

Communication Efficient Curvature Aided Primal-dual Algorithms for Decentralized Optimization

Yichuan Li, Petros G. Voulgaris, Dušan M. Stipanović, and Nikolaos M. Freris

Abstract—This paper presents a family of algorithms for decentralized 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 the 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 rates under convex objective functions with Lipschitz continuous gradients, as well as linear convergence rates when the local functions are further assumed to be strongly convex. Moreover, we explicitly characterize the acceleration due to curvature information. Last but not the least, we present an asynchronous implementation for the proposed algorithms, which removes the need for a central clock, with linear convergence rates established in expectation under strongly convex objectives. We ascertain the effectiveness of the proposed methods with numerical experiments on benchmark datasets.

Index Terms—Asynchronous algorithms, decentralized 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, wireless sensor networks, power grid management, and large-scale machine learning [1]–[5]. 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:

$$\underset{\hat{x} \in \mathbb{R}^d}{\text{minimize}} \quad \left\{ \sum_{i=1}^m f_i(\hat{x}) + g(\hat{x}) \right\}, \quad (1)$$

where each $f_i(\cdot) : \mathbb{R}^d \rightarrow \mathbb{R}$ is a convex and smooth function accessible only by agent i while $g(\cdot) : \mathbb{R}^d \rightarrow \mathbb{R}$ is a convex (possibly nonsmooth) regularizer. The inclusion of

the regularizer is multi-faceted, e.g., it serves for promoting desired structures in the decision vector, such as sparsity in controller designs using the ℓ_1 -norm [6], preventing overfitting in machine learning using the squared ℓ_2 -norm [7], and enforcing constraints using indicator functions of convex sets.

First-order methods [8]–[15] 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 ($g(\cdot) \equiv 0$), [9], [11], [13] exploit the history of gradient and iterate values to achieve linear convergence rate for strongly convex objectives. [14] 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 a large number of total iterations to reach a prescribed accuracy. This constitutes a key limitation for first-order methods, which is most pronounced in applications where high-accuracy solutions are pursued in a 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 [16]–[19] 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 the Hessian scaling in the evaluation of the proximal operator. Another challenge in designing second-order methods lies in constructing distributedly computable Newton updates. Computing curvature-guided updates requires solving a linear system that, in general, involves global information, whence a direct application of the Newton method is not feasible. Moreover, the standard Newton method requires backtracking line search to select appropriate step sizes for ensuring global convergence [20]. 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 [17] propose to use matrix splitting techniques in the dual problem, so that the Hessian inverse admits a distributedly computable Taylor expansion. By truncating the Taylor series to K terms, agents may compute local updates with an additional K rounds of communication

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).

Freris (correspondence) was supported by the Ministry of Science and Technology of China under grant 2019YFB2102200. Voulgaris and Li were supported by NSF grants CCF-1717154, CPS-1932529 and CMMI-2137764.

loops with their neighbors. With $g(\cdot) = 0$, [18] and [19] 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 [18] and [19] are effectively solving a different problem (penalized version) compared to (1) when using constant stepsizes, 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) [21], [22]. In the setting of distributed primal-dual algorithms [23]–[28], agents solve a sub-optimization problem at each iteration, which often involves multiple inner loops and thus induces heavy computation burden. Several approximation schemes [29]–[34] 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 [35]. Further acceleration can be achieved by resorting to Newton or quasi-Newton primal updates [32]–[34]. However, all of them are synchronous algorithms considering the smooth problem ($g(\cdot) \equiv 0$) and [33], [34] 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 [36] and Federated Learning [37], [38], 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 a 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 us 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 DistRistributed 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 $\mathcal{O}(\frac{1}{T})$ convergence rate to optimality under convex objectives, but also theoretically reveals the discrepancies among them. When strong convexity is further assumed, we establish linear convergence rates 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 rates in expectation, under strong convexity. This setting removes the need for a central clock in the network, and further allows for an arbitrary number of agents to be active at each iteration. We demonstrate the merits of the proposed framework through simulations 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_i^t 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 $\{1, \dots, m\}$ is abbreviated as $[m]$ and the proximal mapping associated with a function $g(\cdot) : \mathbb{R}^d \rightarrow \mathbb{R}$ is defined as $\text{prox}_g^\mu(v) := \underset{\theta \in \mathbb{R}^d}{\text{argmin}} \left\{ g(\theta) + \frac{\mu}{2} \|\theta - v\|^2 \right\}$. 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.

A. Problem formulation

We capture the network topology by an undirected graph $G = \{\mathcal{V}, \mathcal{E}\}$ where $\mathcal{V} := [m]$ denotes the vertex set and the edge set $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ contains the pair (i, j) if and only if agent i can communicate with agent j . We do not consider self loops, i.e., $(i, i) \notin \mathcal{E}$ for any $i \in [m]$. For notational convenience, we enumerate the edge set (arbitrary order) and use \mathcal{E}_k to denote the k -th edge, $k \in [n]$, where $n := |\mathcal{E}|$ is the number of edges. Moreover, the set of neighbors of agent i is defined as $\mathcal{N}_i := \{j \in \mathcal{V} : (i, j) \in \mathcal{E}\}$. 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 $(i, j) \in \mathcal{E}$. The consensus formulation is given by:

$$\begin{aligned} & \underset{x_i, \theta, z_{ij} \in \mathbb{R}^d}{\text{minimize}} \quad \left\{ \sum_{i=1}^m f_i(x_i) + g(\theta) \right\}, \\ & \text{s.t. } x_i = z_{ij} = x_j, \forall i \in [m] \text{ and } j \in \mathcal{N}_i. \\ & \quad x_l = \theta, \text{ for one arbitrary } l \in [m]. \end{aligned} \quad (2)$$

Note that we have also introduced θ to separate the argument of the smooth and nonsmooth functions and only enforce the equality constraint for θ 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 \mathcal{E}_k in the graph, $k \in [n]$: $[\hat{A}_s]_{ki} = [\hat{A}_d]_{kj} = 1$ if and only if $\mathcal{E}_k = (i, j)$, and 0 otherwise. Problem (2) can then be compactly expressed using the concatenated column vectors $x := [x_1^\top, \dots, x_m^\top]^\top, z := [z_1^\top, \dots, z_n^\top]^\top$ (we note a slight abuse of notation in using $z_k \equiv z_{ij}$ where $k \in [n]$ is the corresponding edge $(i, j) \in \mathcal{E}$ in the enumeration order) as:

$$\begin{aligned} & \underset{x \in \mathbb{R}^{md}, \theta \in \mathbb{R}^d, z \in \mathbb{R}^{nd}}{\text{minimize}} && \{F(x) + g(\theta)\}, \\ \text{s.t. } & Ax = \begin{bmatrix} \hat{A}_s \otimes I_d \\ \hat{A}_d \otimes I_d \end{bmatrix} x = \begin{bmatrix} I_{nd} \\ I_{nd} \end{bmatrix} z = Bz, \\ & && S^\top x = \theta, \end{aligned} \quad (3)$$

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^\top matrix serves to select the l -th component of x held by the agent l , i.e., $S^\top x = x_l$. 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.

$$\hat{E}_s = \hat{A}_s - \hat{A}_d, \quad \hat{E}_u = \hat{A}_s + \hat{A}_d, \quad (4a)$$

$$\hat{L}_s = \hat{E}_s^\top \hat{E}_s, \quad \hat{L}_u = \hat{E}_u^\top \hat{E}_u, \quad (4b)$$

$$\hat{D} = \frac{1}{2}(\hat{L}_s + \hat{L}_u) = \hat{A}_s^\top \hat{A}_s + \hat{A}_d^\top \hat{A}_d, \quad (4c)$$

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 , and D .

B. Background on ADMM

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

$$\begin{aligned} \mathcal{L}(x, \theta, z; y, \lambda) &:= F(x) + g(\theta) + y^\top (Ax - Bz) \\ &+ \lambda^\top (S^\top x - \theta) + \frac{\mu_z}{2} \|Ax - Bz\|^2 + \frac{\mu_\theta}{2} \|S^\top x - \theta\|^2, \end{aligned} \quad (5)$$

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 μ_z and μ_θ 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, θ, z) , and then performs gradient ascent on the dual variables (y, λ) :

$$x^{t+1} = \underset{x}{\text{argmin}} \mathcal{L}(x, \theta^t, z^t; y^t, \lambda^t), \quad (6a)$$

$$\theta^{t+1} = \underset{\theta}{\text{argmin}} \mathcal{L}(x^{t+1}, \theta, z^t; y^t, \lambda^t), \quad (6b)$$

$$z^{t+1} = \underset{z}{\text{argmin}} \mathcal{L}(x^{t+1}, \theta^{t+1}, z; y^t, \lambda^t), \quad (6c)$$

$$y^{t+1} = y^t + \mu_z (Ax^{t+1} - Bz^{t+1}), \quad (6d)$$

$$\lambda^{t+1} = \lambda^t + \mu_\theta (S^\top x^{t+1} - \theta^{t+1}). \quad (6e)$$

The above iterations fall into the category of 3-block ADMM which is not guaranteed to converge for arbitrary $\mu_z, \mu_\theta > 0$ [39]. 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 ℓ_1 -norm, squared ℓ_2 -norm, and indicators of several convex sets, a closed-form solution exists. For other cases, one would often resort to the fact that the proximal operator is separable, Lipschitz continuous with constant 1, firmly nonexpansive, and the associated Moreau envelope function is continuously differentiable irrespective of the function $g(\cdot)$, to devise efficient algorithms to approximate the proximal mapping. We refer readers to [40] for more details. Step (6c) results from the introduction of $\{z_{ij}\}$ -variables, and it does not require explicit evaluation as we demonstrate in the Section III.

C. Introduction to quasi-Newton methods

Quasi-Newton methods [20] 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 $\mathcal{O}(d^2)$ for general problems, as compared to solving a linear system with computational costs $\mathcal{O}(d^3)$ 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) [41]–[44], which is considered to be the most effective in terms of acceleration and self-correcting capabilities [20].

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:

$$\begin{aligned} & \underset{H^{-1}}{\text{minimize}} \quad \|H^{-1} - (H^t)^{-1}\|_W \\ & \text{s.t. } H^{-1} = (H^{-1})^\top, \quad H^{-1}q^t = s^t, \end{aligned} \quad (9)$$

where $\|M\|_W := \left\| W^{\frac{1}{2}} M W^{\frac{1}{2}} \right\|_F$ denotes the weighted Frobenius norm with W being the average Hessian [20]. Problem (9) admits a closed-form solution which gives rise to the update formula for $(H^{t+1})^{-1}$ in BFGS:

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

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 rates. However, the 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 $\widehat{\mathcal{L}}(\cdot)$ can be obtained using second-order expansion as follows:

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

where we abbreviated the $\mathcal{L}(x^t, \theta^t, z^t; y^t, \lambda^t)$ as $\mathcal{L}^t(x^t)$, 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 $\widehat{\mathcal{L}}(\cdot)$ over x and replace step (6a) with the following one-step update:

$$x^{t+1} = x^t - (H^t)^{-1} \nabla_x \mathcal{L}(x^t, \theta^t, z^t; y^t, \lambda^t). \quad (11)$$

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

$$\theta^{t+1} = \mathbf{prox}_{g/\mu_\theta}(S^\top x^{t+1} + \frac{1}{\mu_\theta} \lambda^t). \quad (12)$$

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:

$$B^\top y^t + \mu_z B^\top (A x^{t+1} - B z^{t+1}) = 0. \quad (13)$$

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:

$$\begin{aligned} x^{t+1} &= x^t - (H^t)^{-1} [\nabla F(x^t) + \phi^t + S \lambda^t + \frac{\mu_z}{2} L_s x^t \\ &\quad + \mu_\theta S (S^\top x^t - \theta^t)], \end{aligned} \quad (14a)$$

$$\theta^{t+1} = \mathbf{prox}_{g/\mu_\theta}(S^\top x^{t+1} + \frac{1}{\mu_\theta} \lambda^t), \quad (14b)$$

$$\phi^{t+1} = \phi^t + \frac{\mu_z}{2} L_s x^{t+1}, \quad (14c)$$

$$\lambda^{t+1} = \lambda^t + \mu_\theta (S^\top x^{t+1} - \theta^{t+1}). \quad (14d)$$

Proof: See Appendix A.

Remark 1. We emphasize that $(H^t)^{-1}$ is for notational purposes and no matrix inversion is needed in all cases (the exact computation scheme will be specified in the subsequent subsections). 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 μ_z, μ_θ 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. We note that a 2-block ADMM can be achieved directly without introducing z variables. However, such a direct formulation induces additional communication rounds when curvature information is computed. We further discuss this in *Remark 2*.

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 approximated Hessian of the augmented Lagrangian as:

$$H^t = J^t + \mu_z D + \mu_\theta S S^\top + \epsilon I_{md}, \quad (15)$$

where we have introduced $\epsilon > 0$ to provide additional robustness for our approximation. Notice that $\mu_z D + \mu_\theta S S^\top + \epsilon I_{md}$ 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, $(H^t)^{-1} \nabla_x \mathcal{L}^t$ in (11) and equivalently in (14a), can be computed individually by agents. More precisely, agent i computes the update u_i^t by solving the following linear system:

$$H_{ii}^t u_i^t = \nabla_x \mathcal{L}_i^t. \quad (16)$$

Therefore, once the right-hand side of (16) is obtained by i -th agent, no additional communication is needed to solve for

u_i^t . 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 Hessian of the augmented Lagrangian 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_i^t either requires the presence of a fusion center that gathers all $\nabla_x \mathcal{L}_i^t$ 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 [18], [19], [33], [34], [45]–[48]. 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 [49]. In contrast, all our proposed methods feature minimal communication complexity (see step 8 of Alg. 1), and are amenable to an asynchronous implementation. Different choices of J^t in (15) affect the local computational cost and convergence rate as we elaborate next.

A. Gradient updates

By choosing $J_{\text{Gradient}}^t \equiv 0$, 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 ϵ . We note that the proposed algorithm recovers Decentralized Linearized ADMM (DLM) [30] with $g(\cdot) = 0$ as a special case of (3). Specifically, agent i computes u_i^t from (16) as:

$$u_i^t = (\mu_z |\mathcal{N}_i| + \delta_{il} \mu_\theta + \epsilon)^{-1} \nabla_x \mathcal{L}_i^t, \quad (17)$$

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 ϵ . Computing updates for agents using (17) involves $\mathcal{O}(d)$ computational costs, and we proceed to specify how curvature information is incorporated with nonzero J^t in the following.

B. Newton updates

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

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

We note that since $F(x^t) = \sum_{i=1}^m f_i(x_i^t)$, $\nabla^2 F(x^t)$ is a block diagonal matrix with the i -th block being $\nabla^2 f_i(x_i^t)$. 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 $\mathcal{O}(d^3)$ 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 to be time-invariant in this paper). In [34], authors propose to estimate $\nabla^2 F(x^t)$ using the BFGS formula with each node's 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 [34]. We note that such schemes not only incur higher communication costs per round, but also induce $\mathcal{O}(d^3)$ 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:

$$q_i^t := \nabla f_i(x_i^{t+1}) - \nabla f_i(x_i^t) + (\mu_z |\mathcal{N}_i| + \delta_{il} \mu_\theta + \epsilon) s_i^t, \text{ and } s_i^t := x_i^{t+1} - x_i^t. \quad (19)$$

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 $(\nabla_x^2 \mathcal{L}_i^t)^{-1}$. 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 (16). Instead, computing u_i^t is tantamount to performing matrix multiplication $(H_{ii}^t)^{-1} \nabla_x \mathcal{L}_i^t$. 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 $\mathcal{O}(d^3)$ to $\mathcal{O}(d^2)$. For the sake of comparison with the gradient and Newton updates, we define:

$$J_{\text{BFGS}}^t := H_{\text{BFGS}}^t - \mu_z D - \mu_\theta S S^\top - \epsilon I_{md}, \quad (20)$$

where H_{BFGS}^t is obtained by the BFGS formula with $\{q_i^t, s_i^t\}_{i=1}^m$ pairs defined in (19). We proceed to describe the distributed implementation of the proposed algorithms.

IV. ASYNCHRONOUS DESCRIPTION

In synchronous algorithms, 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 [21] 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 *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 algorithm by defining the operator $T : \mathbb{R}^{(2m+2)d} \rightarrow \mathbb{R}^{(2m+2)d}$ as follows:

$$v^{t+1} = Tv^t, \quad (21)$$

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:

$$\Omega^t := \begin{bmatrix} X^t & 0 & 0 & 0 \\ 0 & X^t & 0 & 0 \\ 0 & 0 & X_{ll}^t & 0 \\ 0 & 0 & 0 & X_{ll}^t \end{bmatrix}, \quad (22)$$

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 corresponding to the i -th agent and taking values as the identity matrix I_d or a zero matrix. Using the definition (21) and (22), the proposed asynchronous algorithms are expressed as:

$$v^{t+1} = v^t + \Omega^{t+1}(Tv^t - v^t). \quad (23)$$

The above construction corresponds to activating agents, i.e., the i -th agent only updates the corresponding pair (x_i^t, ϕ_i^t) (additionally (θ^t, λ^t) if $i = l$) if and only if $X_{ii}^{t+1} = I_d$. We proceed to describe the implementation details of DRUID.

A. Distributed and Asynchronous Implementation

The proposed algorithms admit the exact same 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_i^t, ϕ_i^t) while the l -th agent additionally holds the pair (θ^t, λ^t) pertaining to the nonsmooth regularization function $g(\cdot)$. The gradient of the augmented Lagrangian pertaining to agent i , $\nabla_x \mathcal{L}_i^t$, is expressed as:

$$h_i^t = \nabla f_i(x_i^t) + \phi_i^t + \frac{\mu_z}{2} \sum_{j \in \mathcal{N}_i} (x_i^t - x_j^t) + \delta_{il} \mu_\theta (x_i^t - \theta^t + \frac{1}{\mu_\theta} \lambda^t). \quad (24)$$

Before we present the asynchronous implementation (Alg. 1), we describe the synchronous case as a special case to shed some light on the design principles. At the beginning of each round, all agents become active and estimate their local curvatures as in steps 3-4 (without communication, irrespective

Algorithm 1 DRUID

Initialization: zero initialization for all variables.

- 1: **for** $t = 0, 1, \dots$ **do**
 - 2: **for** all active agents i **do**
 - 3: Compute the local curvature H_{ii}^t :
 - 4: $J_{ii}^t = \begin{cases} 0 & \text{Gradient updates} \\ \nabla^2 f_i(x_i^t) & \text{Newton updates} \end{cases}$ } only for gradient or Newton updates
 - 5: $(H^t)_{ii} \leftarrow J_{ii}^t + (\mu_z |\mathcal{N}_i| + \delta_{il} \mu_\theta + \epsilon) I_d$
 - 6: Primal update: Compute h_i^t as in (24)
 - 7: $\begin{cases} H_{ii}^t u_i^t = h_i^t & \text{Gradient/Newton updates} \\ u_i^t = (H_{ii}^t)^{-1} h_i^t & \text{BFGS updates} \end{cases}$
 - 8: $x_i^{t+1} = x_i^t - u_i^t$
 - 9: Communication: Broadcast x_i^{t+1} to neighbors
 - 10: Dual update: $\phi_i^{t+1} = \phi_i^t + \frac{\mu_z}{2} \sum_{j \in \mathcal{N}_i} (x_i^{t+1} - x_j^{t+1})$
 - 11: Updates pertaining to the regularization function: **if** $i = l$ **then**
 - 12: $\theta^{t+1} = \text{prox}_{g/\mu_\theta}(x_l^{t+1} + \frac{1}{\mu_\theta} \lambda^t)$
 - 13: $\lambda^{t+1} = \lambda^t + \mu_\theta (x_l^{t+1} - \theta^{t+1})$
 - 14: **end if**
 - 15: Curvature estimation update (BFGS only): Update $(H_{ii}^{t+1})^{-1}$ using $\{q_i^t, s_i^t\}$ in (19) and the formula in (10)
 - 16: **end for**
-

of the computing schemes). For the BFGS computing scheme, no computation is required at these steps. Agents then carry primal updates by first computing h_i^t expressed in (24). We emphasize that h_i^t can be computed without communication, since each agent i already has access to the variables of its neighbors, $\{x_j^t | j \in \mathcal{N}_i\}$, from the previous round with zero initialization. If gradient or Newton updates is opted as the computing scheme, agents compute u_i^t by solving the linear system (for gradient descent u_i^t can be trivially solved since H_{ii}^t is a constant scalar times the identity matrix). For the BFGS scheme, u_i^t is computed by performing matrix-vector multiplication $(H_{ii}^t)^{-1} h_i^t$. Once u_i^t is obtained, agents update their x_i^{t+1} in step 7. We note that the *only communication round* occurs at step 8 where agents broadcast x_i^{t+1} to their neighbors (thus incurring the same cost for all computing schemes, i.e., $|\mathcal{N}_i|d$ for agent i). Dual updates are executed in step 9. We require agents to store $\{x_j^{t+1}, j \in \mathcal{N}_i\}$, to execute step 5 in the next iteration. In addition to the primal-dual variables (x_l, ϕ_l) , the l -th agent further holds (θ, λ) associated with the regularization term $g(\cdot)$, which are updated in steps 11-12. Finally, if BFGS is opted as the updating scheme, agents update local curvature estimation $(H_{ii}^{t+1})^{-1}$ in step 14.

In the case of asynchronous implementation, we equip each agent with a buffer so that even if agents are not active, they can still receive information from their neighbors. Once active, the i -th agent executes steps 3-4 using only local information and then retrieves the most recent x_j^t from its buffer for computing h_i^t in step 5. Once u_i^t is computed and

x_i^{t+1} is updated in steps 6 and 7, respectively, the active agent i broadcasts x_i^{t+1} to its neighbors, whose buffers store the updated x_i^{t+1} . Finally, the active agents check their buffers for most recent x_j^{t+1} and proceed to dual updates and finish their 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 (21), 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 but analyze convergence using v_α for technical convenience. We note that their equivalence is established by Lemma 1 using $\phi = E_s^\top \alpha, z = \frac{1}{2} E_u x$. We first establish the sublinear convergence rate of the synchronous DRUID under the assumption that the local objective functions are convex. By further assuming strong convexity, we establish the global linear convergence rate for both the synchronous and the asynchronous settings.

A. Preliminaries

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

$$m_f I_d \preceq \nabla^2 f_i(x_i) \preceq M_f I_d, \quad (25)$$

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

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

$$(x - y)^\top (\partial g(x) - \partial g(y)) \geq 0, \quad (26)$$

where the inequality is meant for arbitrary elements in the subdifferential sets $\partial g(x)$ and $\partial g(y)$, respectively.

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 [50].

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:

$$H_{\text{BFGS}}^t \preceq \psi I_{md}. \quad (27)$$

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., $(H_{\text{BFGS}}^t)^{-1} = (\widehat{H}_{\text{BFGS}}^t)^{-1} + \frac{1}{\psi} I_{md}$, where $(\widehat{H}_{\text{BFGS}}^t)^{-1}$ is obtained through (10). Other means include using regularized BFGS updates and invoking L-BFGS [51] estimation by using a finite prescribed number of $\{q_i^t, s_i^t\}$ copies. In brief, we make this assumption for convenience and without serious loss in generality; see also [34] and [52].

When local functions are assumed to be only convex ($m_f = 0$), there might be multiple optimal primal solutions, each with multiple optimal dual solution. However, there exists a unique dual pair that lies in the column space of some matrix, to be defined and formalized in the following.

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

$$\begin{aligned} \nabla F(x^*) + E_s^\top \alpha^* + S \lambda^* &= 0, & \text{KKTa} \\ \partial g(\theta^*) - \lambda^* &\ni 0, & \text{KKTb} \\ E_s x^* &= 0, & \text{KKTc} \\ E_u x^* &= 2z^*, & \text{KKTd} \\ S^\top x^* &= \theta^*. & \text{KKT e} \end{aligned}$$

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

Proof: See Appendix A of the extended version of this paper.

We proceed to establish a lemma that characterizes the suboptimality of the iterates when replacing (6a) with (14a).

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

$$\begin{aligned} e^t + \nabla F(x^{t+1}) - \nabla F(x^*) + \epsilon(x^{t+1} - x^t) + E_s^\top (\alpha^{t+1} - \alpha^*) \\ + \mu_z E_u^\top (z^{t+1} - z^t) + S (\lambda^{t+1} - \lambda^* + \mu_\theta (\theta^{t+1} - \theta^t)) = 0 \end{aligned}$$

where the error term is:

$$e^t = \nabla F(x^t) - \nabla F(x^{t+1}) + J^t (x^{t+1} - x^t). \quad (28)$$

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 $\overline{J}^t = J^t + \epsilon I$, and the scaling matrix \mathcal{G}^t as follows:

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

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 μ_z and ϵ such that: $\epsilon > \frac{M_f}{2}, \mu_z \epsilon < \psi^2$. Then the following holds:

$$\begin{aligned} & \frac{1}{T} \frac{\mu_z}{2} \|x^1\|_{L_s}^2 + \frac{\mu_\theta}{T} \|x_1^1 - \theta^1\|^2 + \frac{1}{T} \sum_{t=1}^T \|v_\alpha^{t+1} - v_\alpha^t\|_{\mathcal{G}^t}^2 \quad (30) \\ & \geq \frac{1}{T} \frac{\mu_z}{2} \|x^{T+1}\|_{L_s}^2 + \frac{\mu_\theta}{T} \|x_1^{T+1} - \theta^T\|^2 \\ & + \frac{1}{T} \sum_{t=1}^T \left\{ \frac{\epsilon}{\rho \bar{M}^2} \|\nabla F(x^t) + E_s^\top \alpha^t + S \lambda^t\|^2 + \mu_\theta \|\theta^{t+1} - \theta^t\|^2 \right. \\ & + 2\mu_z \|z^{t+1} - z^t\|^2 + \left(\frac{\mu_z}{2} - \frac{\epsilon \mu_z^2}{2\bar{M}^2} \right) \|x^t\|_{L_s}^2 \\ & \left. + \left(\mu_\theta - \frac{2\epsilon \mu_\theta^2}{\bar{M}^2(\rho-1)} \right) \|S^\top x^t - \theta^t\|^2 \right\}, \end{aligned}$$

where $d_{\max} = \max_i |\mathcal{N}_i|$, $\rho > \max \left\{ \frac{2\epsilon \mu_\theta}{\bar{M}^2}, \sigma_{\max}^{L_s} \right\} + 1$, and \bar{M} (for each scheme) is given by: $\bar{M}_{\text{Gradient}} = \mu_z d_{\max} + \epsilon + \mu_\theta$, $\bar{M}_{\text{Newton}} = M_f + \mu_z d_{\max} + \epsilon + \mu_\theta$, $\bar{M}_{\text{BFGS}} = \psi$.

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:

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

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

$$\frac{1}{T} \sum_{t=1}^T \|S^\top x^t - \theta^t\|^2 = \mathcal{O}\left(\frac{1}{T}\right). \quad (31c)$$

Proof: See Appendix B.

C. Linear Convergence

By further assuming strongly convex $f_i(\cdot)$ ($m_f > 0$), we establish the 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

$$\tau_{\text{Gradient}}^t = M_f, \quad (32a)$$

$$\tau_{\text{Newton}}^t = \min \left\{ 2M_f, \frac{L_f}{2} \|x^{t+1} - x^t\| \right\}, \quad (32b)$$

$$\tau_{\text{BFGS}}^t = \|H^t - H^{t+1}\| \leq 2\psi. \quad (32c)$$

Proof: See Appendix C.

The above lemma complements the result presented in [30] and [32]. 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

Table I: Comparison between updating schemes in terms of communication and computation costs per iteration, and storage costs per agent as a function of vector dimension d and neighborhood size $|\mathcal{N}_i|$. The last column characterizes the decay rate of e^t in terms of the difference $x_e^t := x^{t+1} - x^t$.

Methods	Comm. costs	Comp. costs	Storage costs	Decay rate
Gradient	$ \mathcal{N}_i d$	$\mathcal{O}(d)$	$\mathcal{O}(d)$	$\mathcal{O}\left(\ x_e^t\ \right)$
BFGS	$ \mathcal{N}_i d$	$\mathcal{O}(d^2)$	$\mathcal{O}(d^2)$	$o\left(\ x_e^t\ \right)$
Newton	$ \mathcal{N}_i d$	$\mathcal{O}(d^3)$	$\mathcal{O}(d^2)$	$\mathcal{O}\left(\ x_e^t\ ^2\right)$

(32b). 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 $\|H^t - H^{t+1}\| \rightarrow 0$, we conclude that the error term in BFGS diminishes superlinearly with respect to $\|x^{t+1} - x^t\|$. We have summarized these along with other properties of different updating schemes in Table I. Note that the l -th agent, that performs updates pertaining to $g(\cdot)$, additionally holds the (θ, λ) pair; thus, storage increases by $2d$ and additional computation is incurred for evaluating the proximal operator (typically $\mathcal{O}(d)$). The fact that all computing schemes share the same communication cost (equal to the vector dimension d) is because agents only communicate once per iteration with their neighbors (step 8 of Alg. 1).

Before establishing the linear convergence rate of DRUID, we recall v_α and introduce the following diagonal scaling matrix $\mathcal{H} = \text{diag}[\epsilon, 2\mu_z, \frac{2}{\mu_z}, \mu_\theta, \frac{1}{\mu_\theta}]$ similar to (29).

Theorem 2. Under Assumptions 1–4 with $m_f > 0$, we denote the maximum and minimum eigenvalue of L_u as $\sigma_{\max}^{L_u}$ and $\sigma_{\min}^{L_u}$ respectively. Let σ_{\min}^+ be the smallest positive eigenvalue of CC^\top , where $C := \begin{bmatrix} E_s \\ S^\top \end{bmatrix}$, and $c_{\max} = 2 \cdot \max\{M_f, \psi\}$. By selecting $\mu_z = 2\mu_\theta$, $\epsilon > \frac{c_{\max}^2(m_f + M_f)}{2m_f M_f}$, and arbitrary constant $\zeta \in \left(\frac{m_f + M_f}{2m_f M_f}, \frac{\epsilon}{(\tau^t)^2} \right)$, the iterates generated by (14) satisfy:

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

where η satisfies:

$$\eta = \min \left\{ \left(\frac{2m_f M_f}{m_f + M_f} - \frac{1}{\zeta} \right) \frac{1}{\epsilon + \mu_\theta (\sigma_{\max}^{L_u} + 2)}, \frac{1}{2}, \frac{2}{5} \frac{\mu_\theta \sigma_{\min}^+}{m_f + M_f}, \frac{\mu_\theta \sigma_{\min}^+ (\epsilon - \zeta (\tau^t)^2)}{5((\tau^t)^2 + \epsilon^2)}, \frac{\sigma_{\min}^+}{5 \max\{1, \sigma_{\max}^{L_u}\}} \right\} \quad (33)$$

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, 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:

$$\eta_{\text{exact}} = \min \left\{ \frac{2m_f M_f}{m_f + M_f} \frac{1}{\mu_\theta (\sigma_{\max}^{L_u} + 2)}, \frac{1}{2}, \frac{2}{5} \frac{\mu_\theta \sigma_{\min}^+}{(m_f + M_f)}, \frac{\sigma_{\min}^+}{5 \max\{1, \sigma_{\max}^{L_u}\}} \right\}. \quad (34)$$

Denoting $\kappa = M_f/m_f$ and choosing $\mu_\theta = m_f \sqrt{\kappa}$, we obtain an iteration complexity of $\mathcal{O}(\sqrt{\kappa} \log(1/\epsilon))$ from (34),

where ε is the solution accuracy and not to be confused with the hyperparameter ϵ . Moreover, since σ_{\min}^+ is related to the smallest positive eigenvalue of L_s , i.e., the algebraic connectivity of the graph, (34) implies that a more connected graph (larger σ_{\min}^+) gives rises to a larger η_{exact} , and faster convergence rates. On the other hand, the rate η established in (33) is no larger than η_{exact} in (34): this is due to the fact that we have replaced the exact optimization step with the one step update (14a). Characterizing the gap between η and η_{exact} serves to reveal the differences between using gradients, Newton, and BFGS updates. This is achieved by comparing the upper bound for the error term τ^t , and how e^t (Lemma 4) evolves as characterized by the last column of Table I. As established in Section V-B, $\lim_{t \rightarrow \infty} \|x^{t+1} - x^t\| = 0$, the error bound $\tau_{\text{Newton/BFGS}}^t \rightarrow 0$ from inspecting (32b) and (32c). In other words, we can recover the convergence rate in (34) only if we use curvature-aided updates.

We recall that the asynchronous implementation in (23) 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_α to converge under random activation, and then show that the implementation (23) satisfies this condition. We first define the following activation matrix corresponding to $v_\alpha = [x; z; \alpha; \theta; \lambda] \in \mathbb{R}^{(m+2n+2)d}$:

$$\Omega_\alpha^t := \begin{bmatrix} X^t & 0 & 0 & 0 & 0 \\ 0 & Y^t & 0 & 0 & 0 \\ 0 & 0 & Y^t & 0 & 0 \\ 0 & 0 & 0 & X_{ll}^t & 0 \\ 0 & 0 & 0 & 0 & X_{ll}^t \end{bmatrix}. \quad (35)$$

The activation matrix Ω_α^t differs from Ω^t in (22) as we allow (z^t, α^t) to be updated independently from x^t , captured by the random matrix $Y^t \in \mathbb{R}^{nd \times nd}$. We can similarly develop an asynchronous algorithm as:

$$v_\alpha^{t+1} = v_\alpha^t + \Omega_\alpha^t (T_\alpha v_\alpha^t - v_\alpha^t), \quad (36)$$

where the operator $T_\alpha : \mathbb{R}^{(m+2n+2)d} \rightarrow \mathbb{R}^{(m+2n+2)d}$ is equivalent to the synchronous updates (14). The update (36) captures a wider range of random activation schemes than the update in (23), but it is more costly to implement. Therefore, we only use (36) as a guideline for analysis. We proceed to define $\mathbb{E}^t[\cdot] := \mathbb{E}[\cdot | \mathcal{F}^t]$, where \mathcal{F}^t is the filtration generated by (X^1, \dots, X^t) and (Y^1, \dots, Y^t) .

Theorem 3. Consider the iterates generated by the asynchronous algorithm (36). Under the same setting as in the Theorem 2 and any activation scheme such that $\mathbb{E}^t[\Omega_\alpha^{t+1}] = \Omega_\alpha \succ 0$, then the following holds:

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

where for $i \in [m], j \in [n]$, we denote $\mathbb{E}^t[X_{ii}^{t+1}] = p_i^X, \mathbb{E}^t[Y_{jj}^{t+1}] = p_j^Y, p^{\min} := \min_{i \in [m], j \in [n]} \{p_i^X, p_j^Y\}$, and η is given by (33).

Proof: See Appendix C.

We note that the activation of the asynchronous scheme using $(\Omega_\alpha^{t+1}, v_\alpha^{t+1})$ described by (36) 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 $\mathbb{E}^t[\Omega_\alpha^{t+1}] \succ 0$, iterates v_α^t converge linearly in expectation. On the other hand, the implementation using (Ω^{t+1}, v^{t+1}) described by (23), only needs to specify X^{t+1} , i.e., activating agents. The difference of the two lies in the fact that (36) first updates a subset of α_k , then computes $\phi = E_s^\top \alpha$, while (23) directly updates a subset of ϕ_i . In the following corollary, we show that using the activation scheme described by (23), the induced iterates v_α^t converge linearly in expectation.

Corollary 2. Consider activation matrices Ω^{t+1} in (22) and Ω_α^{t+1} in (35). Under the same X^{t+1} and updating scheme (23), if $\mathbb{E}^t[X^{t+1}] \succ 0$, then it holds that $\mathbb{E}^t[\Omega_\alpha^{t+1}] \succ 0$.

Proof: See Appendix C.

Since $\mathbb{E}^t[\Omega_\alpha^{t+1}] \succ 0$, we conclude that the implementation using (Ω^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 PGE [10], P2D2 [12], and ESOM [33]. Note that ESOM and other existing (quasi) Newton methods [19], [32], [34], [47] do not support nonsmooth regularization functions and therefore ESOM is only compared in the Fig 4, where the regularization function is differentiable. We consider the following distributed optimization problem:

$$\underset{x \in \mathbb{R}^d}{\text{minimize}} \ G(x) = \left\{ \sum_{i=1}^m f_i(x) + g(x) \right\}, \quad (37)$$

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 LIBSVM¹ and UCI Machine Learning Repository². 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 if necessary. The mixing matrices of P2D2 and ESOM are generated using the Metropolis rule [12] while the mixing matrix of PG-EXTRA is generated by the Laplacian-based constant weight matrix [10], respectively.

A. Distributed LASSO

The distributed LASSO problem considers solving (37) with $g(x) = \gamma \|x\|_1$, $\gamma \in \mathbb{R}$, and $f_i(\cdot) : \mathbb{R}^d \rightarrow \mathbb{R}$ defined as:

$$f_i(x) = \frac{1}{2} \sum_{i=1}^{m_i} (a_i^\top x - b_i)^2. \quad (38)$$

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, using 9,000 data points of dimension $d = 4$.

We note that all algorithms have the same communication

¹<https://www.csie.ntu.edu.tw/~cjlin/libsvm/>

²<https://archive.ics.uci.edu/ml/index.php>

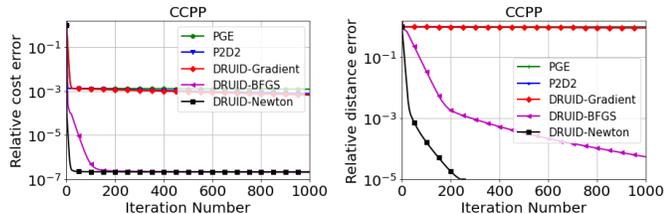


Figure 1: Performance comparison on the CCPP dataset. We plot the iteration number versus the relative cost error (left) $\frac{G(x^t) - G(x^*)}{G(x^0) - G(x^*)}$ and the relative distance error (right) $\frac{\|x^t - x^*\|}{\|x^0 - x^*\|}$ on a randomly generated graph consisting of $m = 20$ agents.

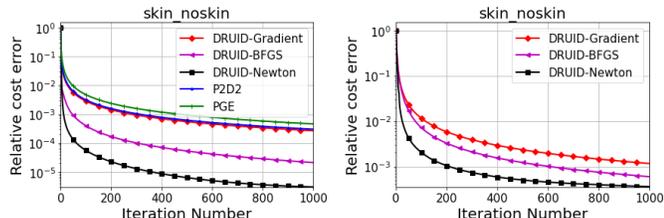


Figure 2: Performance comparison of DRUID algorithms and existing methods in a network with $m = 20$ agents, in synchronous (left) and asynchronous (right) settings. In each iteration of the asynchronous setting, half of the agents in the network are activated in a uniformly random fashion.

costs per iteration (this is due to the fact that only one round of communication of the local variable x_i is required for all updating schemes; see step 8 of Alg. 1), while first-order methods have lower computational costs. However, a significant reduction of iteration numbers for prescribed accuracy can be achieved by using (quasi) Newton methods.

B. Distributed Logistic Regression

The distributed logistic regression solves (37) with $g(x) = \gamma \|x\|_1$ and $f_i(\cdot) : \mathbb{R}^d \rightarrow \mathbb{R}$ defined as:

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

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. We consider 5,000 data points from the skin_noskin dataset with dimensions $d = 3$, and 2,000 data points from the ijcn1 dataset with dimensions $d = 22$. In Figure 2, we observe that the convergence is slower when only a subset of agents become active (due to less total computation/communication per round compared to the synchronous case). Note that P2D2 and PGE do not support asynchronous implementations. We further explore the effect of the graph topology by varying the size of the network m in Figure 3. We observe that DRUID is insensitive to networks with different sizes, but with fixed $p = 0.2$. This is consistent with our analysis where the convergence rate is affected by algebraic connectivity, but not system size m .

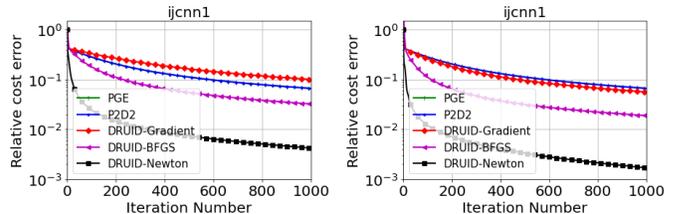


Figure 3: Performance comparison using the ijcn1 dataset with different network sizes. We plot the iteration number versus the relative cost error on random graphs with $m = 10$ (left) and $m = 20$ (right).

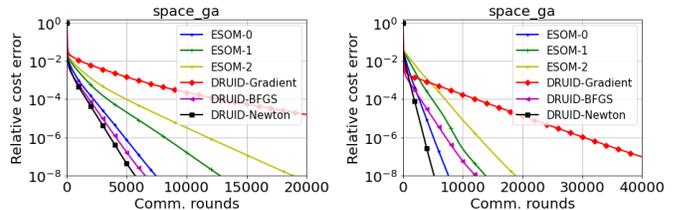


Figure 4: Performance comparison using the space_ga dataset. We use the communication rounds as the metric, with the number of agents and probability of generating an edge equal to $m = 20, p = 0.2$ (left) and $m = 40, p = 0.8$ (right).

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 the same as in (38) but with $g(x) = \gamma \|x\|^2$. We compare DRUID with ESOM- K , where K denotes the number of inner communication loops. 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 the use of consensus variables $\{z_{ij}\}$, DRUID-Newton utilizes the exact Hessian without inducing inner loops, and thus achieves the highest communication efficiency.

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. An asynchronous extension of the proposed algorithms is also presented. We establish a unified analytical framework for the proposed algorithms that reveals the difference between various updating schemes. Some future directions include extensions to time-varying and directed network topologies, stochastic gradient evaluation, and hybrid updating schemes.

APPENDIX A

Proof of Lemma 1: The proof was similarly derived in [23] and [26] for the case $g(\cdot) = 0$ and the suboptimization problem (6a) was solved exactly. We generalize the results by first writing $y^t = [\alpha^t; \beta^t]$ and recalling 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:

$$\alpha^{t+1} = \alpha^t + \mu_z(A_s x^{t+1} - z^{t+1}), \quad (39)$$

$$-\alpha^{t+1} = -\alpha^t + \mu_z(A_d x^{t+1} - z^{t+1}). \quad (40)$$

Recall that $E_s = A_s - A_d$ and $E_u = A_s + A_d$. By taking the sum and difference of (39) and (40), we obtain for $t \geq 0$,

$$z^{t+1} = \frac{\mu_z}{2} E_u x^{t+1}, \quad (41)$$

$$\alpha^{t+1} = \alpha^t + \frac{\mu_z}{2} E_s x^{t+1}. \quad (42)$$

This establishes that the dual update (6d) for y^{t+1} can be replaced by (42). Using the definition of $\phi^t = E_s^\top \alpha^t$ and premultiplying (42) 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:

$$\begin{aligned} x^{t+1} &= x^t - (H^t)^{-1} [\nabla F(x^t) + A^\top y^t + S\lambda^t \\ &\quad + \mu_z A^\top (Ax^t - Bz^t) + \mu_\theta S(S^\top x^t - \theta^t)]. \end{aligned} \quad (12)$$

Since $y^t = [\alpha^t; -\alpha^t]$ and $z^t = \frac{1}{2} E_u x^t$, we obtain:

$$A^\top y^t = [A_s^\top \quad A_d^\top] y^t = E_s^\top \alpha^t = \phi^t, \quad (43)$$

$$\mu_z A^\top (Ax^t - Bz^t) = \frac{\mu_z}{2} (2D - L_u) x^t = \frac{\mu_z}{2} L_s x^t, \quad (44)$$

where we have used the identity $A_s^\top - A_d^\top = E_s^\top$, $D = A^\top A$, and $\mu_z A^\top Bz^t = \frac{\mu_z}{2} E_u^\top E_u x^t = \frac{\mu_z}{2} L_u x^t$. After substituting (43) and (44) into (11), we obtain the desired. ■

Proof of Lemma 2: The KKT conditions for (3) are:

$$\nabla F(x^*) + A^\top y^* + S\lambda^* = 0, \quad (45a)$$

$$B^\top y^* = 0, \quad (45b)$$

$$\partial g(\theta^*) - \lambda^* \ni 0, \quad (45c)$$

$$Ax^* = Bz^*, \quad (45d)$$

$$S^\top x^* = \theta^*. \quad (45e)$$

Since the objective function is convex with linear constraints, strong duality holds. Recall the definition $B = [I_{md}; I_{md}] \in \mathbb{R}^{2md \times md}$. The condition (45b) 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 (45a) can be rewritten as:

$$\nabla F(x^*) + E_s^\top \alpha^* + S\lambda^* = 0. \quad (46)$$

Note that since E_s^\top has a nontrivial kernel for any network with agent number $m > 1$, there exist multiple α^* that satisfy (46). 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 $\xi^0 := [\alpha^0; \lambda^0]$ be any dual optimal that satisfies (46) and (45c). 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 $\text{col}(E_s^\top) \cap \text{col}(S) = 0$ and $\ker(S) = 0$, it holds that $\lambda^0 = \lambda^*$. We prove the uniqueness of ξ^* 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 one dimensional subspace spanned by consensus vector $\mathbf{1}$ and the kernel of SS^\top is spanned by vectors with the l -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:

$$\begin{aligned} \nabla F(x^t) + E_s^\top \alpha^t + S\lambda^t + \frac{\mu_z}{2} L_s x^t + \mu_\theta S(S^\top x^t - \theta^t) \\ + H^t(x^{t+1} - x^t) = 0 \end{aligned} \quad (47)$$

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

$$E_s^\top \alpha^t + \frac{\mu_z}{2} L_s x^t = E_s^\top \alpha^{t+1} - \frac{\mu_z}{2} L_s(x^{t+1} - x^t). \quad (48)$$

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

$$\begin{aligned} S\lambda^t + \mu_\theta S(S^\top x^t - \theta^t) \\ = S\lambda^{t+1} - \mu_\theta S(S^\top(x^{t+1} - x^t) - (\theta^{t+1} - \theta^t)). \end{aligned} \quad (49)$$

After substituting (48) and (49) into (47), we obtain:

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

Recall $2D = L_s + L_u$ from (4c). After adding and subtracting $(\mu_z D + \mu_\theta SS^\top + \epsilon I)(x^{t+1} - x^t)$ from (50), we obtain:

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

Moreover, $\frac{\mu_z}{2} L_u(x^{t+1} - x^t) = \frac{\mu_z}{2} E_u^\top E_u(x^{t+1} - x^t) = \mu_z E_u^\top(z^{t+1} - z^t)$ by (41). Recall 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 $\frac{\mu_z}{2} L_u(x^{t+1} - x^t)$ into the above, we obtain the desired. ■

APPENDIX B

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

$$(\lambda^{t+1} - \lambda^t)^\top (\theta^{t+1} - \theta^t) \geq 0, \quad (51)$$

$$(\lambda^{t+1} - \lambda^*)^\top (\theta^{t+1} - \theta^*) \geq 0. \quad (52)$$

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

$$\theta^{t+1} = \operatorname{argmin}_\theta \left\{ g(\theta) + \frac{\mu_\theta}{2} \left\| S^\top x^{t+1} + \frac{1}{\mu_\theta} \lambda^t - \theta \right\|^2 \right\}. \quad (53)$$

By the optimality condition of (53), we obtain:

$$0 \in \partial g(\theta^{t+1}) - \mu_\theta \left(\frac{1}{\mu_\theta} \lambda^t + S^\top x^{t+1} - \theta^{t+1} \right) = \partial g(\theta^{t+1}) - \lambda^{t+1},$$

where the last equality follows from the dual update (14d). Therefore, it holds that:

$$(\lambda^{t+1} - \lambda^t)^\top (\theta^{t+1} - \theta^t) \in (\partial g(\theta^{t+1}) - \partial g(\theta^t))^\top (\theta^{t+1} - \theta^t) \geq 0,$$

where the inequality follows from the convexity of $g(\cdot)$. Moreover,

$$(\lambda^{t+1} - \lambda^*)^\top (\theta^{t+1} - \theta^*) \in (\partial g(\theta^{t+1}) - \partial g(\theta^*))^\top (\theta^{t+1} - \theta^*) \geq 0,$$

where the inclusion follows from KKTb. The rest of the proof is constituted by the following:

(i) Establishing the convergence of $\|v_\alpha^t - v_\alpha^*\|_{\mathcal{G}^t}^2$ 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:

$$\begin{aligned} \frac{1}{M_f} \|\nabla F(x^t) - \nabla F(x^*)\|^2 &\leq (x^t - x^*)^\top (\nabla F(x^t) - \nabla F(x^*)) \\ &= (x^{t+1} - x^*)^\top (\nabla F(x^t) - \nabla F(x^*)) \\ &\quad + (x^t - x^{t+1})^\top (\nabla F(x^t) - \nabla F(x^*)). \end{aligned} \quad (54)$$

We proceed to establish an upper bound for the right-hand side of (54) by separately bounding the two components. Recall H^t in (15). From Lemma 3, the following holds:

$$\begin{aligned} \nabla F(x^t) - \nabla F(x^*) &= -\{E_s^\top (\alpha^{t+1} - \alpha^*) + S(\lambda^{t+1} - \lambda^* \\ &\quad + \mu_\theta(\theta^{t+1} - \theta^t)) + (J^t + \epsilon I)(x^{t+1} - x^t) \\ &\quad + \mu_z E_u^\top (z^{t+1} - z^t)\}. \end{aligned} \quad (55)$$

Denoting $\bar{J}^t = J^t + \epsilon I$ and using (55), we rewrite the first component of the right hand side of (54) as:

$$\begin{aligned} (x^{t+1} - x^*)^\top (\nabla F(x^t) - \nabla F(x^*)) &= \\ &= (x^{t+1} - x^*)^\top E_s^\top (\alpha^{t+1} - \alpha^*) \\ &= (x^{t+1} - x^*)^\top S (\lambda^{t+1} - \lambda^* + \mu_\theta(\theta^{t+1} - \theta^t)) \\ &= (x^{t+1} - x^*)^\top \bar{J}^t (x^{t+1} - x^t) \\ &= \mu_z (x^{t+1} - x^*)^\top E_u^\top (z^{t+1} - z^t). \end{aligned} \quad (56)$$

From the dual update, KKT conditions, and Lemma 1, the following holds:

$$\begin{aligned} (x^{t+1} - x^*)^\top E_s^\top &= \frac{2}{\mu_z} (\alpha^{t+1} - \alpha^t)^\top, \\ (x^{t+1} - x^*)^\top S &= (\theta^{t+1} - \theta^*)^\top + \frac{1}{\mu_\theta} (\lambda^{t+1} - \lambda^t)^\top, \\ (x^{t+1} - x^*)^\top E_u^\top &= (z^{t+1} - z^*)^\top. \end{aligned}$$

Using these expressions for $(x^{t+1} - x^*)^\top E_s^\top$, $(x^{t+1} - x^*)^\top S$, and $(x^{t+1} - x^*)^\top E_u^\top$, we rewrite (56) as:

$$\begin{aligned} (x^{t+1} - x^*)^\top (\nabla F(x^t) - \nabla F(x^*)) &= \\ &= -\frac{2}{\mu_z} (\alpha^{t+1} - \alpha^t)^\top (\alpha^{t+1} - \alpha^*) \\ &\quad - \underbrace{(\theta^{t+1} - \theta^*)^\top (\lambda^{t+1} - \lambda^*)}_{\leq 0 \text{ from (52)}} - \frac{1}{\mu_\theta} (\lambda^{t+1} - \lambda^t)^\top (\lambda^{t+1} - \lambda^*) \\ &\quad - \underbrace{\mu_\theta (\theta^{t+1} - \theta^*)^\top (\theta^{t+1} - \theta^t)}_{\leq 0 \text{ from (51)}} - \underbrace{(\lambda^{t+1} - \lambda^t)^\top (\theta^{t+1} - \theta^t)}_{\leq 0 \text{ from (51)}} \\ &= (x^{t+1} - x^*)^\top \bar{J}^t (x^{t+1} - x^t) - 2\mu_z (z^{t+1} - z^*)^\top (z^{t+1} - z^t) \\ &\leq -\frac{2}{\mu_z} (\alpha^{t+1} - \alpha^t)^\top (\alpha^{t+1} - \alpha^*) \\ &\quad - \frac{1}{\mu_\theta} (\lambda^{t+1} - \lambda^t)^\top (\lambda^{t+1} - \lambda^*) - \mu_\theta (\theta^{t+1} - \theta^*)^\top (\theta^{t+1} - \theta^t) \\ &\quad - (x^{t+1} - x^*)^\top \bar{J}^t (x^{t+1} - x^t) - 2\mu_z (z^{t+1} - z^*)^\top (z^{t+1} - z^t) \\ &\stackrel{(i)}{=} \frac{1}{2} \left\{ \frac{2}{\mu_z} \left(\|\alpha^t - \alpha^*\|^2 - \|\alpha^{t+1} - \alpha^*\|^2 - \|\alpha^{t+1} - \alpha^t\|^2 \right) \right. \\ &\quad \left. + \frac{1}{\mu_\theta} \left(\|\lambda^t - \lambda^*\|^2 - \|\lambda^{t+1} - \lambda^*\|^2 - \|\lambda^{t+1} - \lambda^t\|^2 \right) \right. \\ &\quad \left. + \mu_\theta \left(\|\theta^t - \theta^*\|^2 - \|\theta^{t+1} - \theta^*\|^2 - \|\theta^{t+1} - \theta^t\|^2 \right) \right. \\ &\quad \left. + \|x^t - x^*\|_{\bar{J}^t}^2 - \|x^{t+1} - x^*\|_{\bar{J}^t}^2 - \|x^{t+1} - x^t\|_{\bar{J}^t}^2 \right. \\ &\quad \left. + 2\mu_z \left(\|z^t - z^*\|^2 - \|z^{t+1} - z^*\|^2 - \|z^{t+1} - z^t\|^2 \right) \right\} \\ &\stackrel{(ii)}{=} \frac{1}{2} \left(\|v_\alpha^t - v_\alpha^*\|_{\mathcal{G}^t}^2 - \|v_\alpha^{t+1} - v_\alpha^*\|_{\mathcal{G}^t}^2 - \|v_\alpha^{t+1} - v_\alpha^t\|_{\mathcal{G}^t}^2 \right), \end{aligned} \quad (57)$$

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 (54) 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:

$$\begin{aligned} (x^t - x^{t+1})^\top (\nabla F(x^t) - \nabla F(x^*)) &= \\ &\leq \frac{1}{M_f} \|\nabla F(x^t) - \nabla F(x^*)\|^2 + \frac{M_f}{4} \|x^{t+1} - x^t\|^2. \end{aligned} \quad (58)$$

After substituting (57) and (58) into (54), we obtain:

$$\begin{aligned} \frac{1}{M_f} \|\nabla F(x^t) - \nabla F(x^*)\|^2 &\leq \frac{1}{2} \left(\|v_\alpha^t - v_\alpha^*\|_{\mathcal{G}^t}^2 - \|v_\alpha^{t+1} - v_\alpha^*\|_{\mathcal{G}^t}^2 \right. \\ &\quad \left. - \|v_\alpha^{t+1} - v_\alpha^t\|_{\mathcal{G}^t}^2 \right) + \frac{1}{M_f} \|\nabla F(x^t) - \nabla F(x^*)\|^2 \\ &\quad + \frac{M_f}{4} \|x^{t+1} - x^t\|^2. \end{aligned}$$

By canceling the identical term and rearranging, we obtain:

$$\begin{aligned} \frac{1}{2} \left(\|v_\alpha^t - v_\alpha^*\|_{\mathcal{G}^t}^2 - \|v_\alpha^{t+1} - v_\alpha^*\|_{\mathcal{G}^t}^2 \right) &= \\ &\geq \frac{1}{2} \|v_\alpha^{t+1} - v_\alpha^t\|_{\mathcal{G}^t}^2 - \frac{M_f}{4} \|x^{t+1} - x^t\|^2 \\ &= \frac{1}{2} \left(\|x^{t+1} - x^t\|_{\bar{J}^t}^2 - \frac{M_f}{2} \|x^{t+1} - x^t\|^2 \right. \\ &\quad \left. + \mu_\theta \|\theta^{t+1} - \theta^t\|^2 + \frac{2}{\mu_z} \|\alpha^{t+1} - \alpha^t\|^2 + \frac{1}{\mu_\theta} \|\lambda^{t+1} - \lambda^t\|^2 \right). \end{aligned} \quad (59)$$

Recall $\bar{J}^t = J^t + \epsilon I$ and we proceed to find a uniform lower bound for $\|x^{t+1} - x^t\|_{\bar{J}^t}^2$, for gradient descent, Newton, and BFGS computing scheme. Since $J_{\text{Gradient}}^t = 0$ and $J_{\text{Newton}}^t = \nabla^2 F(x^t) \succeq 0$ by construction, it holds that $\epsilon I \preceq \bar{J}_{\text{Gradient/Newton}}^t$. It remains to find a lower bound for

the case of BFGS. Recall J_{BFGS}^t defined in (20). By the secant condition, $(J_{\text{BFGS}}^t + \mu_z D + \mu_\theta S S^\top + \epsilon I) s^{t-1} = q^{t-1}$, where $s^{t-1} = x^t - x^{t-1}$,

$$q^{t-1} = \nabla F(x^t) - \nabla F(x^{t-1}) + (\mu_z D + \mu_\theta S S^\top + \epsilon I) s^{t-1}.$$

Therefore, it holds that:

$$J_{\text{BFGS}}^t s^{t-1} = \nabla F(x^t) - \nabla F(x^{t-1}). \quad (60)$$

By premultiplying $(s^{t-1})^\top$ on both sides of (60), we obtain:

$$(s^{t-1})^\top J_{\text{BFGS}}^t s^{t-1} = (x^t - x^{t-1})^\top (\nabla F(x^t) - \nabla F(x^{t-1})) \geq 0.$$

Therefore, the following holds:

$$\sigma_{\min}^{\bar{J}} \|x^{t+1} - x^t\|^2 \leq \|x^{t+1} - x^t\|_{\bar{J}^t}^2,$$

where $\sigma_{\min}^{\bar{J}} = \epsilon$. By selecting $\epsilon > \frac{M_f}{2}$, we obtain:

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

We denote $\delta = \frac{\sigma_{\min}^{\bar{J}} - \frac{M_f}{2}}{\sigma_{\min}^{\bar{J}}}$ and since $\delta < 1$, it holds that:

$$\begin{aligned} & 2\mu_z \|z^{t+1} - z^t\|^2 + \mu_\theta \|\theta^{t+1} - \theta^t\|^2 + \frac{2}{\mu_z} \|\alpha^{t+1} - \alpha^t\|^2 \\ & + \frac{1}{\mu_\theta} \|\lambda^{t+1} - \lambda^t\|^2 \geq \delta \left(2\mu_z \|z^{t+1} - z^t\|^2 + \mu_\theta \|\theta^{t+1} - \theta^t\|^2 \right. \\ & \quad \left. + \frac{2}{\mu_z} \|\alpha^{t+1} - \alpha^t\|^2 + \frac{1}{\mu_\theta} \|\lambda^{t+1} - \lambda^t\|^2 \right). \end{aligned}$$

Therefore, (59) can be rewritten as:

$$\|v_\alpha^{t+1} - v_\alpha^*\|_{\mathcal{G}^t}^2 \leq \|v_\alpha^t - v_\alpha^*\|_{\mathcal{G}^t}^2 - \delta \|v_\alpha^{t+1} - v_\alpha^t\|_{\mathcal{G}^t}^2. \quad (61)$$

Since (61) shows that $\|v_\alpha^t - v_\alpha^*\|_{\mathcal{G}^t}^2$ is monotonically decreasing, it is therefore convergent. We proceed to show Part (ii).

Part (ii): Recall (47) and after rearranging, we obtain:

$$\begin{aligned} H^t(x^{t+1} - x^t) &= -\{\nabla F(x^t) + E_s^\top \alpha^t + S \lambda^t + \frac{\mu_z}{2} L_s x^t \\ & \quad + \mu_\theta S(S^\top x^t - \theta^t)\}. \end{aligned} \quad (62)$$

Since $H^t = J^t + \mu_z D + \epsilon I + \mu_\theta S S^\top$ as in (15), an upper bound $H^t \preceq \bar{M} I$ can be obtained by using (25) and (27):

$$\begin{aligned} \bar{M}_{\text{Gradient}} &= \mu_z d_{\max} + \epsilon + \mu_\theta, \\ \bar{M}_{\text{Newton}} &= M_f + \mu_z d_{\max} + \epsilon + \mu_\theta, \\ \bar{M}_{\text{BFGS}} &= \psi, \end{aligned}$$

where $d_{\max} = \max_i |\mathcal{N}_i|$ denotes the maximum degree. Therefore, the following holds:

$$\bar{M}^2 \|x^{t+1} - x^t\|^2 \geq \|x^{t+1} - x^t\|_{(H^t)^2}^2. \quad (63)$$

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

$$\begin{aligned} \|x^{t+1} - x^t\|_{\bar{J}^t}^2 &\geq \sigma_{\min}^{\bar{J}} \|x^{t+1} - x^t\|^2 \stackrel{(i)}{\geq} \frac{\sigma_{\min}^{\bar{J}}}{\bar{M}^2} \|x^{t+1} - x^t\|_{(H^t)^2}^2 \\ &\stackrel{(ii)}{=} \frac{\sigma_{\min}^{\bar{J}}}{\bar{M}^2} \left\| \nabla F(x^t) + E_s^\top \alpha^t + S \lambda^t + \frac{\mu_z}{2} L_s x^t + \mu_\theta S(x_i^t - \theta^t) \right\|^2 \\ &\stackrel{(iii)}{\geq} \frac{\sigma_{\min}^{\bar{J}}}{\bar{M}^2} \left(\frac{1}{\rho} \|\nabla F(x^t) + E_s^\top \alpha^t + S \lambda^t\|^2 \right. \\ & \quad \left. - \frac{1}{\rho-1} \left\| \frac{\mu_z}{2} L_s x^t + \mu_\theta S(S^\top x^t - \theta^t) \right\|^2 \right) \\ &\stackrel{(iv)}{\geq} \frac{\sigma_{\min}^{\bar{J}}}{\bar{M}^2} \left(\frac{1}{\rho} \|\nabla F(x^t) + E_s^\top \alpha^t + S \lambda^t\|^2 - \frac{2}{\rho-1} \left\| \frac{\mu_z}{2} L_s x^t \right\|^2 \right. \\ & \quad \left. - \frac{2}{\rho-1} \|\mu_\theta (x_i^t - \theta^t)\|^2 \right), \end{aligned} \quad (64)$$

where (i) follows from (63); (ii) follows from (62); (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 $\|\mu_\theta S(S^\top x^t - \theta^t)\| = \|\mu_\theta (x_i^t - \theta^t)\|$ by definition of $S = s_l \otimes I_d$ being the selection matrix. Further observe that the following holds due to dual updates (14c) and (14d):

$$\begin{aligned} \alpha^{t+1} - \alpha^t &= \frac{\mu_z}{2} E_s x^{t+1}, \\ \lambda^{t+1} - \lambda^t &= \mu_\theta (S^\top x^{t+1} - \theta^{t+1}). \end{aligned}$$

Therefore, we obtain the following:

$$\frac{2}{\mu_z} \|\alpha^{t+1} - \alpha^t\|^2 = \frac{\mu_z}{2} \|E_s x^{t+1}\|^2 = \frac{\mu_z}{2} \|x^{t+1}\|_{L_s}^2, \quad (65)$$

$$\frac{1}{\mu_\theta} \|\lambda^{t+1} - \lambda^t\|^2 = \mu_\theta \|x_i^{t+1} - \theta^{t+1}\|^2, \quad (66)$$

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

$$\frac{\sigma_{\min}^{\bar{J}}}{\bar{M}^2} \frac{2}{\rho-1} \left\| \frac{\mu_z}{2} L_s x^t \right\|^2 \leq \frac{\sigma_{\min}^{\bar{J}} \mu_z^2}{2\bar{M}^2 \sigma_{\max}^{L_s}} \|x^t\|_{(L_s)^2}^2 \leq \frac{\sigma_{\min}^{\bar{J}} \mu_z^2}{2\bar{M}^2} \|x^t\|_{L_s}^2. \quad (67)$$

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

$$\begin{aligned} & \frac{1}{T} \frac{\mu_z}{2} \|x^1\|_{L_s}^2 + \frac{\mu_\theta}{T} \|x_l^1 - \theta^1\|^2 + \frac{1}{T} \sum_{t=1}^T \|v_\alpha^{t+1} - v_\alpha^t\|_{\mathcal{G}^t}^2 \\ &= \frac{1}{T} \frac{\mu_z}{2} \|x^1\|_{L_s}^2 + \frac{\mu_\theta}{T} \|x_l^1 - \theta^1\|^2 + \frac{1}{T} \sum_{t=1}^T \left(\|x^{t+1} - x^t\|_{\bar{J}^t}^2 \right. \\ & \quad \left. + 2\mu_z \|z^{t+1} - z^t\|^2 + \mu_\theta \|\theta^{t+1} - \theta^t\|^2 + \frac{2}{\mu_z} \|\alpha^{t+1} - \alpha^t\|^2 \right. \\ & \quad \left. + \frac{1}{\mu_\theta} \|\lambda^{t+1} - \lambda^t\|^2 \right) \stackrel{(i)}{\geq} \frac{1}{T} \frac{\mu_z}{2} \|x^{T+1}\|_{L_s}^2 + \frac{\mu_\theta}{T} \|x_l^{T+1} - \theta^{T+1}\|^2 \\ & \quad + \frac{1}{T} \sum_{t=1}^T \left(\frac{\sigma_{\min}^{\bar{J}}}{\bar{M}^2 \rho} \|\nabla F(x^t) + E_s^\top \alpha^t + S \lambda^t\|^2 + 2\mu_z \|z^{t+1} - z^t\|^2 \right. \\ & \quad \left. + \mu_\theta \|\theta^{t+1} - \theta^t\|^2 + \left(\frac{\mu_z}{2} - \frac{\sigma_{\min}^{\bar{J}} \mu_z^2}{2\bar{M}^2} \right) \|x^t\|_{L_s}^2 \right. \\ & \quad \left. + \left(\mu_\theta - \frac{2\sigma_{\min}^{\bar{J}} \mu_\theta^2}{\bar{M}^2 (\rho-1)} \right) \|x_l^t - \theta^t\|^2 \right) \end{aligned}$$

where (i) follows from substituting (64)-(67). All coefficients are ensured to be positive by selecting: $\mu_z \epsilon < \psi^2$, and $\rho > \max \left\{ \frac{2\sigma_{\min}^{\bar{J}} \mu_\theta}{\bar{M}^2}, \sigma_{\max}^{L_s} \right\} + 1$, where $\sigma_{\min}^{\bar{J}} = \epsilon$.

Proof of Corollary 1: Following Theorem 1 and standard analysis techniques in [53] and [54], we obtain that $\|v_\alpha^t - v_\alpha^*\| \rightarrow 0$ as $t \rightarrow \infty$. After taking telescoping sum from $t = 1$ to ∞ on both sides of (61), we obtain:

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

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

APPENDIX C

Proof of Lemma 4: 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:

$$\|e^t\| \leq \|\nabla F(x^t) - \nabla F(x^{t+1})\| + \|J^t\| \|x^{t+1} - x^t\|. \quad (68)$$

In the case of gradient updates, $J^t = 0$. Therefore,

$$\|e_{\text{Gradient}}^t\| \leq \|\nabla F(x^t) - \nabla F(x^{t+1})\| \leq M_f \|x^{t+1} - x^t\|,$$

where the last inequality follows from Assumption 2. Setting $\tau_{\text{Gradient}}^t = M_f$, we obtain (32a). In the case of Newton updates, $J^t = \nabla^2 F(x^t)$. By Assumption 2 and (68), we obtain:

$$\|e^t\| \leq 2M_f \|x^{t+1} - x^t\|. \quad (69)$$

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

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

By adding and subtracting $\int_0^1 \nabla^2 F(x^t)(x^{t+1} - x^t) ds$, we further obtain:

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

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

$$\begin{aligned} \|\nabla F(x^{t+1}) - \nabla F(x^t) - \nabla^2 F(x^t)(x^{t+1} - x^t)\| &= \\ \left\| \int_0^1 (\nabla^2 F(sx^{t+1} + (1-s)x^t) - \nabla^2 F(x^t))(x^{t+1} - x^t) ds \right\| &\leq \\ \int_0^1 \|\nabla^2 F(sx^{t+1} + (1-s)x^t) - \nabla^2 F(x^t)\| \cdot \|x^{t+1} - x^t\| ds &\leq \\ \int_0^1 sL_f \|x^{t+1} - x^t\|^2 ds = \frac{L_f}{2} \|x^{t+1} - x^t\|^2. \end{aligned}$$

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 (69) and the above, we obtain: $\|e^t\| \leq \tau_{\text{Newton}}^t \|x^{t+1} - x^t\|$, where τ_{Newton}^t is defined in (32b). We proceed to establish (32c). Recall the definition of J_{BFGS}^t in (20):

$$J_{\text{BFGS}}^t = H_{\text{BFGS}}^t - \mu_z D - \mu_\theta S S^\top - \epsilon I_{md}. \quad (70)$$

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:

$$\begin{aligned} s^t &= x^{t+1} - x^t, \\ q^t &= \nabla F(x^{t+1}) - \nabla F(x^t) + (\mu_z D + \mu_\theta S S^\top + \epsilon I)s^t. \end{aligned}$$

From the secant condition, it holds that:

$$\begin{aligned} \nabla F(x^t) - \nabla F(x^{t+1}) &= \\ - (H^{t+1} - \mu_z D - \mu_\theta S S^\top - \epsilon I)(x^{t+1} - x^t). \end{aligned}$$

Using (70) and the expression for $\nabla F(x^t) - \nabla F(x^{t+1})$ into (28), we obtain:

$$\begin{aligned} \|e^t\| &= \|(H^t - H^{t+1})(x^{t+1} - x^t)\| \\ &\leq \|H^t - H^{t+1}\| \|x^{t+1} - x^t\|. \end{aligned}$$

Denoting $\tau_{\text{BFGS}}^t = \|H^t - H^{t+1}\|$ and using (27), we obtain (32c). ■

The following Lemma that will be useful for establishing Theorem 2.

Lemma 5. Recall $C := \begin{bmatrix} E_s \\ S^\top \end{bmatrix}$ and $\phi^t = E_s^\top \alpha^t$ in (14). Denote the smallest positive eigenvalue of CC^\top as σ_{\min}^+ and consider the unique dual optimal pair (α^*, λ^*) that lies in the column space of C as established in Lemma 2. The following holds:

$$\begin{aligned} \sigma_{\min}^+ \left(\|\alpha^{t+1} - \alpha^*\|^2 + \|\lambda^{t+1} - \lambda^*\|^2 \right) &= \\ \leq \|E_s^\top (\alpha^{t+1} - \alpha^*) + S(\lambda^{t+1} - \lambda^*)\|^2. \end{aligned} \quad (71)$$

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

$$\begin{bmatrix} \alpha^{t+1} \\ \lambda^{t+1} \end{bmatrix} = \begin{bmatrix} \alpha^t \\ \lambda^t \end{bmatrix} + \begin{bmatrix} \frac{\mu_z}{2} E_s \\ \mu_\theta S^\top \end{bmatrix} x^{t+1} - \begin{bmatrix} 0 \\ \mu_\theta I_d \end{bmatrix} \theta^{t+1}.$$

We show that the column space of $M := \begin{bmatrix} 0 \\ \mu_\theta I_d \end{bmatrix}$ belongs in

the column space of $N := \begin{bmatrix} \frac{\mu_z}{2} E_s \\ \mu_\theta S^\top \end{bmatrix}$. Consider fixed $r^x \in \mathbb{R}^d$.

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

$$\begin{bmatrix} \frac{\mu_z}{2} E_s \\ \mu_\theta S^\top \end{bmatrix} r^y = \begin{bmatrix} 0 \\ \mu_\theta r_i^y \end{bmatrix} = \begin{bmatrix} 0 \\ \mu_\theta I_d \end{bmatrix} r^x.$$

, which shows $\text{col}(M) \subset \text{col}(N)$. By choosing $\mu_z = 2\mu_\theta$, we conclude that $[\alpha^{t+1} - \alpha^*; \lambda^{t+1} - \lambda^*]$ lies in the column space of C . ■

Proof of Theorem 2: Using Lemma 3, we obtain:

$$\begin{aligned} \nabla F(x^{t+1}) - \nabla F(x^*) &= -(E_s^\top (\alpha^{t+1} - \alpha^*) + \epsilon(x^{t+1} - x^t) \\ &+ S(\lambda^{t+1} - \lambda^* + \mu_\theta(\theta^{t+1} - \theta^t))) + e^t + \mu_z E_u^\top (z^{t+1} - z^t), \end{aligned}$$

Since $F(x)$ is strongly convex with Lipschitz continuous gradient, the following inequality holds [50]:

$$\begin{aligned} \frac{m_f M_f}{m_f + M_f} \|x^{t+1} - x^*\|^2 + \frac{1}{m_f + M_f} \|\nabla F(x^{t+1}) - \nabla F(x^*)\|^2 &\leq \\ (x^{t+1} - x^*)^\top (\nabla F(x^{t+1}) - \nabla F(x^*)) &\leq \\ -(x^{t+1} - x^*)^\top e^t - \epsilon(x^{t+1} - x^*)^\top (x^{t+1} - x^t) & \\ -(x^{t+1} - x^*) E_s^\top (\alpha^{t+1} - \alpha^*) - (x^{t+1} - x^*)^\top S (\lambda^{t+1} - \lambda^* & \\ + \mu_\theta(\theta^{t+1} - \theta^t)) - \mu_z (x^{t+1} - x^*)^\top E_u^\top (z^{t+1} - z^t), & \end{aligned}$$

where the last inequality follows from substituting the expression of $\nabla F(x^{t+1}) - \nabla F(x^*)$ above. Using similar techniques used in deriving (54)-(57), we obtain

$$\begin{aligned}
& \frac{2m_f M_f}{m_f + M_f} \|x^{t+1} - x^*\|^2 + \frac{2}{m_f + M_f} \|\nabla F(x^{t+1}) - \nabla F(x^*)\|^2 \\
& \leq \epsilon (\|x^t - x^*\|^2 - \|x^{t+1} - x^*\|^2 - \|x^{t+1} - x^t\|^2) \\
& + 2\mu_z (\|z^t - z^*\|^2 - \|z^{t+1} - z^*\|^2 - \|z^{t+1} - z^t\|^2) \\
& + \frac{1}{\mu_\theta} (\|\lambda^t - \lambda^*\|^2 - \|\lambda^{t+1} - \lambda^*\|^2 - \|\lambda^{t+1} - \lambda^t\|^2) \\
& + \mu_\theta (\|\theta^t - \theta^*\|^2 - \|\theta^{t+1} - \theta^*\|^2 - \|\theta^{t+1} - \theta^t\|^2) \\
& + \frac{1}{\mu_z} (\|\alpha^{t+1} - \alpha^*\|^2 - \|\alpha^{t+1} - \alpha^*\|^2 - \|\alpha^{t+1} - \alpha^t\|^2) \\
& - 2(x^{t+1} - x^*)^\top e^t \\
& = \|v_\alpha^t - v_\alpha^*\|_{\mathcal{H}}^2 - \|v_\alpha^{t+1} - v_\alpha^*\|_{\mathcal{H}}^2 - \|v_\alpha^{t+1} - v_\alpha^t\|_{\mathcal{H}}^2 \\
& - 2(x^{t+1} - x^*)^\top e^t, \tag{72}
\end{aligned}$$

Therefore, we obtain:

$$\begin{aligned}
& \frac{2m_f M_f}{m_f + M_f} \|x^{t+1} - x^*\|^2 + \frac{2}{m_f + M_f} \|\nabla F(x^{t+1}) - \nabla F(x^*)\|^2 \\
& + \|v_\alpha^{t+1} - v_\alpha^t\|_{\mathcal{H}}^2 + 2(x^{t+1} - x^*)^\top e^t \\
& \leq \|v_\alpha^t - v_\alpha^*\|_{\mathcal{H}}^2 - \|v_\alpha^{t+1} - v_\alpha^*\|_{\mathcal{H}}^2. \tag{73}
\end{aligned}$$

To establish linear convergence, we need to show the following holds for some $\eta > 0$:

$$\eta \|v_\alpha^{t+1} - v_\alpha^*\|_{\mathcal{H}}^2 \leq \|v_\alpha^t - v_\alpha^*\|_{\mathcal{H}}^2 - \|v_\alpha^{t+1} - v_\alpha^*\|_{\mathcal{H}}^2. \tag{74}$$

We expand the expression of $\eta \|v_\alpha^{t+1} - v_\alpha^*\|_{\mathcal{H}}^2$ as follows:

$$\begin{aligned}
& \eta \|v_\alpha^{t+1} - v_\alpha^*\|_{\mathcal{H}}^2 = \eta \left(\epsilon \|x^{t+1} - x^*\|^2 + 2\mu_z \|z^{t+1} - z^*\|^2 \right. \\
& \left. + \frac{2}{\mu_z} \|\alpha^{t+1} - \alpha^*\|^2 + \mu_\theta \|\theta^{t+1} - \theta^*\|^2 + \frac{1}{\mu_\theta} \|\lambda^{t+1} - \lambda^*\|^2 \right). \tag{75}
\end{aligned}$$

We proceed to establish an upper bound for each component of (75). From Lemma 3, the following holds:

$$\begin{aligned}
& E_s^\top (\alpha^{t+1} - \alpha^*) + S(\lambda^{t+1} - \lambda^*) = -\left\{ \nabla F(x^{t+1}) - \nabla F(x^*) \right. \\
& \left. + \epsilon(x^{t+1} - x^t) + \mu_z E_u^\top (z^{t+1} - z^t) + \mu_\theta S(\theta^{t+1} - \theta^t) + e^t \right\}.
\end{aligned}$$

Then we obtain:

$$\begin{aligned}
& \sigma_{\min}^+ \left(\|\alpha^{t+1} - \alpha^*\|^2 + \|\lambda^{t+1} - \lambda^*\|^2 \right) \\
& \stackrel{(i)}{\leq} \|E_s^\top (\alpha^{t+1} - \alpha^*) + S(\lambda^{t+1} - \lambda^*)\|^2 \\
& \stackrel{(ii)}{\leq} 5 \left(\|\nabla F(x^{t+1}) - \nabla F(x^*)\|^2 + \epsilon^2 \|x^{t+1} - x^t\|^2 \right. \\
& \left. + \mu_\theta^2 \|\theta^{t+1} - \theta^t\|^2 + \|e^t\|^2 + \sigma_{\max}^{Lu} \mu_z^2 \|z^{t+1} - z^t\|^2 \right), \tag{76}
\end{aligned}$$

where (i) follows from Lemma 5; (ii) follows from the inequality $(\sum_{i=1}^n a_i)^2 \leq \sum_{i=1}^n n a_i^2$. Recalling that we have selected $\mu_z = 2\mu_\theta$, we obtain:

$$\begin{aligned}
& \frac{2}{\mu_z} \|\alpha^{t+1} - \alpha^*\|^2 + \frac{1}{\mu_\theta} \|\lambda^{t+1} - \lambda^*\|^2 \\
& = \frac{1}{\mu_\theta} \left(\|\alpha^{t+1} - \alpha^*\|^2 + \|\lambda^{t+1} - \lambda^*\|^2 \right) \\
& \stackrel{(i)}{\leq} \frac{5}{\mu_\theta \sigma_{\min}^+} \left(\|\nabla F(x^{t+1}) - \nabla F(x^*)\|^2 + \epsilon^2 \|x^{t+1} - x^t\|^2 \right. \\
& \left. + \mu_\theta^2 \|\theta^{t+1} - \theta^t\|^2 + \|e^t\|^2 + \sigma_{\max}^{Lu} \mu_z^2 \|z^{t+1} - z^t\|^2 \right), \tag{77}
\end{aligned}$$

where (i) follows from dividing (76) by σ_{\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:

$$2\mu_z \|z^{t+1} - z^*\|^2 \leq \frac{\mu_z \sigma_{\max}^{Lu}}{2} \|x^{t+1} - x^*\|^2.$$

Using the upper bound for $2\mu_z \|z^{t+1} - z^*\|^2$, the inequality (77), and $\mu_\theta \|\theta^{t+1} - \theta^*\|^2 \leq 2\mu_\theta \|x^{t+1} - x^*\|^2 + \frac{2}{\mu_\theta} \|\lambda^{t+1} - \lambda^t\|^2$ from (14d) and KKTd, we obtain an upper bound for (75) as:

$$\begin{aligned}
& \eta \|v_\alpha^{t+1} - v_\alpha^*\|_{\mathcal{H}}^2 \leq \eta \left\{ \frac{5}{\mu_\theta \sigma_{\min}^+} \left(\|\nabla F(x^{t+1}) - \nabla F(x^*)\|^2 \right. \right. \\
& \left. \left. + \epsilon^2 \|x^{t+1} - x^t\|^2 + \mu_\theta^2 \|\theta^{t+1} - \theta^t\|^2 + \|e^t\|^2 \right. \right. \\
& \left. \left. + \sigma_{\max}^{Lu} \mu_z^2 \|z^{t+1} - z^t\|^2 \right) + \frac{2}{\mu_\theta} \|\lambda^{t+1} - \lambda^t\|^2 \right. \\
& \left. + (\epsilon + 2\mu_\theta + \frac{\mu_z \sigma_{\max}^{Lu}}{2}) \|x^{t+1} - x^*\|^2 \right\}.
\end{aligned}$$

Recall that the right-hand side of (74) is lower bounded as in (73). Therefore, it suffices to prove the following to establish (74):

$$\begin{aligned}
& \eta \left\{ \frac{5}{\mu_\theta \sigma_{\min}^+} \left(\|\nabla F(x^{t+1}) - \nabla F(x^*)\|^2 + \epsilon^2 \|x^{t+1} - x^t\|^2 \right. \right. \\
& \left. \left. + \mu_\theta^2 \|\theta^{t+1} - \theta^t\|^2 + \|e^t\|^2 + \sigma_{\max}^{Lu} \mu_z^2 \|z^{t+1} - z^t\|^2 \right) \right. \\
& \left. + \frac{2}{\mu_\theta} \|\lambda^{t+1} - \lambda^t\|^2 + (\epsilon + 2\mu_\theta + \frac{\mu_z \sigma_{\max}^{Lu}}{2}) \|x^{t+1} - x^*\|^2 \right\} \\
& \leq \frac{2m_f M_f}{m_f + M_f} \|x^{t+1} - x^*\|^2 + \frac{2}{m_f + M_f} \|\nabla F(x^{t+1}) - \nabla F(x^*)\|^2 \\
& + \|v_\alpha^{t+1} - v_\alpha^t\|_{\mathcal{H}}^2 + 2(x^{t+1} - x^*)^\top e^t, \tag{78}
\end{aligned}$$

Note that $-\zeta \|e^t\|^2 - \frac{1}{\zeta} \|x^{t+1} - x^*\|^2 \leq 2(x^{t+1} - x^*)^\top e^t$ holds for any $\zeta > 0$. To prove (78), it is therefore sufficient to show:

$$\begin{aligned}
& \zeta (\tau^t)^2 \|x^{t+1} - x^t\|^2 + \eta \left\{ \frac{5}{\mu_\theta \sigma_{\min}^+} \left(\|\nabla F(x^{t+1}) - \nabla F(x^*)\|^2 \right. \right. \\
& \left. \left. + ((\tau^t)^2 + \epsilon^2) \|x^{t+1} - x^t\|^2 + \mu_\theta^2 \|\theta^{t+1} - \theta^t\|^2 \right. \right. \\
& \left. \left. + \sigma_{\max}^{Lu} \mu_z^2 \|z^{t+1} - z^t\|^2 \right) + \frac{2}{\mu_\theta} \|\lambda^{t+1} - \lambda^t\|^2 \right. \\
& \left. + (\epsilon + 2\mu_\theta + \frac{\mu_z \sigma_{\max}^{Lu}}{2}) \|x^{t+1} - x^*\|^2 \right\} \\
& \leq \left(\frac{2m_f M_f}{m_f + M_f} - \frac{1}{\zeta} \right) \|x^{t+1} - x^*\|^2 + \epsilon \|x^{t+1} - x^t\|^2 \\
& + 2\mu_z \|z^{t+1} - z^t\|^2 + \frac{2}{\mu_z} \|\alpha^{t+1} - \alpha^t\|^2 + \mu_\theta \|\theta^{t+1} - \theta^t\|^2 \\
& + \frac{1}{\mu_\theta} \|\lambda^{t+1} - \lambda^t\|^2 + \frac{2}{m_f + M_f} \|\nabla F(x^{t+1}) - \nabla F(x^*)\|^2
\end{aligned} \tag{79}$$

where we have used $\|e^t\|^2 \leq (\tau^t)^2 \|x^{t+1} - x^t\|^2$ from Lemma 4. Establishing (79) 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 η as in (33), we establish (79). Therefore, the inequality (74) holds, which equivalently establishes the linear convergence rate. ■

Proof of Theorem 3: The proof proceeds as follows:

$$\begin{aligned}
& \|v_\alpha^{t+1} - v_\alpha^*\|_{\mathcal{H}\Omega_\alpha^{-1}}^2 = \|v_\alpha^t + \Omega_\alpha^{t+1} (T v_\alpha^t - v_\alpha^t) - v_\alpha^*\|_{\mathcal{H}\Omega_\alpha^{-1}}^2 \\
& = \|v_\alpha^t - v_\alpha^*\|_{\mathcal{H}\Omega_\alpha^{-1}}^2 + 2(v_\alpha^t - v_\alpha^*)^\top \mathcal{H}\Omega_\alpha^{-1} \Omega_\alpha^{t+1} (T v_\alpha^t - v_\alpha^t) \\
& + (T v_\alpha^t - v_\alpha^t)^\top \Omega_\alpha^{t+1} \mathcal{H}\Omega_\alpha^{-1} \Omega_\alpha^{t+1} (T v_\alpha^t - v_\alpha^t), \tag{80}
\end{aligned}$$

Since Ω_α^{t+1} , Ω_α^{-1} , and \mathcal{H} are all diagonal matrices, they commute with each other. Moreover, since each sub-block of

Ω_α^{t+1} is I_d or 0, it holds that $\Omega_\alpha^{t+1}\Omega_\alpha^{t+1} = \Omega_\alpha^{t+1}$. After taking conditional expectation on both sides of (80), we obtain:

$$\begin{aligned} \mathbb{E}^t \left[\|v_\alpha^{t+1} - v_\alpha^*\|_{\mathcal{H}\Omega_\alpha^{-1}}^2 \right] &= \|v_\alpha^t - v_\alpha^*\|_{\mathcal{H}\Omega_\alpha^{-1}}^2 + \|Tv_\alpha^t - v_\alpha^t\|_{\mathcal{H}}^2 \\ &\quad + 2(v_\alpha^t - v_\alpha^*)^\top \mathcal{H}(Tv_\alpha^t - v_\alpha^t) \stackrel{(i)}{\leq} \\ &\|v_\alpha^t - v_\alpha^*\|_{\mathcal{H}\Omega_\alpha^{-1}}^2 - \frac{\eta}{1+\eta} \|v_\alpha^t - v_\alpha^*\|_{\mathcal{H}}^2 \stackrel{(ii)}{\leq} \\ &\left(1 - \frac{\rho^{\min} \eta}{1+\eta}\right) \|v_\alpha^t - v_\alpha^*\|_{\mathcal{H}\Omega_\alpha^{-1}}^2, \end{aligned}$$

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

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 $i < j$. For each agent i , we 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 Ω^{t+1} . Recall $\alpha_k^{t+1} = \alpha_k^t + \frac{\mu_z}{2} (x_i^{t+1} - x_j^{t+1})$. The dual updates are described by:

$$\begin{aligned} \phi_i^{t+1} &= \phi_i^t + \frac{\mu_z}{2} X_{ii}^{t+1} \sum_{j \in \mathcal{N}_i} (x_i^{t+1} - x_j^{t+1}) \\ &= \phi_i^t + X_{ii}^{t+1} \left\{ \sum_{k \in \mathcal{P}_i} (\alpha_k^{t+1} - \alpha_k^t) + \sum_{k \in \mathcal{S}_i} (\alpha_k^t - \alpha_k^{t+1}) \right\}. \end{aligned}$$

Therefore, if $X_{ii}^{t+1} = I_d$, then $Y_{kk}^{t+1} = I_d$ for $k \in \mathcal{P}_i \cup \mathcal{S}_i$ for the corresponding Ω_α^{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:

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

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

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

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

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

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] W. Ananduta, A. Nedić, and C. Ocampo-Martinez, "Distributed augmented lagrangian method for link-based resource sharing problems of multiagent systems," *IEEE Transactions on Automatic Control*, vol. 67, no. 6, pp. 3067–3074, 2022.
- [6] 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.
- [7] C. Saunders, A. Gammernan, and V. Vovk, "Ridge regression learning algorithm in dual variables," in *Proceedings of the International Conference on Machine Learning*, 1998, p. 515–521.
- [8] 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.
- [9] 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.
- [10] —, "A proximal gradient algorithm for decentralized composite optimization," *IEEE Transactions on Signal Processing*, vol. 63, no. 22, pp. 6013–6023, 2015.
- [11] 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.
- [12] S. A. Alghunaim, K. Yuan, and A. H. Sayed, "A linearly convergent proximal gradient algorithm for decentralized optimization," in *Proceedings of the International Conference on Neural Information Processing Systems*, 2019.
- [13] 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.
- [14] 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.
- [15] R. Xin, U. A. Khan, and S. Kar, "A fast randomized incremental gradient method for decentralized non-convex optimization," *IEEE Transactions on Automatic Control*, 2021.
- [16] 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.
- [17] 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.
- [18] 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.
- [19] 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.
- [20] J. Nocedal and S. J. Wright, *Numerical Optimization*. Springer, 2006.
- [21] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*. Prentice-Hall, Inc., 1989.
- [22] 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.
- [23] 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.
- [24] 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.
- [25] E. Wei and A. Ozdaglar, "Distributed alternating direction method of multipliers," in *IEEE Conference on Decision and Control*, 2012, pp. 5445–5450.
- [26] 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.
- [27] 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.

- [28] 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.
- [29] 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.
- [30] 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.
- [31] 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.
- [32] 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.
- [33] —, "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.
- [34] 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.
- [35] 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.
- [36] 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.
- [37] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y. Arcas, "Communication-Efficient Learning of Deep Networks from Decentralized Data," in *Proceedings of the International Conference on Artificial Intelligence and Statistics*, 2017, pp. 1273–1282.
- [38] Y. Gong, Y. Li, and N. M. Freris, "FedADMM: A robust federated deep learning framework with adaptivity to system heterogeneity," in *IEEE International Conference on Data Engineering*, 2022, pp. 2575–2587.
- [39] 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.
- [40] N. Parikh and S. Boyd, "Proximal algorithms," *Foundations and Trends in Optimization*, vol. 1, no. 3, p. 127–239, 2014.
- [41] C. G. Broyden, "The convergence of a class of double-rank minimization algorithms 1. general considerations," *IMA Journal of Applied Mathematics*, vol. 6, no. 1, pp. 76–90, 1970.
- [42] R. Fletcher, "A new approach to variable metric algorithms," *The Computer Journal*, vol. 13, no. 3, pp. 317–322, 1970.
- [43] D. Goldfarb, "A family of variable-metric methods derived by variational means," *Mathematics of Computation*, pp. 23–26, 1970.
- [44] D. F. Shanno, "Conditioning of quasi-Newton methods for function minimization," *Mathematics of Computation*, vol. 24, no. 111, pp. 647–656, 1970.
- [45] Y. Li, N. M. Freris, P. Voulgaris, and D. Stipanović, "DN-ADMM: Distributed newton admm for multi-agent optimization," in *IEEE Conference on Decision and Control*, 2021, pp. 3343–3348.
- [46] —, "D-SOP: Distributed second order proximal method for convex composite optimization," in *American Control Conference*, 2020, pp. 2844–2849.
- [47] M. Eisen, A. Mokhtari, and A. Ribeiro, "Decentralized quasi-Newton methods," *IEEE Transactions on Signal Processing*, vol. 65, no. 10, pp. 2613–2628, 2017.
- [48] 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.
- [49] 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.
- [50] Y. Nesterov, *Lectures on Convex Optimization*. Springer Publishing Company, 2018.
- [51] Y. Li, P. G. Voulgaris, and N. M. Freris, "A communication efficient quasi-newton method for large-scale distributed multi-agent optimization," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2022, pp. 4268–4272.
- [52] A. Mokhtari and A. Ribeiro, "RES: Regularized stochastic BFGS algorithm," *IEEE Transactions on Signal Processing*, vol. 62, no. 23, pp. 6089–6104, 2014.
- [53] B. He, "A new method for a class of linear variational inequalities," *Mathematical Programming*, p. 137–144, 1994.
- [54] W. Deng, M.-J. Lai, Z. Peng, and W. Yin, "Parallel multi-block ADMM with $\mathcal{O}(1/k)$ convergence," *Journal of Scientific Computing*, p. 712–736, 2017.



Yichuan Li received the B.S. degree in 2016, the M.S. degree in Mechanical Engineering in 2018, the M.S. degree in Applied Mathematics, and the Ph.D. degree in Mechanical Engineering in 2022, all from the University of Illinois at Urbana-Champaign, Champaign, IL, USA. His research interests include multi-agent optimization, distributed machine learning, and control.



Professor Petros G. Voulgaris received the Diploma in Mechanical Engineering from the National Technical University, Athens, Greece, in 1986, and the S.M. and Ph.D. degrees in Aeronautics and Astronautics from the Massachusetts Institute of Technology in 1988 and 1991, respectively. He is currently Chair, Founding Aerospace Program Director, and Victor LaMar Lockhart Professor in Mechanical Engineering at University of Nevada, Reno. Before joining UNR in 2020 and since 1991, he has been a faculty with the Department of Aerospace Engineering, University of Illinois at Urbana-Champaign holding also appointments with the Coordinated Science Laboratory, and the department of Electrical and Computer Engineering. His research interests are in the general area of robust and optimal control and coordination of autonomous systems. Dr. Voulgaris is a recipient of several awards including the NSF Research Initiation Award, the ONR Young Investigator Award and the UIUC Xerox Award for research. He has also been a Visiting ADGAS Chair Professor, Mechanical Engineering, Petroleum Institute, Abu Dhabi, UAE and a Visiting Gaungbiao Chair at Zhejiang University, China. His research has been supported by several agencies including NSF, ONR, AFOSR, NASA. He is also a Fellow of IEEE.



Dr. Dušan M. Stipanović received his B.S. degree in electrical engineering from the University of Belgrade, Belgrade, Serbia, in 1994, and the M.S.E.E. and Ph.D. degrees (under supervision of Professor Dragoslav Šiljak) in electrical engineering from Santa Clara University, Santa Clara, California, in 1996 and 2000, respectively. Dr. Stipanović had been an Adjunct Lecturer and Research Associate with the Department of Electrical Engineering at Santa Clara University (1998-2001), and a Research Associate in Professor Claire Tomlin's Hybrid Systems

Laboratory of the Department of Aeronautics and Astronautics at Stanford University (2001-2004). In 2004 he joined the University of Illinois at Urbana-Champaign where he is now Professor in the Controls Group of the Coordinated Science Laboratory and Department of Industrial and Enterprise Systems Engineering. Dr. Stipanović served as an Associate Editor on the Editorial Boards of the *IEEE Transactions on Circuits and Systems I and II*. Currently he is an Associate Editor for *Journal of Optimization Theory and Applications*.



Nikolaos M. Freris (Senior Member, IEEE) received the Diploma in ECE from the National Technical University of Athens (NTUA), Athens, Greece, in 2005, the M.S. degree in ECE, the M.S. degree in Mathematics, and the Ph.D. degree in ECE, all from the University of Illinois at Urbana-Champaign (UIUC), Champaign, IL, USA, in 2007, 2008, and 2010, respectively. He is a Professor with the School of Computer Science and Technology and the Vice Dean of the International College at the University of Science and Technology of China (USTC), Hefei,

China. His research lies in AIoT/CPS/IoT: machine learning, distributed optimization, data mining, wireless networks, control, and signal processing, with applications in power systems, sensor networks, transportation, cyber security, and robotics. Dr. Freris has published several papers in high-profile conferences and journals held by IEEE, ACM, and SIAM, and he holds three patents. His research has been sponsored by the Ministry of Science and Technology of China, Anhui Dept. of Science and Technology, Tencent, and NSF, and was recognized with the USTC Alumni Foundation Innovation Scholar award, the IBM High Value Patent award, two IBM invention achievement awards, and the Gerondelis foundation award. Previously, he was with the faculty of NYU and, before that, he held senior researcher and postdoctoral researcher positions at EPFL and IBM Research, respectively. Dr. Freris is a Senior Member of ACM and IEEE, and a member of CCF and SIAM.