^*)) + (x^t - x^{t+1}) \trans (\nabla F(x^t) - \nabla F(x^*)). \quad (57)\]

We proceed to establish an upper bound for the right-hand side of (57) by separately bounding the two components. Recall \(H^t\) in (15). From Lemma 3, the following holds:

\[\nabla F(x^t) - \nabla F(x^*) = -\{ E^\trans_s (\alpha^{t+1} - \alpha^*) + S (\lambda^{t+1} - \lambda^* + \mu_\theta (\theta^{t+1} - \theta^t)) + (J^t + \epsilon I) (x^{t+1} - x^t) + \mu_z E^\trans_u (z^{t+1} - z^t) \}. \quad (58)\]

Denoting \(\overline{J}^t = J^t + \epsilon I\) and using (58), we rewrite \((x^{t+1} - x^*) \trans (\nabla F(x^t) - \nabla F(x^*))\) in (57) as:

\[(x^{t+1} - x^*) \trans (\nabla F(x^t) - \nabla F(x^*)) = - (x^{t+1} - x^*) \trans E^\trans_s (\alpha^{t+1} - \alpha^*) - (x^{t+1} - x^*) \trans S (\lambda^{t+1} - \lambda^* + \mu_\theta (\theta^{t+1} - \theta^t)) - (x^{t+1} - x^*) \trans \overline{J}^t (x^{t+1} - x^t) - \mu_z (x^{t+1} - x^*) \trans E^\trans_u (z^{t+1} - z^t). \quad (59)\]
From the dual update, KKT conditions, and Lemma [1] the following holds: $(x^{t+1} - x^*)^\top E_s^\top = \frac{2}{\mu_x} (\alpha^{t+1} - \alpha^t)^\top, (x^{t+1} - x^*)^\top S = (\theta^{t+1} - \theta^*)^\top + \frac{1}{\mu_\theta} (\lambda^{t+1} - \lambda^t)^\top, (x^{t+1} - x^*)^\top E_u^\top = (z^{t+1} - z^*)^\top$

Using these expressions for $(x^{t+1} - x^*)^\top E_s^\top, (x^{t+1} - x^*)^\top S, \text{ and } (x^{t+1} - x^*)^\top E_u^\top$, we rewrite (59) as:

$$
(x^{t+1} - x^*)^\top (\nabla F(x^t) - \nabla F(x^*))
$$

$$
= -\frac{2}{\mu_x} (\alpha^{t+1} - \alpha^t)^\top (\alpha^{t+1} - \alpha^*) - (\theta^{t+1} - \theta^*)^\top (\lambda^{t+1} - \lambda^*) - \frac{1}{\mu_\theta} (\lambda^{t+1} - \lambda^t)^\top (\lambda^{t+1} - \lambda^*)
$$

$$
- \mu_\theta (\theta^{t+1} - \theta^*)^\top (\theta^{t+1} - \theta^t) - (\lambda^{t+1} - \lambda^t)^\top (\theta^{t+1} - \theta^t)
$$

$$
- (x^{t+1} - x^*)^\top \mathcal{J} (x^{t+1} - x^t)
$$

$$
\leq -\frac{2}{\mu_x} (\alpha^{t+1} - \alpha^t)^\top (\alpha^{t+1} - \alpha^*) - \frac{1}{\mu_\theta} (\lambda^{t+1} - \lambda^t)^\top (\lambda^{t+1} - \lambda^*) - \mu_\theta (\theta^{t+1} - \theta^*)^\top (\theta^{t+1} - \theta^t)
$$

$$
- (x^{t+1} - x^*)^\top \mathcal{J} (x^{t+1} - x^t) - 2\mu_z (z^{t+1} - z^*)^\top (z^{t+1} - z^t)
$$

$$
\leq \frac{1}{2} \left\{ \frac{2}{\mu_x} \left( \| \alpha^t - \alpha^* \|^2 - \| \alpha^{t+1} - \alpha^* \|^2 - \| \alpha^{t+1} - \alpha^t \|^2 \right) 
$$

$$
+ \frac{1}{\mu_\theta} \left( \| \lambda^t - \lambda^* \|^2 - \| \lambda^{t+1} - \lambda^* \|^2 - \| \lambda^{t+1} - \lambda^t \|^2 \right) 
$$

$$
+ \mu_\theta \left( \| \theta^t - \theta^* \|^2 - \| \theta^{t+1} - \theta^* \|^2 - \| \theta^{t+1} - \theta^t \|^2 \right) 
$$

$$
+ \| x^t - x^* \|^2_{\mathcal{J}^t} - \| x^{t+1} - x^* \|^2_{\mathcal{J}^t} - \| x^{t+1} - x^t \|^2_{\mathcal{J}^t} 
$$

$$
+ 2\mu_z \left( \| z^t - z^* \|^2 - \| z^{t+1} - z^* \|^2 - \| z^{t+1} - z^t \|^2 \right) \right\}
$$

$$
= \frac{1}{2} \left( \| v^t_{\alpha} - v^*_{\alpha} \|^2_{G^t} - \| v^{t+1}_{\alpha} - v^*_{\alpha} \|^2_{G^t} - \| v^{t+1}_{\alpha} - v^t_{\alpha} \|^2_{G^t} \right),
$$

(60)

where (i) follows from the identity $-2(a - b)^\top (a - c) = \| b - c \|^2 - \| a - b \|^2 - \| a - c \|^2$; (ii) follows from the definition (29). We proceed to establish an upper bound for the second term of (57) in the following. Note that $a^\top b \leq \frac{1}{2\zeta} a^2 + \frac{\zeta}{2} b^2$ holds for any $\zeta > 0$. By setting $\zeta = \frac{M_f}{2}$, we obtain the following:

$$
(x^t - x^{t+1})^\top (\nabla F(x^t) - \nabla F(x^*)) \leq \frac{1}{M_f} \| \nabla F(x^t) - \nabla F(x^*) \|^2 + \frac{M_f}{4} \| x^{t+1} - x^t \|^2.
$$

(61)

After substituting (60) and (61) into (57), we obtain:

$$
\frac{1}{M_f} \| \nabla F(x^t) - \nabla F(x^*) \|^2 \leq \frac{1}{2} \left( \| v^t_{\alpha} - v^*_{\alpha} \|^2_{G^t} - \| v^{t+1}_{\alpha} - v^*_{\alpha} \|^2_{G^t} - \| v^{t+1}_{\alpha} - v^t_{\alpha} \|^2_{G^t} \right)
$$

$$
+ \frac{1}{M_f} \| \nabla F(x^t) - \nabla F(x^*) \|^2 + \frac{M_f}{4} \| x^{t+1} - x^t \|^2.
$$
By canceling the identical term and rearranging, we obtain:

\[
\frac{1}{2} \left( \|v^t_\alpha - v^*_\alpha\|^2_{G^t} - \|v^{t+1}_\alpha - v^*_\alpha\|^2_{G^t} \right) \geq \frac{1}{2} \left( \|v^{t+1}_\alpha - v^*_\alpha\|^2_{G^t} - \frac{M_f}{\delta} \|x^{t+1} - x^t\|^2 \right)
\]

\[
= \frac{1}{2} \left( \|x^{t+1} - x^t\|^2_{J^t} - \frac{M_f}{\delta} I \right) + 2\mu_z \|z^{t+1} - z^t\|^2 + \mu_\theta \|\theta^{t+1} - \theta^t\|^2 + \frac{\mu_\theta}{\mu_\lambda} \|\lambda^{t+1} - \lambda^t\|^2.
\]

Recall \(J^t = J^t + \epsilon I\) and we proceed to find a uniform lower bound for \(\|x^{t+1} - x^t\|^2_{J^t}\), for gradient descent, Newton, and BFGS computing scheme. Since \(J^t_{\text{Gradient}} = 0\) and \(J^t_{\text{Newton}} = \nabla^2 F(x^t) \succeq 0\) by construction, it holds that \(\epsilon I \preceq J^t_{\text{Gradient/Newton}}\). It remains to find a lower bound for the case of BFGS. Recall \(J^t_{\text{BFGS}}\) defined in (20). By the secant condition, \((J^t_{\text{BFGS}} + \mu_z D + \mu_\theta SS^T + \epsilon I) s^{t-1} = q^{t-1}\), where \(s^{t-1} = x^t - x^{t-1}\), \(q^{t-1} = \nabla F(x^t) - \nabla F(x^{t-1}) + (\mu_z D + \mu_\theta SS^T + \epsilon I) s^{t-1}\). Therefore, it holds that:

\[
J^t_{\text{BFGS}} s^{t-1} = \nabla F(x^t) - \nabla F(x^{t-1}).
\]

By premultiplying \((s^{t-1})^\top\) on both sides of (63), we obtain:

\[
(s^{t-1})^\top J^t_{\text{BFGS}} s^{t-1} = (x^t - x^{t-1})^\top (\nabla F(x^t) - \nabla F(x^{t-1})) \geq 0.
\]

Therefore, we obtain \(\sigma_{\text{min}} J^t_{\text{BFGS}} \|x^{t+1} - x^t\|^2 \leq \|x^{t+1} - x^t\|^2_{J^t}\) where \(\sigma_{\text{min}} J^t = \epsilon\). By selecting \(\epsilon > \frac{M_f}{2}\), the following holds:

\[
\|x^{t+1} - x^t\|^2_{J^t} - \frac{M_f}{\sigma_{\text{min}} J^t} I \geq \sigma_{\text{min}} J^t \|x^{t+1} - x^t\|^2_{J^t}.
\]

We denote \(\delta = \frac{\sigma_{\text{min}} - \frac{M_f}{2}}{\sigma_{\text{min}}}\) and since \(\delta < 1\), it holds that

\[
2\mu_z \|z^{t+1} - z^t\|^2 + \mu_\theta \|\theta^{t+1} - \theta^t\|^2 + \frac{\mu_\theta}{\mu_\lambda} \|\lambda^{t+1} - \lambda^t\|^2 \geq \delta \left( \frac{M_f}{\sigma_{\text{min}}} \|x^{t+1} - x^t\|^2_{J^t} \right).
\]

Therefore, (62) can be rewritten as:

\[
\|v^{t+1}_\alpha - v^*_\alpha\|^2_{G^t} \leq \|v^t_\alpha - v^*_\alpha\|^2_{G^t} - \delta \|v^{t+1}_\alpha - v^t_\alpha\|^2_{G^t}.
\]

Since (64) shows that \(\|v^t_\alpha - v^*_\alpha\|^2_{G^t}\) is monotonically decreasing, it is therefore convergent. We proceed to show Part (ii).

Part (ii): Recall (50) and after rearranging, we obtain:

\[
H^t(x^{t+1} - x^t) = - \{ \nabla F(x^t) + E_s^\top \alpha^t + SL^t + \frac{\mu_\lambda}{2} L_s x^t + \mu_\theta S (S^\top x^t - \theta^t) \}.
\]
Since \( H^t = J^t + \mu z D + \epsilon I + \mu \theta SS^T \) as in (15), an upper bound \( H^t \preceq \bar{M} I \) can be obtained by using (25) and (27):

\[
\bar{M}_{\text{Gradient}} = \mu_z d_{\max} + \epsilon + \mu \theta,
\]

\[
\bar{M}_{\text{Newton}} = M_f + \mu_z d_{\max} + \epsilon + \mu \theta, \quad \bar{M}_{\text{BFGS}} = \psi,
\]

where \( d_{\max} = \max_i |N_i| \) denotes the maximum degree. Therefore, the following holds:

\[
\bar{M}^2 \| x^{t+1} - x^t \|^2 \geq \| x^{t+1} - x^t \|_{(H^t)^2}^2.
\]

(66)

We proceed to establish a lower bound for \( \| x^{t+1} - x^t \|_{\bar{J}}^2 \):

\[
\| x^{t+1} - x^t \|_{\bar{J}}^2 \geq \sigma_{\min}^2 \| x^{t+1} - x^t \|^2 \geq \sigma_{\min}^2 \| x^{t+1} - x^t \|^2_{(H^t)^2}
\]

\[
= \frac{\sigma_{\min}^2}{M^2} \left( \frac{1}{\rho} \| \nabla F(x^t) + E_s^T \alpha^t + S \lambda^t \| - \frac{1}{\rho-1} \| \frac{\mu_s}{2} L_s x^t + \mu \theta (S^T x^t - \theta^t) \| \right)^2
\]

\[
\geq \frac{\sigma_{\min}^2}{M^2} \left( \frac{1}{\rho} \| \nabla F(x^t) + E_s^T \alpha^t + S \lambda^t \|^2 - \frac{2}{\rho-1} \frac{\mu_s}{2} L_s x^t \|^2 - \frac{2}{\rho-1} \| \mu \theta (x^t_i - \theta^t) \|^2 \right),
\]

(67)

where (i) follows from (66); (ii) follows from (65); (iii) follows from \((a+b)^2 \geq \frac{1}{\rho} a^2 - \frac{1}{\rho-1} b^2\) for any \( \rho > 1 \); (iv) follows from \(- (a+b)^2 \geq -2(a^2 + b^2)\). Also note that \( \| \mu \theta S(S^T x^t - \theta^t) \| = \| \mu \theta (x^t_i - \theta^t) \| \) by definition of \( S = s_t \otimes I_d \) being the selection matrix. Further observe that the following holds due to dual updates (14c) and (14d): \( \alpha^{t+1} - \alpha^t = \frac{\mu_s}{2} E_s x^{t+1}, \lambda^{t+1} - \lambda^t = \mu \theta (S^T x^{t+1} - \theta^{t+1}) \). Therefore, we obtain the following:

\[
\frac{2}{\mu_z} \| \alpha^{t+1} - \alpha^t \|^2 = \frac{\mu_s}{2} \| E_s x^{t+1} \|^2 = \frac{\mu_s}{2} \| x^{t+1} \|_{L_s}^2,
\]

(68)

\[
\frac{1}{\mu \theta} \| \lambda^{t+1} - \lambda^t \|^2 = \mu \theta \| x^{t+1} - \theta^{t+1} \|^2,
\]

(69)

By denoting the maximum eigenvalue of \( L_s \) as \( \sigma_{\max}^2 \) and selecting \( \rho - 1 > \sigma_{\max}^2 \), we obtain:

\[
\frac{\sigma_{\min}^2}{M^2} \frac{2}{\rho-1} \frac{\mu_s}{2} L_s x^t \|^2 \leq \frac{\sigma_{\min}^2}{2M} \frac{\mu_s}{\sigma_{\max}^2} \| x^t \|_{(L_s)^2}^2 \leq \frac{\sigma_{\min}^2}{2M} \| x^t \|_{L_s}^2.
\]

(70)
Recall the definition (29). We establish (30) as follows:

$$\begin{align*}
&\frac{1}{T} \frac{\mu_T}{2} \left\| x^1 \right\|_{L_s}^2 + \frac{\mu_T}{T} \left\| x_i - \theta \right\|_2^2 + \frac{1}{T} \sum_{t=1}^{T} \left\| v_{\alpha \alpha}^{t+1} - v_{\alpha \alpha}^t \right\|_{G_t}^2 \\
= &\frac{1}{T} \frac{\mu_T}{2} \left\| x^1 \right\|_{L_s}^2 + \frac{\mu_T}{T} \left\| x_i - \theta \right\|_2^2 + \frac{1}{T} \sum_{t=1}^{T} \left( \left\| x^{t+1} - x^t \right\|_{G_t}^2 + 2\mu_T \left\| z^{t+1} - z^t \right\|_2^2 + \mu_T \left\| \theta^{t+1} - \theta^t \right\|_2^2 \\
&+ \frac{2\mu_T}{\mu_T} \left\| \alpha^{t+1} - \alpha^t \right\|_2^2 + \frac{1}{\mu_T} \left\| \mu^{t+1} - \mu^t \right\|_2^2 \right) \\
\geq &\frac{1}{T} \frac{\mu_T}{2} \| x^{T+1} \|_{L_s}^2 + \frac{\mu_T}{T} \left\| x_i^{T+1} - \theta^{T+1} \right\|_2^2 + \frac{1}{T} \sum_{t=1}^{T} \left( \frac{\mu_T}{T} \left\| \nabla F(x^t) + E_s^T \alpha^t + S\lambda^t \right\|_2^2 \\
&+ 2\mu_T \left\| x^{t+1} - x^t \right\|_2^2 + \mu_T \left\| \theta^{t+1} - \theta^t \right\|_2^2 + \left( \frac{\mu_T}{T} - \frac{\mu_T \mu_T}{2\mu_T} \right) \left\| x^t \right\|_{L_s}^2 + \left( \mu_T - \frac{2\mu_T \mu_T}{\mu_T} \right) \left\| x^t - \theta^t \right\|_2^2 \right) \right)
\end{align*}$$

where (i) follows from substituting (67)-(70). All coefficients are ensured to be positive by selecting: \( \mu_T \epsilon < \psi^2 \), and \( \rho > \max \left\{ \frac{2\sigma_T \min \mu_T}{\mu_T}, \sigma_{L_s} \right\} + 1 \), where \( \sigma_T = \epsilon \).

**Proof of Corollary** Following Theorem and standard analysis techniques in [51] and [52], we obtain that \( \| v_{\alpha}^{1} - v_{\alpha}^* \| \rightarrow 0 \) as \( t \rightarrow \infty \). After taking telescoping sum from \( t = 1 \) to \( \infty \) on both sides of (64), we obtain:

$$\delta \sum_{t=1}^{\infty} \left\| v_{\alpha}^{t+1} - v_{\alpha}^t \right\|^2_{G_t} \leq \left\| v_{\alpha}^1 - v_{\alpha}^* \right\|^2_{G_t},$$

i.e., \( \sum_{t=1}^{\infty} \left\| v_{\alpha}^{t+1} - v_{\alpha}^t \right\|^2_{G_t} \) is bounded. Define \( b^T := \frac{1}{T} \sum_{t=1}^{T} \left\| v_{\alpha}^{t+1} - v_{\alpha}^t \right\|^2_{G_t} \). Then \( \lim_{T \rightarrow \infty} Tb^T = \lim_{T \rightarrow \infty} \sum_{t=1}^{T} \left\| v_{\alpha}^{t+1} - v_{\alpha}^t \right\|^2_{G_t} < \infty \). Therefore, \( b^T = \frac{1}{T} \sum_{t=1}^{T} \left\| v_{\alpha}^{t+1} - v_{\alpha}^t \right\|^2_{G_t} = \mathcal{O}(\frac{1}{T}) \). By (30), each term in (31) is of order \( \mathcal{O}(\frac{1}{T}) \).

**APPENDIX C**

**Proof of Lemma** Recall the definition of \( e^t \) in (28): \( e^t = \nabla F(x^t) - \nabla F(x^{t+1}) + J^t(x^{t+1} - x^t) \).

By applying the triangle and Cauchy-Schwartz inequality, we obtain:

$$\| e^t \| \leq \| \nabla F(x^t) - \nabla F(x^{t+1}) \| + \| J^t \| \| x^{t+1} - x^t \|.$$

(71)

In the case of gradient updates, \( J^t = 0 \). Therefore, \( \| e^t \| \leq \| \nabla F(x^t) - \nabla F(x^{t+1}) \| \leq M_f \| x^{t+1} - x^t \| \), where the last inequality follows from Assumption. Setting \( \tau^t_{Gradient} = M_f \), we obtain (33a). In the case of Newton updates, \( J^t = \nabla^2 F(x^t) \). By Assumption and (71), we obtain:

$$\| e^t \| \leq 2M_f \| x^{t+1} - x^t \|.$$

(72)
Moreover, by the fundamental theorem of calculus, $\nabla F(x^{t+1}) - \nabla F(x^t)$ can be written as:

$$\nabla F(x^{t+1}) - \nabla F(x^t) = \int_0^1 \nabla^2 F(sx^{t+1} + (1-s)x^t)(x^{t+1} - x^t)ds.$$  

(73)

By adding and subtracting $\int_0^1 \nabla^2 F(x^t)(x^{t+1} - x^t)ds$ on the right-hand side of (73), we further obtain:

$$\nabla F(x^{t+1}) - \nabla F(x^t) = \int_0^1 \nabla^2 F(x^t)(x^{t+1} - x^t)ds$$

$$+ \int_0^1 (\nabla^2 F(sx^{t+1} + (1-s)x^t) - \nabla^2 F(x^t))(x^{t+1} - x^t)ds.$$  

Since the integrand of the first term is constant with respect to $s$, it holds that:

$$\left\| \nabla F(x^{t+1}) - \nabla F(x^t) - \nabla^2 F(x^t)(x^{t+1} - x^t) \right\|$$

$$= \left\| \int_0^1 (\nabla^2 F(sx^{t+1} + (1-s)x^t) - \nabla^2 F(x^t))(x^{t+1} - x^t)ds \right\|$$

$$\leq \int_0^1 \left\| \nabla^2 F(sx^{t+1} + (1-s)x^t) - \nabla^2 F(x^t) \right\| \left\| x^{t+1} - x^t \right\| ds$$

$$\leq \int_0^1 sL_f \left\| x^{t+1} - x^t \right\|^2 ds$$

$$= \frac{L_f}{2} \left\| x^{t+1} - x^t \right\|^2.$$  

(74)

Note that in the case of Newton updates, $\| e^t \| = \| \nabla F(x^{t+1}) - \nabla F(x^t) - \nabla^2 F(x^t)(x^{t+1} - x^t) \|$. By combining (72) and (74), we obtain: $\| e^t \| \leq \tau^t_{\text{Newton}} \| x^{t+1} - x^t \|$, where $\tau^t_{\text{Newton}}$ is defined in (33b). We proceed to establish (33c). Recall the definition of $J^t_{\text{BFGS}}$ in (20):

$$J^t_{\text{BFGS}} = H^t_{\text{BFGS}} - \mu_z D - \mu_\theta S S^\top - \epsilon I_{md}.$$  

(75)

Therefore, $H^{t+1}$ (suppressing the subscript BFGS) satisfies the secant condition: $H^{t+1} s^t = q^t$, where $\{q^t, s^t\}$ as per the definition in (19) can be written as: $s^t = x^{t+1} - x^t, q^t = \nabla F(x^{t+1}) - \nabla F(x^t) + (\mu_z D + \mu_\theta S S^\top + \epsilon I) s^t$. From the secant condition, it holds that: $\nabla F(x^t) - \nabla F(x^{t+1}) = -(H^{t+1} - \mu_z D - \mu_\theta S S^\top - \epsilon I)(x^{t+1} - x^t)$. Using (75) and the expression for $\nabla F(x^t) - \nabla F(x^{t+1})$ into (28), we obtain: $\| e^t \| = \| (H^t - H^{t+1})(x^{t+1} - x^t) \| \leq \| H^t - H^{t+1} \| \| x^{t+1} - x^t \|$. Denoting $\tau^t_{\text{BFGS}} = \| H^t - H^{t+1} \|$ and using (27), we obtain (33c).  

The following Lemma that will be useful for establishing Theorem 2.
Lemma 5. Recall $C := \begin{bmatrix} E_s \\ S^T \end{bmatrix}$ and $\phi^t = E_s^T \alpha^t$ in (14). Denote the smallest positive eigenvalue of $CC^T$ as $\sigma_{\min}^+$ and consider the unique dual optimal pair $(\alpha^*, \lambda^*)$ that lies in the column space of $C$ as established in Lemma 2. The following holds:

$$\sigma_{\min}^+ \left( \| \alpha^{t+1} - \alpha^* \|^2 + \| \lambda^{t+1} - \lambda^* \|^2 \right) \leq \| E_s^T (\alpha^{t+1} - \alpha^*) + S(\lambda^{t+1} - \lambda^*) \|^2. \quad (76)$$

Proof: We proceed by showing that $[\alpha^{t+1}; \lambda^{t+1}]$ lies in $\text{col}(C)$. We rewrite dual updates (14c)–(14d) as:

$$\begin{bmatrix} \alpha^{t+1} \\ \lambda^{t+1} \end{bmatrix} = \begin{bmatrix} \alpha^t \\ \lambda^t \end{bmatrix} + \begin{bmatrix} \mu_s E_s \\ \mu \mu S^T \end{bmatrix} x^{t+1} - \begin{bmatrix} 0 \\ \mu \mu I_d \end{bmatrix} \theta^{t+1}. \quad \text{We show that the column space of}$$

$$M := \begin{bmatrix} 0 \\ \mu \mu I_d \end{bmatrix} \text{ belongs in the column space of } N := \begin{bmatrix} \mu_s E_s \\ \mu \mu S^T \end{bmatrix}. \quad \text{Consider fixed } r^x \in \mathbb{R}^d. \text{ Let}$$

$$r^y \in \mathbb{R}^{md} \text{ such that each sub-vector component } r^y_i = r^x, \text{ i.e., } r^y = [r^x, \ldots; r^x]. \text{ Then it holds that}$$

$$\begin{bmatrix} \mu_s E_s \\ \mu \mu S^T \end{bmatrix} r^y = \begin{bmatrix} 0 \\ \mu \mu I_d \end{bmatrix} r^x, \quad \text{which shows } \text{col}(M) \subset \text{col}(N). \quad \text{By choosing } \mu_z = 2 \mu, \text{ we conclude that} \quad [\alpha^{t+1} - \alpha^*; \lambda^{t+1} - \lambda^*] \text{ lies in the column space of } C, \text{ which establishes} \quad (76).$$

Proof of Theorem 2. Using Lemma 3 we obtain:

$$\nabla F(x^{t+1}) - \nabla F(x^*) = - \left( E_s^T (\alpha^{t+1} - \alpha^*) + \epsilon (x^{t+1} - x^t) + S(\lambda^{t+1} - \lambda^* + \mu \theta (\theta^{t+1} - \theta^t)) + \epsilon^t \right.$$  

$$
+ \mu \mu E_a^T (z^{t+1} - z^t)), \quad (77)
$$

Since $F(x)$ is strongly convex with Lipschitz continuous gradient, the following inequality holds:

$$\frac{m_f M_f}{m_f + M_f} \| x^{t+1} - x^* \|^2 + \frac{1}{m_f + M_f} \| \nabla F(x^{t+1}) - \nabla F(x^*) \|^2$$

$$\leq (x^{t+1} - x^*)^T (\nabla F(x^{t+1}) - \nabla F(x^*))$$

$$\leq - (x^{t+1} - x^*)^T \epsilon^t - \epsilon (x^{t+1} - x^*)^T (x^{t+1} - x^t) - (x^{t+1} - x^*) E_s^T (\alpha^{t+1} - \alpha^*)$$

$$- (x^{t+1} - x^*)^T S (\lambda^{t+1} - \lambda^* + \mu \theta (\theta^{t+1} - \theta^t)) - \mu \mu (x^{t+1} - x^*)^T E_a^T (z^{t+1} - z^t),$$

where the last inequality follows from substituting the expression of $\nabla F(x^{t+1}) - \nabla F(x^*)$ in
Following the same techniques used in deriving (57)-(60), we obtain
\[
\frac{2m_t M_f}{m_f + M_f} \left\| x^{t+1} - x^* \right\|^2 + \frac{2}{m_f + M_f} \left\| \nabla F(x^{t+1}) - \nabla F(x^*) \right\|^2 \\
\leq \epsilon \left( \left\| x^t - x^* \right\|^2 - \left\| x^{t+1} - x^* \right\|^2 - \left\| x^{t+1} - x^t \right\|^2 \right) \\
+ 2\mu_z \left( \left\| z^t - z^* \right\|^2 - \left\| z^{t+1} - z^* \right\|^2 - \left\| z^{t+1} - z^t \right\|^2 \right) \\
+ \frac{1}{\mu_e} \left( \left\| \lambda^t - \lambda^* \right\|^2 - \left\| \lambda^{t+1} - \lambda^* \right\|^2 - \left\| \lambda^{t+1} - \lambda^t \right\|^2 \right) \\
+ \mu_\theta \left( \left\| \theta^t - \theta^* \right\|^2 - \left\| \theta^{t+1} - \theta^* \right\|^2 - \left\| \theta^{t+1} - \theta^t \right\|^2 \right) \\
+ \frac{2}{\mu_z} \left[ \left\| \alpha^{t+1} - \alpha^* \right\|^2 - \left\| \alpha^{t+1} - \alpha^* \right\|^2 - \left\| \alpha^{t+1} - \alpha^t \right\|^2 \right] - 2(x^{t+1} - x^*)^T e^t.
\]
where the last equality follows from the definition (35). Therefore, the following holds:
\[
\frac{2m_t M_f}{m_f + M_f} \left\| x^{t+1} - x^* \right\|^2 + \frac{2}{m_f + M_f} \left\| \nabla F(x^{t+1}) - \nabla F(x^*) \right\|^2 + \left\| v^{t+1}_\alpha - v^*_\alpha \right\|^2_H + 2(x^{t+1} - x^*)^T e^t \\
\leq \left\| v^{t+1}_\alpha - v^*_\alpha \right\|^2_H - \left\| v^{t+1}_\alpha - v^*_\alpha \right\|^2_H.
\]
To establish linear convergence, we need to show the following holds for some \( \eta > 0 \):
\[
\eta \left\| v^{t+1}_\alpha - v^*_\alpha \right\|^2_H \leq \left\| v^{t+1}_\alpha - v^*_\alpha \right\|^2_H - \left\| v^{t+1}_\alpha - v^*_\alpha \right\|^2_H.
\]
By the definition in (35), we expand the expression of \( \eta \left\| v^{t+1}_\alpha - v^*_\alpha \right\|^2_H \) as follows:
\[
\eta \left\| v^{t+1}_\alpha - v^*_\alpha \right\|^2_H = \eta \left( \epsilon \left\| x^{t+1} - x^* \right\|^2 + \frac{2}{\mu_z} \left\| z^{t+1} - z^* \right\|^2 + \frac{2}{\mu_z} \left\| \alpha^{t+1} - \alpha^* \right\|^2 + \mu_\theta \left\| \theta^{t+1} - \theta^* \right\|^2 \\
+ \frac{1}{\mu_e} \left\| \lambda^{t+1} - \lambda^* \right\|^2 \right).
\]
We proceed to establish an upper bound for each component of (81). From Lemma 3, the following holds:
\[
E_s^T (\alpha^{t+1} - \alpha^*) + S(\lambda^{t+1} - \lambda^*) = - \left\{ \nabla F(x^{t+1}) - \nabla F(x^*) + \epsilon (x^{t+1} - x^*) + \mu_z E_u^T (z^{t+1} - z^*) + \mu_\theta (\theta^{t+1} - \theta^*) + e^t \right\}.
\]
Then it holds that:
\[
\sigma_{\min}^T \left( \left\| \alpha^{t+1} - \alpha^* \right\|^2 + \left\| \lambda^{t+1} - \lambda^* \right\|^2 \right) \\
\leq \left\| E_s^T (\alpha^{t+1} - \alpha^*) + S(\lambda^{t+1} - \lambda^*) \right\|^2 \\
\leq 5 \left( \left\| \nabla F(x^{t+1}) - \nabla F(x^*) \right\|^2 + \epsilon^2 \left\| x^{t+1} - x^* \right\|^2 + \mu_\theta \left\| \theta^{t+1} - \theta^* \right\|^2 \\
+ \left\| e^t \right\|^2 + \sigma_{\max} E_u^T (z^{t+1} - z^*) \right)^2 \right),
\]
Recalling that we have selected $\mu_\varepsilon = 2\mu_\theta$, we obtain:

$$\frac{2}{\mu_\varepsilon} \|\alpha^{t+1} - \alpha^*\|^2 + \frac{1}{\mu_\theta} \|\lambda^{t+1} - \lambda^*\|^2 = \frac{1}{\mu_\theta} \left( \|\alpha^{t+1} - \alpha^*\|^2 + \|\lambda^{t+1} - \lambda^*\|^2 \right) \leq \frac{5}{\mu_\theta \sigma_{\min}} \left( \|\nabla F(x^{t+1}) - \nabla F(x^*)\|^2 + \epsilon^2 \|x^{t+1} - x^t\|^2 + \mu_\theta^2 \|\theta^{t+1} - \theta^t\|^2 + \|e^t\|^2 + \sigma_{\max}^L \mu_\varepsilon^2 \|z^{t+1} - z^t\|^2 \right),$$

(83)

where (i) follows from dividing (82) by $\sigma_{\min}^+$ on both sides and substituting. Note that since $z^{t+1} - z^* = \frac{1}{2} E_u(x^{t+1} - x^*)$, it holds that $2\mu_\varepsilon \|z^{t+1} - z^*\|^2 \leq \frac{\mu_\varepsilon \sigma_{\max}^L}{2} \|x^{t+1} - x^*\|^2$. Using the upper bound for $2\mu_\varepsilon \|z^{t+1} - z^*\|^2$, the inequality (83), and $\mu_\theta \|\theta^{t+1} - \theta^*\|^2 \leq 2\mu_\theta \|x^{t+1} - x^*\|^2 + \frac{2}{\mu_\theta} \|\lambda^{t+1} - \lambda^*\|^2$ from (14d) and KKTd, we obtain an upper bound for (81) as:

$$\eta \|v^{t+1}_\alpha - v^*_\alpha\|^2_H \leq \eta \left\{ \frac{5}{\mu_\theta \sigma_{\min}} \left( \|\nabla F(x^{t+1}) - \nabla F(x^*)\|^2 + \epsilon^2 \|x^{t+1} - x^t\|^2 + \mu_\theta^2 \|\theta^{t+1} - \theta^t\|^2 + \|e^t\|^2 + \sigma_{\max}^L \mu_\varepsilon^2 \|z^{t+1} - z^t\|^2 \right) + \frac{2}{\mu_\theta} \|\lambda^{t+1} - \lambda^*\|^2 + (\epsilon + 2\mu_\theta + \frac{\mu_\varepsilon \sigma_{\max}^L}{2}) \|x^{t+1} - x^*\|^2 \right\}. \quad (84)$$

Recall that the right-hand side of (80) is lower bounded as in (79). Using (84), we prove the following to establish (80):

$$\eta \left\{ \frac{5}{\mu_\theta \sigma_{\min}} \left( \|\nabla F(x^{t+1}) - \nabla F(x^*)\|^2 + \epsilon^2 \|x^{t+1} - x^t\|^2 + \mu_\theta^2 \|\theta^{t+1} - \theta^t\|^2 + \|e^t\|^2 + \sigma_{\max}^L \mu_\varepsilon^2 \|z^{t+1} - z^t\|^2 \right) + \frac{2}{\mu_\theta} \|\lambda^{t+1} - \lambda^*\|^2 + (\epsilon + 2\mu_\theta + \frac{\mu_\varepsilon \sigma_{\max}^L}{2}) \|x^{t+1} - x^*\|^2 \right\} \leq \frac{2m_f M_f}{m_f + M_f} \|x^{t+1} - x^*\|^2_H + \frac{2}{m_f + M_f} \|\nabla F(x^{t+1}) - \nabla F(x^*)\|^2 + \|v^{t+1}_\alpha - v^*_\alpha\|^2_H + 2(x^{t+1} - x^*)^T e^t, \quad (85)$$

Note that $-\zeta \|e^t\|^2 - \frac{1}{\zeta} \|x^{t+1} - x^*\|^2 \leq 2(x^{t+1} - x^*)^T e^t$ holds for any $\zeta > 0$. To prove (85), it is therefore sufficient to show:

$$\zeta \|x^{t+1} - x^*\|^2 + \eta \left\{ \frac{5}{\mu_\theta \sigma_{\min}} \left( \|\nabla F(x^{t+1}) - \nabla F(x^*)\|^2 + (\epsilon^2 + 2\mu_\varepsilon) \|x^{t+1} - x^t\|^2 \right) + \mu_\theta^2 \|\theta^{t+1} - \theta^t\|^2 + \sigma_{\max}^L \mu_\varepsilon^2 \|z^{t+1} - z^t\|^2 \right\} \leq \left( \frac{2m_f M_f - 1}{\zeta} \right) \|x^{t+1} - x^*\|^2 + \epsilon \|x^{t+1} - x^t\|^2 + 2\mu_\varepsilon \|z^{t+1} - z^t\|^2 + \frac{2}{\mu_\varepsilon} \|\alpha^{t+1} - \alpha^*\|^2 + \mu_\theta \|\theta^{t+1} - \theta^t\|^2 + \frac{1}{\mu_\theta} \|\lambda^{t+1} - \lambda^*\|^2 + \frac{2}{m_f + M_f} \|\nabla F(x^{t+1}) - \nabla F(x^*)\|^2 \quad (86)$$
where we have used $\|e^t\|^2 \leq (\tau^t)^2\|x^{t+1} - x^t\|^2$ from Lemma 4. Establishing (86) amounts to ensuring the coefficient of each term in the left-hand side is bounded by the coefficient of the corresponding term on the right-hand side. By selecting $\eta$ as in (36), we establish (86). Therefore, the inequality (80) holds, which equivalently establishes the linear convergence rate.

Proof of Theorem 3  The proof proceeds as follows:

$$
\|v_{t+1}^\alpha - v^*_\alpha\|_{\mathcal{H}\Omega^{-1}}^2 = \|v_{t+1}^\alpha - v^*_\alpha\|_{\mathcal{H}\Omega^{-1}}^2 + \|Tv_{t+1}^\alpha - v_{t+1}^\alpha\|_{\mathcal{H}}^2 \\
= \|v_t^\alpha - v^*_\alpha\|_{\mathcal{H}\Omega^{-1}}^2 + 2(v_t^\alpha - v^*_\alpha)^\top \mathcal{H}\Omega^{-1}\Omega_{t+1}^\alphaTv_t^\alpha - v_t^\alpha) \\
+ (Tv_t^\alpha - v_t^\alpha)^\top \Omega_{t+1}^\alpha \Omega_{t+1}^{-1}(Tv_t^\alpha - v_t^\alpha),
$$

(87)

Since $\Omega_{t+1}^\alpha, \Omega_{t}^{-1}$, and $\mathcal{H}$ are all diagonal matrices, they commute with each other. Moreover, since each sub-block of $\Omega_{t+1}^\alpha$ is $I_d$ or 0, it holds that $\Omega_{t+1}^\alpha\Omega_{t}^{-1} = \Omega_{t}^{t+1}$. After taking conditional expectation on both sides of (87), we obtain:

$$
\mathbb{E}_t \left[\|v_{t+1}^\alpha - v^*_\alpha\|_{\mathcal{H}\Omega^{-1}}^2\right] = \|v_t^\alpha - v^*_\alpha\|_{\mathcal{H}\Omega^{-1}}^2 + \|Tv_t^\alpha - v_{t}^\alpha\|_{\mathcal{H}}^2 \\
+ 2(v_t^\alpha - v^*_\alpha)^\top \mathcal{H}(Tv_t^\alpha - v_t^\alpha) \\
\leq \|v_t^\alpha - v^*_\alpha\|_{\mathcal{H}\Omega^{-1}}^2 - \frac{n}{1+\eta}\|Tv_t^\alpha - v_{t}^\alpha\|_{\mathcal{H}}^2 \\
\leq \left(1 - \frac{\eta}{1+\eta}\right)\|v_t^\alpha - v^*_\alpha\|_{\mathcal{H}\Omega^{-1}}^2,
$$

(88)

where (i) follows from the fact that $2(v_t^\alpha - v^*_\alpha)^\top \mathcal{H}(Tv_t^\alpha - v_t^\alpha) + \|Tv_t^\alpha - v_{t}^\alpha\|_{\mathcal{H}}^2 \leq -\frac{n}{1+\eta}\|Tv_t^\alpha - v_{t}^\alpha\|_{\mathcal{H}}^2$ holds for any $v_t^\alpha \in \mathbb{R}^{(m+2n+2)d}$ using Theorem 2; (ii) follows from $\frac{n}{1+\eta}\|Tv_t^\alpha - v_{t}^\alpha\|_{\mathcal{H}}^2 \geq \frac{\eta}{1+\eta}\|Tv_t^\alpha - v_{t}^\alpha\|_{\mathcal{H}\Omega^{-1}}^2$.

Proof of Corollary 2  We first distribute each $\alpha_k, k \in [n]$, to each edge and label agents and edges with an arbitrary order. For each edge $\mathcal{E}_k$, we write $\mathcal{E}_k = (i,j)$ with the convention $i < j$. Then for each agent $i$, we can divide the incident edges to two groups: $\mathcal{P}_i = \{k : \mathcal{E}_k = (i,j), j \in \mathcal{N}_i\}$ and $\mathcal{S}_i = \{k : \mathcal{E}_k = (j,i), j \in \mathcal{N}_i\}$. Consider the activation scheme using $\Omega_{t+1}$. Recall $\alpha_{t+1}^\alpha = \alpha_{t}^\alpha + \frac{\mu}{2}(x_{t+1}^i - x_{t+1}^j)$. Then the dual updates are described by:

$$
\phi_{i_i}^{t+1} = \phi_{i_i}^t + \frac{\mu}{2}\sum_{j \in \mathcal{N}_i}(x_{t+1}^i - x_{t+1}^j) \\
= \phi_{i_i}^t + \sum_{k \in \mathcal{P}_i}(\alpha_{t+1}^\alpha - \alpha_{t}^\alpha) + \sum_{k \in \mathcal{S}_i}(\alpha_{t+1}^\alpha - \alpha_{t}^\alpha).
$$
Therefore, if $X_{ii}^{t+1} = I_d$, then $Y_{kk}^{t+1} = I_d$ for $k \in P_i \cup S_i$ for the corresponding $\Omega_{\alpha}^{t+1}$, i.e., all incident edges are active. It can be verified that we can map $X^{t+1}$ to $Y^{t+1}$ as:

$$Y^{t+1} = \text{Blkdiag} \left( \frac{E_u X^{t+1}(1 \otimes I_d)}{2} \right),$$

where $[\cdot]$ is the entry-wise ceiling operation and $1 \in \mathbb{R}^m$ is the all one vector. To show $\mathbb{E}^t[\Omega_{\alpha}^{t+1}] \succ 0$, we only need to show $\mathbb{E}^t[Y^{t+1}] \succ 0$, which amounts to showing that $\mathbb{E}^t \left[ \frac{E_u X^{t+1}(1 \otimes I_d)}{2} \right]_{k,k} \in \mathbb{R}^{d \times d}$, $k \in [n]$, is positive definite. We can explicitly express each block as:

$$\left[ \frac{E_u X^{t+1}(1 \otimes I_d)}{2} \right]_{k,k} = \left[ \frac{X_{ii}^{t+1} + X_{jj}^{t+1}}{2} \right],$$

where $(i,j) \in \mathcal{E}_k$. Therefore,

$$\mathbb{E}^t \left[ \frac{E_u X^{t+1}(1 \otimes I_d)}{2} \right]_{k,k} = \mathbb{E}^t \left[ \frac{X_{ii}^{t+1} + X_{jj}^{t+1}}{2} \right] \succ 0,$$

which shows that $\mathbb{E}^t[Y^{t+1}] \succ 0$.

---

**References**

[1] A. Nedić and J. Liu, “Distributed optimization for control,” *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 1, no. 1, pp. 77–103, 2018.

[2] N. M. Freris, H. Kowshik, and P. R. Kumar, “Fundamentals of large sensor networks: Connectivity, capacity, clocks, and computation,” *Proceedings of the IEEE*, vol. 98, no. 11, pp. 1828–1846, 2010.

[3] T. Huang, N. M. Freris, P. R. Kumar, and L. Xie, “A synchrophasor data-driven method for forced oscillation localization under resonance conditions,” *IEEE Transactions on Power Systems*, vol. 35, no. 5, pp. 3927–3939, 2020.

[4] A. Nedic, “Distributed gradient methods for convex machine learning problems in networks: Distributed optimization,” *IEEE Signal Processing Magazine*, vol. 37, no. 3, pp. 92–101, 2020.

[5] F. Lin, M. Fardad, and M. R. Jovanović, “Design of optimal sparse feedback gains via the alternating direction method of multipliers,” *IEEE Transactions on Automatic Control*, vol. 58, no. 9, pp. 2426–2431, 2013.

[6] C. Saunders, A. Gammerman, and V. Vovk, “Ridge regression learning algorithm in dual variables,” in *ICML*, 1998, p. 515–521.

[7] A. Nedic and A. Ozdaglar, “Distributed subgradient methods for multi-agent optimization,” *IEEE Transactions on Automatic Control*, vol. 54, no. 1, pp. 48–61, 2009.

[8] 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.

[9] ——, “A proximal gradient algorithm for decentralized composite optimization,” *IEEE Transactions on Signal Processing*, vol. 63, no. 22, pp. 6013–6023, 2015.

[10] G. Qu and N. Li, “Harnessing smoothness to accelerate distributed optimization,” *IEEE Transactions on Control of Network Systems*, vol. 5, no. 3, pp. 1245–1260, 2018.

[11] S. A. Alghunaim, K. Yuan, and A. H. Sayed, “A linearly convergent proximal gradient algorithm for decentralized optimization,” in *NeurIPS*, 2019.
[12] K. Yuan, B. Ying, X. Zhao, and A. H. Sayed, “Exact diffusion for distributed optimization and learning—part i: Algorithm development,” IEEE Transactions on Signal Processing, vol. 67, no. 3, pp. 708–723, 2019.

[13] J. Xu, Y. Tian, Y. Sun, and G. Scutari, “Distributed algorithms for composite optimization: Unified framework and convergence analysis,” IEEE Transactions on Signal Processing, vol. 69, pp. 3555–3570, 2021.

[14] E. Wei, A. Ozdaglar, and A. Jadbabaie, “A distributed Newton method for network utility maximization—i: Algorithm,” IEEE Transactions on Automatic Control, vol. 58, no. 9, pp. 2162–2175, 2013.

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

[16] A. Mokhtari, Q. Ling, and A. Ribeiro, “Network Newton distributed optimization methods,” IEEE Transactions on Signal Processing, vol. 65, no. 1, pp. 146–161, 2017.

[17] F. Mansoori and E. Wei, “A fast distributed asynchronous Newton-based optimization algorithm,” IEEE Transactions on Automatic Control, vol. 65, no. 7, pp. 2769–2784, 2020.

[18] J. Nocedal and S. J. Wright, Numerical Optimization. Springer, 2006.

[19] M. R. Hestenes, “Multiplier and gradient methods,” Journal of Optimization Theory and Applications, vol. 4, pp. 303–320, 1969.

[20] D. P. Bertsekas and J. N. Tsitsiklis, Parallel and Distributed Computation: Numerical Methods. Prentice-Hall, Inc., 1989.

[21] S. P. Boyd, N. Parikh, E. K. wah Chu, B. Peleato, and J. Eckstein, “Distributed optimization and statistical learning via the alternating direction method of multipliers,” Foundations and Trends in Machine Learning, vol. 3, pp. 1–122, 2011.

[22] 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.

[23] J. C. Duchi, A. Agarwal, and M. J. Wainwright, “Dual averaging for distributed optimization: convergence analysis and network scaling,” IEEE Transactions on Automatic Control, vol. 57, no. 3, pp. 592–606, 2012.

[24] E. Wei and A. Ozdaglar, “Distributed alternating direction method of multipliers,” in IEEE CDC, 2012, pp. 5445–5450.

[25] 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.

[26] M. Hong and Z. Q. Luo, “On the linear convergence of the alternating direction method of multipliers,” Mathematical Programming, vol. 162, pp. 165–199, 2017.

[27] P. Latafat, N. M. Freris, and P. Patrinos, “A new randomized block-coordinate primal-dual proximal algorithm for distributed optimization,” IEEE Transactions on Automatic Control, vol. 64, no. 10, pp. 4050–4065, 2019.

[28] D. Jakovetić, J. M. F. Moura, and J. Xavier, “Linear convergence rate of a class of distributed augmented lagrangian algorithms,” IEEE Transactions on Automatic Control, vol. 60, no. 4, pp. 922–936, 2015.

[29] 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.

[30] 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, 2015.

[31] A. Mokhtari, W. Shi, Q. Ling, and A. Ribeiro, “DQM: Decentralized quadratically approximated alternating direction method of multipliers,” IEEE Transactions on Signal Processing, vol. 64, no. 19, pp. 5158–5173, 2016.

[32] ——, “A decentralized second-order method with exact linear convergence rate for consensus optimization,” IEEE Transactions on Signal and Information Processing over Networks, vol. 2, no. 4, pp. 507–522, 2016.

[33] M. Eisen, A. Mokhtari, and A. Ribeiro, “A primal-dual quasi-Newton method for exact consensus optimization,” IEEE Transactions on Signal Processing, vol. 67, no. 23, pp. 5983–5997, 2019.
[34] D. Jakovetić, D. Bajović, J. Xavier, and J. M. F. Moura, “Primal–dual methods for large-scale and distributed convex optimization and data analytics,” Proceedings of the IEEE, vol. 108, no. 11, pp. 1923–1938, 2020.

[35] K.-D. Kim and P. R. Kumar, “Cyber–physical systems: A perspective at the centennial,” Proceedings of the IEEE, vol. 100, no. Special Centennial Issue, pp. 1287–1308, 2012.

[36] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y. Arcas, “Communication-Efficient Learning of Deep Networks from Decentralized Data,” in AISTATS, vol. 54. PMLR, 2017, pp. 1273–1282.

[37] T. Lin, S. Ma, and S. Zhang, “Global convergence of unmodified 3-block ADMM for a class of convex minimization problems,” Journal of Scientific Computing, vol. 76, no. 1, pp. 69–88, 2018.

[38] C. G. Broyden, “The convergence of a class of double-rank minimization algorithms I. general considerations,” IMA Journal of Applied Mathematics, vol. 6, no. 1, pp. 76–90, 1970.

[39] R. Fletcher, “A new approach to variable metric algorithms,” The Computer Journal, vol. 13, no. 3, pp. 317–322, 1970.

[40] D. Goldfarb, “A family of variable-metric methods derived by variational means,” Mathematics of Computation, pp. 23–26, 1970.

[41] D. F. Shanno, “Conditioning of quasi-Newton methods for function minimization,” Mathematics of Computation, vol. 24, no. 111, pp. 647–656, 1970.

[42] Y. Li, N. M. Freris, P. Voulgaris, and D. Stipanović, “D-SOP: Distributed second order proximal method for convex composite optimization,” in ACC, 2020, pp. 2844–2849.

[43] M. Eisen, A. Mokhtari, and A. Ribeiro, “Decentralized quasi-Newton methods,” IEEE Transactions on Signal Processing, vol. 65, no. 10, pp. 2613–2628, 2017.

[44] D. Bajović, D. Jakovetić, N. Krejić, and N. K. Jerinkić, “Newton-like method with diagonal correction for distributed optimization,” SIAM Journal on Optimization, vol. 27, no. 2, pp. 1171–1203, 2017.

[45] A. Nedic, “Asynchronous broadcast-based convex optimization over a network,” IEEE Transactions on Automatic Control, vol. 56, no. 6, pp. 1337–1351, 2011.

[46] A. Zouzias and N. M. Freris, “Randomized gossip algorithms for solving laplacian systems,” in ECC, 2015, pp. 1920–1925.

[47] Z. Peng, Y. Xu, M. Yan, and W. Yin, “Arock: An algorithmic framework for asynchronous parallel coordinate updates,” SIAM Journal on Scientific Computing, vol. 38, no. 5, pp. A2851–A2879, 2016.

[48] Y. Nesterov, Lectures on Convex Optimization. Springer Publishing Company, 2018.

[49] A. Mokhtari and A. Ribeiro, “RES: Regularized stochastic BFGS algorithm,” IEEE Transactions on Signal Processing, vol. 62, no. 23, pp. 6089–6104, 2014.

[50] N. M. Freris, S. R. Graham, and P. R. Kumar, “Fundamental limits on synchronizing clocks over networks,” IEEE Transactions on Automatic Control, vol. 56, no. 6, pp. 1352–1364, 2011.

[51] B. He, “A new method for a class of linear variational inequalities,” Mathematical Programming, p. 137–144, 1994.

[52] W. Deng, M.-J. Lai, Z. Peng, and W. Yin, “Parallel multi-block ADMM with o(1 / k) convergence,” Journal of Scientific Computing, p. 712–736, 2017.