DECENTRALIZED OPTIMIZATION WITH COUPLED CONSTRAINTS

Anonymous authors
Paper under double-blind review

ABSTRACT

We consider the decentralized minimization of a separable objective $\sum_{i=1}^{n} f_i(x_i)$, where the variables are coupled through an affine constraint $\sum_{i=1}^{n} (\mathbf{A}_i x_i - b_i) = 0$. We assume that the functions f_i, matrices $\mathbf{A}_i$, and vectors b_i are stored locally by the nodes of a computational network, and that the functions f_i are smooth and strongly convex.

This problem has significant applications in resource allocation and systems control and can also arise in distributed machine learning. We propose lower complexity bounds for decentralized optimization problems with coupled constraints and a first-order algorithm achieving the lower bounds. To the best of our knowledge, our method is also the first linearly convergent first-order decentralized algorithm for problems with general affine coupled constraints.

1 INTRODUCTION

We consider the decentralized optimization problem with coupled constraints

$$\min_{x_1 \in \mathbb{R}^{d_1}, \ldots, x_n \in \mathbb{R}^{d_n}} \sum_{i=1}^{n} f_i(x_i) \quad \text{s.t.} \quad \sum_{i=1}^{n} (\mathbf{A}_i x_i - b_i) = 0, \tag{1}$$

where for $i \in \{1, \ldots, n\}$ functions $f_i(x_i) \colon \mathbb{R}^{d_i} \to \mathbb{R}$ are continuously differentiable, $\mathbf{A}_i \in \mathbb{R}^{m \times d_i}$ and $b_i \in \mathbb{R}^m$ are constraint matrices and vectors respectively.

We are interested in solving problem (1) in a decentralized distributed setting. That is, we assume the existence of a communication network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{1, \ldots, n\}$ is the set of compute nodes, and $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ is the set of communication links in the network. Each compute node $i \in \mathcal{V}$ locally stores the objective function $f_i(x_i)$, the constraint matrix $\mathbf{A}_i$ and the vector b_i. Compute node $i \in \mathcal{V}$ can send information (e.g., vectors, scalars, etc.) to compute node $j \in \mathcal{V}$ if and only if there is an edge $(i, j) \in \mathcal{E}$ in the communication network.

Coupled constraints arise in various application scenarios, where sharing resources or information takes place. Often, due to the distributed nature of such problems, decentralization is desired for communication and/or privacy related reasons. Let us briefly describe several practical cases of optimization problems with coupled constraints.

- **Optimal exchange.** Also known as the resource allocation problem Boyd et al. (2011); Nedić et al. (2018), it writes as

$$\min_{x_1, \ldots, x_n \in \mathbb{R}^d} \sum_{i=1}^{n} f_i(x_i) \quad \text{s.t.} \quad \sum_{i=1}^{n} x_i = b,$$

where $x_i \in \mathbb{R}^d$ represents the quantities of commodities exchanged among the agents of the system, and $b \in \mathbb{R}^d$ represents the shared budget or demand for each commodity. This problem is essential in economics Arrow and Debreu (1954), and systems control Dominguez-Garcia et al. (2012).

- **Problems on graphs.** In various applications, distributed systems are formed on the basis of physical networks. This is the case for electrical microgrids, telecommunication networks and drone swarms. Distributed optimization on graphs applies to such systems and encompasses, to name a few, optimal power flow Wang et al. (2016) and power system state estimation Zhang et al. (2024) problems.

As an example, consider an electric power network. Let $x_i \in \mathbb{R}^2$ denote the voltage phase angle and the magnitude at i-th electric node, and let s be the vector of (active and reactive) power flows for each pair of adjacent electric nodes. Highly accurate linearization approaches Yang et al. (2016); Van den Bergh et al. (2014) allow to formulate the necessary relation between voltages and power flows as a linear system of equations $\sum_{i=1}^{n} \mathbf{A}_i x_i = s$. An important property of the matrices $\mathbf{A}_i$ is that their compatibility with the physical network (but not necessary with the communication network). This means that for each row of the matrix $(\mathbf{A}_1, \ldots, \mathbf{A}_n)$, there is a node k such that $\mathbf{A}_i$ can have nonzero elements in this row only if nodes i and k are connected in the physical network, or $k = i$.

• **Consensus optimization.** Related to the previous example is the consensus optimization Boyd et al. (2011)

$$\min_{x_1,\ldots,x_n \in \mathbb{R}^d} \sum_{i=1}^{n} f_i(x_i) \quad \text{s.t.} \quad x_1 = x_2 = \ldots = x_n.$$

It is widely used in horizontal federated learning Kairouz et al. (2021), as well as in the more general context of decentralized optimization of finite-sum objectives Gorbunov et al. (2022); Scaman et al. (2017).

To handle the consensus constraint, decentralized algorithms either reformulate it as $\sum_{i=1}^{n} \mathbf{W}_i x_i = 0$, where $\mathbf{W}_i$ is the i-th vertical block of a gossip matrix (an example of which is the communication graph's Laplacian), or utilize the closely related mixing matrix approach Gorbunov et al. (2022). Mixing and gossip matrices are used because they are communication-friendly: calculating the sum $\sum_{i=1}^{n} \mathbf{W}_i x_i$ only requires each compute node to communicate once with each of its adjacent nodes. Clearly, consensus optimization with gossip matrix reformulation can be reduced to (1) by setting $\mathbf{A}_i = \mathbf{W}_i$. However, the principial difference between this example and (1), is that (1) does not assume $\mathbf{A}_i$ to be communication-friendly.

• **Vertical federated learning (VFL).** In the case of VFL, the data is partitioned by features, differing from the usual (horizontal) federated learning, where the data is partitioned by samples Yang et al. (2019); Boyd et al. (2011). Let $\mathbf{F}$ be the matrix of features, split vertically between compute nodes into submatrices $\mathbf{F}_i$, so that each node possesses its own subset of features for all data samples. Let $l \in \mathbb{R}^m$ denote the vector of labels, and let $x_i \in \mathbb{R}^{d_i}$ be the vector of model parameters owned by the i-th node. VFL problem formulates as

$$\min_{\substack{z \in \mathbb{R}^m \\ x_1 \in \mathbb{R}^{d_1}, \ldots, x_n \in \mathbb{R}^{d_n}}} \ell(z, l) + \sum_{i=1}^{n} r_i(x_i) \quad \text{s.t.} \quad \sum_{i=1}^{n} \mathbf{F}_i x_i = z, \tag{2}$$

where ℓ is a loss function, and r_i are regularizers. The constraints in (2) are coupled constraints, and the objective is separable; therefore, it is a special case of (1). We return to the VFL example in Section 6.

Paper organization. In Section 2 we present a literature review. Subsequently, in Section 3 we introduce the assumptions and problem parameters. Section 4 describes the key ideas of algorithm development and Section 5 presents the convergence rate of the method and the lower complexity bounds. Finally, in Section 6, we provide numerical simulations.

2 RELATED WORK AND OUR CONTRIBUTION

Decentralized optimization algorithms were initially proposed for consensus optimization Nedić and Ozdaglar (2009), based on earlier research in distributed optimization Tsitsiklis (1984); Bertsekas and Tsitsiklis (1989) and algorithms for decentralized averaging (*consensus* or *gossip* algorithms) Boyd et al. (2006); Olshevsky and Tsitsiklis (2009), which assumed the existence of a communication network, as does the present paper. The optimal complexity for consensus optimization was first achieved with a dual accelerated gradient descent in Scaman et al. (2017), where the method required computing gradients of Fenchel conjugates of $f_i(x)$. The corresponding complexity lower bounds were also established in the same paper. This result was later generalized to primal algorithms (which use gradients of the functions $f_i(x)$ themselves) Kovalev et al. (2020), time-varying communication graphs Li and Lin (2021); Kovalev et al. (2021) and methods that use stochastic gradients Dvinskikh and Gasnikov (2021). Today there also exist algorithms with communication compression Beznosikov et al. (2023), asynchronous algorithms Koloskova (2024), algorithms for saddle-point formulations Rogozin et al. (2021) and gradient-free oracles Beznosikov et al. (2020), making decentralized consensus optimization a quite well-developed field Nedić (2020); Gorbunov et al. (2022), benefiting systems control Ram et al. (2009) and machine learning Lian et al. (2017).

Beginning with the addition of local constraints to consensus optimization Nedic et al. (2010); Zhu and Martinez (2011), constrained decentralized optimization has been established as a research direction. A zoo of distributed problems with constraints was investigated in Necoara et al. (2011); Necoara and Nedelcu (2014; 2015).

Primarily motivated by the demand from the power systems community, various decentralized algorithms for coupled constraints have been proposed. Generally designed for versatile engineering applications, many of these algorithms assume restricted function domains Wang and Hu (2022); Liang et al. (2019); Nedić et al. (2018); Gong and Zhang (2023); Zhang et al. (2021); Wu et al. (2022), nonlinear inequality constraints Liang et al. (2019); Gong and Zhang (2023); Wu et al. (2022), time-varying graphs Zhang et al. (2021); Nedić et al. (2018) or utilize specific problem structure Wang and Hu (2022).

Table 1: Comparison of algorithms for decentralized optimization with coupled constraints

Reference	Oracle	Rate
Doan and Olshevsky (2017) †	First-order	Linear
Falsone et al. (2020)	Prox	Sub-linear
Wu et al. (2022)	Prox	Sub-linear
Chang (2016)	Prox	Sub-linear
Li et al. (2018) †	Prox	Linear
Gong and Zhang (2023)	Inexact prox	Linear
Nedić et al. (2018) †	First-order	Accelerated
This work	First-order	Optimal

† Applicable only for resource allocation problem

Works of Doan and Olshevsky (2017); Li et al. (2018); Nedić et al. (2018) focus on the resource allocation problem. For undirected time-varying graphs Doan and Olshevsky (2017) proposes a first-order algorithm with $O(\frac{\kappa_f n^2}{B} \ln \frac{1}{\varepsilon})$ complexity bound, where B is the time required for the time-varying graph to reach connectivity. Li et al. (2018) applies a combination of gradient tracking and push-sum approaches from Nedic et al. (2017) to obtain linear convergence on directed time-varying graphs in the restricted domain case, *i.e.*, $x_i \in \Omega_i$, where Ω_i is a nonempty closed convex set. Nedić et al. (2018) achieve accelerated linear convergence via a proximal point method in the restricted domain case. When $\Omega_i = \mathbb{R}^d$, they also show that Nesterov's accelerated gradient descent can be applied to achieve optimal $O(\sqrt{\kappa_f} \ln \frac{1}{\varepsilon})$ complexity. In Gong and Zhang (2023) an inexact proximal-point method is proposed to solve problems with coupled affine equality and convex inequality constraints. Linear convergence is proved when the inequalities are absent, and Ω_i are convex polyhedrons. The papers Wu et al. (2022), Chang (2016), Falsone et al. (2020) present algorithms with sub-linear convergence.

As summarized in Table 1, no accelerated linearly convergent algorithms for general affine-equality coupled constraints were present in the literature prior to our work. Also, most of the algorithms require proximal oracle, which allows to handle more general problem formulations, but has higher computational burden than the first-order oracle. We propose a new first-order decentralized algorithm with optimal (accelerated) linear convergence rate. We prove its optimality by providing lower bounds for the number of objective's gradient computations, matrix multiplications and decentralized communications, which match complexity bounds for our algorithm.

3 Mathematical setting and assumptions

Let us begin by introducing the notation. The largest and smallest nonzero eigenvalues (or singular values) of a matrix $\mathbf{C}$ are denoted by $\lambda_{\max}(\mathbf{C})$ (or $\sigma_{\max}(\mathbf{C})$) and $\lambda_{\min^+}(\mathbf{C})$ (or $\sigma_{\min^+}(\mathbf{C})$), respectively. For vectors $x_i \in \mathbb{R}^{d_i}$ we introduce a column-stacked vector $x = \operatorname{col}(x_1, \ldots, x_m) = (x_1^\top \ldots x_m^\top)^\top \in \mathbb{R}^d$. We denote the identity matrix by $\mathbf{I}_m \in \mathbb{R}^{m \times m}$. The symbol $\otimes$ denotes the Kronecker product of matrices. By $\mathcal{L}_m$ we denote the so-called consensus space, which is given as $\mathcal{L}_m = \{(y_1, \ldots, y_n) \in (\mathbb{R}^m)^n : y_1, \ldots, y_n \in \mathbb{R}^m \text{ and } y_1 = \cdots = y_n\}$, and $\mathcal{L}_m^\perp$ denotes the orthogonal complement to $\mathcal{L}_m$, which is given as

$$\mathcal{L}_m^\perp = \{(y_1, \ldots, y_n) \in (\mathbb{R}^m)^n : y_1, \ldots, y_n \in \mathbb{R}^m \text{ and } y_1 + \cdots + y_n = 0\}. \tag{3}$$

Assumption 1. *Continuously differentiable functions $f_i(x) \colon \mathbb{R}^{d_i} \to \mathbb{R}$, $i \in \{1, \ldots, n\}$ are L_f-smooth and μ_f-strongly convex, where $L_f \geq \mu_f > 0$. That is, for all $x_1, x_2 \in \mathbb{R}^{d_i}$ and $i \in \{1, \ldots, n\}$, the following inequalities hold:*

$$\frac{\mu_f}{2}\|x_2 - x_1\|^2 \leq f_i(x_2) - f_i(x_1) - \langle \nabla f_i(x_1), x_2 - x_1 \rangle \leq \frac{L_f}{2}\|x_2 - x_1\|^2.$$

By κ_f we denote the condition number $\kappa_f = L_f/\mu_f$.

Assumption 2. *There exists $x^* = (x_1^*, \ldots, x_n^*)$, $x_i^* \in \mathbb{R}^{d_i}$ such that $\sum_{i=1}^n (\mathbf{A}_i x_i^* - b_i) = 0$. There exist constants $L_\mathbf{A} \geq \mu_\mathbf{A} > 0$, such that the constraint matrices $\mathbf{A}_1, \ldots, \mathbf{A}_n$ satisfy the following inequalities:*

$$\sigma^2_{\max}(\mathbf{A}) = \max_{i \in \{1, \ldots, n\}} \sigma^2_{\max}(\mathbf{A}_i) \leq L_\mathbf{A}, \qquad \mu_\mathbf{A} \leq \lambda_{\min^+}(\mathbf{S}), \tag{4}$$

where the matrix $\mathbf{S} \in \mathbb{R}^{m \times m}$ is defined as $\mathbf{S} = \frac{1}{n}\sum_{i=1}^n \mathbf{A}_i \mathbf{A}_i^\top$. We also define the condition number of the matrix $\mathbf{A}$ as $\kappa_\mathbf{A} = L_\mathbf{A}/\mu_\mathbf{A}$.

For any matrix $\mathbf{M}$ other than $\mathbf{A}$ we denote by $L_\mathbf{M}$ and $\mu_\mathbf{M}$ some upper and lower bound on its maximal and minimal positive squared singular values respectively:

$$\lambda_{\max}(\mathbf{M}^\top \mathbf{M}) = \sigma^2_{\max}(\mathbf{M}) \leq L_\mathbf{M}, \qquad \mu_\mathbf{M} \leq \sigma^2_{\min^+}(\mathbf{M}) = \lambda_{\min^+}(\mathbf{M}^\top \mathbf{M}). \tag{5}$$

We also assume the existence of a so-called gossip matrix $W \in \mathbb{R}^{n \times n}$ associated with the communication network $\mathcal{G}$, which satisfies the following assumption.

Assumption 3. *The gossip matrix W is a $n \times n$ symmetric positive semidefinite matrix such that:*

1. *$W_{ij} \neq 0$ if and only if $(i,j) \in \mathcal{E}$ or $i = j$.*
2. *$Wy = 0$ if and only if $y \in \mathcal{L}_1$, i.e. $y_1 = \ldots = y_n$.*
3. *There exist constants $L_\mathbf{W} \geq \mu_\mathbf{W} > 0$ such that $\mu_\mathbf{W} \leq \lambda^2_{\min^+}(W)$ and $\lambda^2_{\max}(W) \leq L_\mathbf{W}$.*

We will use a dimension-lifted analogue of the gossip matrix defined as $\mathbf{W} = W \otimes \mathbf{I}_m$. From the properties of the Kronecker product of matrices it follows that $\lambda^2_{\min^+}(\mathbf{W}) = \lambda^2_{\min^+}(W)$ and $\lambda^2_{\max}(\mathbf{W}) = \lambda^2_{\max}(W)$. By $\kappa_\mathbf{W}$ we denote the condition number

$$\kappa_\mathbf{W} = \sqrt{\frac{L_\mathbf{W}}{\mu_\mathbf{W}}} \geq \frac{\lambda_{\max}(\mathbf{W})}{\lambda_{\min^+}(\mathbf{W})}. \tag{6}$$

Moreover, the kernel and range spaces of W and $\mathbf{W}$ are given by

$$\ker W = \mathcal{L}_1, \ \operatorname{range} W = \mathcal{L}_1^\perp, \quad \ker \mathbf{W} = \mathcal{L}_m, \ \operatorname{range} \mathbf{W} = \mathcal{L}_m^\perp. \tag{7}$$

4 Derivation of the algorithm

4.1 Strongly convex communication-friendly reformulation

Let $\mathbf{W}'$ be any positive semidefinite matrix such that

$$\operatorname{range} \mathbf{W}' = (\ker \mathbf{W}')^\perp = \mathcal{L}_m^\perp, \tag{8}$$

and multiplication of a vector $y = (y_1, \ldots, y_n) \in (\mathbb{R}^m)^n$ by $\mathbf{W}'$ can be performed efficiently in the decentralized manner if its i-th block component y_i is stored at i-th node of the computation network. Similarly to eq. (6), we define

$$\kappa_{\mathbf{W}'} = \sqrt{\frac{L_{\mathbf{W}'}}{\mu_{\mathbf{W}'}}} \geq \frac{\lambda_{\max}(\mathbf{W}')}{\lambda_{\min^+}(\mathbf{W}')}. \tag{9}$$

Due to the definition of $\mathbf{W}$ and eq. (7), the simplest choice for $\mathbf{W}'$ might be to set $\mathbf{W}' = \mathbf{W}$. Later we will specify another way to choose $\mathbf{W}'$ for optimal algorithmic performance.

Problem (1) can be reformulated as follows:

$$\min_{x\in\mathbb{R}^d, y\in(\mathbb{R}^m)^n} G(x,y) \quad \text{s.t.} \quad \mathbf{A}x + \gamma\mathbf{W}'y = \mathbf{b}, \tag{10}$$

where the function $G(x,y)\colon \mathbb{R}^d \times (\mathbb{R}^m)^n \to \mathbb{R}$ is defined as

$$G(x,y) = F(x) + \frac{r}{2}\|\mathbf{A}x + \gamma\mathbf{W}'y - \mathbf{b}\|^2, \tag{11}$$

the function $F(x)\colon \mathbb{R}^d \to \mathbb{R}$ is defined as $F(x) = \sum_{i=1}^n f_i(x_i)$, where $x = (x_1,\ldots,x_n), x_i \in \mathbb{R}^{d_i}$, the matrix $\mathbf{A} \in \mathbb{R}^{mn\times d}$ is the block-diagonal matrix $\mathbf{A} = \operatorname{diag}(\mathbf{A}_1,\ldots,\mathbf{A}_n)$, the vector $\mathbf{b}$ is the column-stacked vector $\mathbf{b} = \operatorname{col}(b_1,\ldots,b_n) \in \mathbb{R}^{mn}$, and $r,\gamma > 0$ are scalar constants that will be determined later.

From the definitions of $\mathbf{A}$, $\mathbf{b}$ and $\mathcal{L}_m^\perp$ (eq. (3)) it is clear that $\sum_{i=1}^n(\mathbf{A}_i x_i - b_i) = 0$ if and only if $\mathbf{A}x - \mathbf{b} \in \mathcal{L}_m^\perp$. Since $\operatorname{range}\mathbf{W}' = \mathcal{L}_m^\perp$, the constraint in problem (10) is equivalent to the coupled constraint in (1). For all x, y satisfying the constraint, the augmented objective function $G(x,y)$ is equal to the original objective function $F(x)$. Therefore, problem (10) is equivalent to problem (1).

The following Lemma 1 shows that the function $G(x,y)$ is strongly convex and smooth.

Lemma 1. *Let r and γ be defined as follows:*

$$r = \frac{\mu_f}{2L_{\mathbf{A}}}, \quad \gamma^2 = \frac{\mu_{\mathbf{A}} + L_{\mathbf{A}}}{\mu_{\mathbf{W}'}}. \tag{12}$$

Then, the strong convexity and smoothness constants of $G(x,y)$ on $\mathbb{R}^d \times \mathcal{L}_m^\perp$ are given by

$$\mu_G = \mu_f \min\left\{\frac{1}{2}, \frac{\mu_{\mathbf{A}} + L_{\mathbf{A}}}{4L_{\mathbf{A}}}\right\}, \quad L_G = \max\left\{L_f + \mu_f, \mu_f\frac{\mu_{\mathbf{A}} + L_{\mathbf{A}}}{L_{\mathbf{A}}}\frac{L_{\mathbf{W}'}}{\mu_{\mathbf{W}'}}\right\}. \tag{13}$$

Let the matrix $\mathbf{B} \in \mathbb{R}^{mn\times(d+mn)}$ be defined as $\mathbf{B} = [\mathbf{A} \quad \gamma\mathbf{W}']$. The following Lemma 2 connects the spectral properties of $\mathbf{B}$, $\mathbf{A}$ and $\mathbf{W}'$.

Lemma 2. *The following bounds on the singular values of $\mathbf{B}$ hold:*

$$\sigma^2_{\min^+}(\mathbf{B}) \geq \mu_{\mathbf{B}} = \frac{\mu_{\mathbf{A}}}{2}, \quad \sigma^2_{\max}(\mathbf{B}) \leq L_{\mathbf{B}} = L_{\mathbf{A}} + (L_{\mathbf{A}} + \mu_{\mathbf{A}})\frac{L_{\mathbf{W}'}}{\mu_{\mathbf{W}'}}, \tag{14}$$

and

$$\frac{\sigma^2_{\max}(\mathbf{B})}{\sigma^2_{\min^+}(\mathbf{B})} \leq \frac{L_{\mathbf{B}}}{\mu_{\mathbf{B}}} = \kappa_{\mathbf{B}} = 2\left(\kappa_{\mathbf{A}} + \frac{L_{\mathbf{W}'}}{\mu_{\mathbf{W}'}}(1+\kappa_{\mathbf{A}})\right). \tag{15}$$

Proofs of Lemma 1 and Lemma 2 are provided in Appendix A.

4.2 Chebyshev acceleration

Chebyshev acceleration allows us to decouple the number of computations of the objective's gradient $\nabla F(x)$ from the properties of the communication network and the constraint matrix — specifically, from the condition numbers $\kappa_{\mathbf{W}}$ and $\kappa_{\mathbf{A}}$. The Chebyshev trick enables to replace the matrix with a matrix polynomial with a better condition number.

Consider some affine relation $\mathbf{M}u = \mathbf{d}$ and let $\mathcal{P}_{\mathbf{M}}$ be a polynomial such that $\mathcal{P}_{\mathbf{M}}(\lambda) = 0 \Leftrightarrow \lambda = 0$ for any eigenvalue λ of $\mathbf{M}^\top\mathbf{M}$. Note that here we interchangeably use $\mathcal{P}$ as a polynomial of a matrix and a polynomial of a scalar. We denote any feasible point for the constraint $\mathbf{M}u = \mathbf{d}$ as u_0. Then,

$$\begin{aligned}\mathbf{M}u = \mathbf{d} \Leftrightarrow \mathbf{M}(u - u_0) = 0 &\overset{(a)}{\Leftrightarrow} \mathbf{M}^\top\mathbf{M}(u - u_0) = 0 \\ &\overset{(b)}{\Leftrightarrow} \mathcal{P}_{\mathbf{M}}(\mathbf{M}^\top\mathbf{M})(u - u_0) = 0 \overset{(c)}{\Leftrightarrow} \sqrt{\mathcal{P}_{\mathbf{M}}(\mathbf{M}^\top\mathbf{M})}(u - u_0) = 0\end{aligned}$$

where (a) and (c) is due to $\ker\mathbf{M}^\top\mathbf{M} = \ker\mathbf{M}$; (b) is due to $\ker\mathcal{P}_{\mathbf{M}}(\mathbf{M}^\top\mathbf{M}) = \ker\mathbf{M}^\top\mathbf{M}$ by the assumption about $\mathcal{P}_{\mathbf{M}}(\lambda)$.

Following Salim et al. (2022a) and Scaman et al. (2017), we use the translated and scaled Chebyshev polynomials, because they are the best at compressing the spectrum Auzinger and Melenk (2011).

Lemma 3 (Salim et al. (2022a), Section 6.3.2)**.** *Consider a matrix* $\mathbf{M}$*. Let* $\ell = \left\lceil \sqrt{\frac{L_{\mathbf{M}}}{\mu_{\mathbf{M}}}} \right\rceil \geq \left\lceil \sqrt{\frac{\lambda_{\max}(\mathbf{M}^\top \mathbf{M})}{\lambda_{\min^+}(\mathbf{M}^\top \mathbf{M})}} \right\rceil$*. Define* $\mathcal{P}_{\mathbf{M}}(t) = 1 - \frac{T_\ell((L_{\mathbf{M}}+\mu_{\mathbf{M}}-2t)/(L_{\mathbf{M}}-\mu_{\mathbf{M}}))}{T_\ell((L_{\mathbf{M}}+\mu_{\mathbf{M}})/(L_{\mathbf{M}}-\mu_{\mathbf{M}}))}$*, where* T_ℓ *is the Chebyshev polynomial of the first kind of degree* n *defined by* $T_\ell(t) = \frac{1}{2}\left(\left(t+\sqrt{t^2-1}\right)^\ell + \left(t-\sqrt{t^2-1}\right)^\ell\right)$*. Then,* $\mathcal{P}_{\mathbf{M}}(0) = 0$*, and*

$$\lambda_{\max}\left(\mathcal{P}_{\mathbf{M}}(\mathbf{M}^\top \mathbf{M})\right) \leq \max_{t\in[\mu_{\mathbf{M}}, L_{\mathbf{M}}]} \mathcal{P}_{\mathbf{M}}(t) \leq \frac{19}{15}, \tag{16}$$

$$\lambda_{\min^+}\left(\mathcal{P}_{\mathbf{M}}(\mathbf{M}^\top \mathbf{M})\right) \geq \min_{t\in[\mu_{\mathbf{M}}, L_{\mathbf{M}}]} \mathcal{P}_{\mathbf{M}}(t) \geq \frac{11}{15}. \tag{17}$$

Results of this section are summarized in the following Lemma 4.

Lemma 4. *Define*

$$\mathbf{W}' = \mathcal{P}_{\sqrt{\mathbf{W}}}(\mathbf{W}) \tag{18}$$

and

$$\mathbf{K} = \sqrt{\mathcal{P}_{\mathbf{B}}(\mathbf{B}^\top \mathbf{B})}. \tag{19}$$

Let $G(u) = G(x, y)$*,* $\mathcal{U} = \mathbb{R}^d \times \mathcal{L}_m^\perp$ *and* $\mathbf{b}' = \sqrt{\mathcal{P}_{\mathbf{B}}(\mathbf{B}^\top \mathbf{B})}u_0$*. Then, problem*

$$\min_{u\in\mathcal{U}} G(u) \quad \text{s.t.} \quad \mathbf{K}u = \mathbf{b}' \tag{20}$$

is an equivalent preconditioned reformulation of problem (10)*, and, in turn, of problem* (1)*.*

4.3 Base algorithm

Our base algorithm, Algorithm 1, is the Proximal Alternating Predictor-Corrector (PAPC) with Nesterov's acceleration, called Accelerated PAPC (APAPC). It was proposed in Salim et al. (2022a) to obtain an optimal algorithm for optimization problems formulated as (20). See Kovalev et al. (2020); Salim et al. (2022b) for the review of related algorithms and history of their development.

Algorithm 1 APAPC

```
Parameters: $u^0 \in \mathcal{U}$ $\eta, \theta, \alpha > 0$, $\tau \in (0, 1)$
Set $u_f^0 = u^0$, $z^0 = 0 \in \mathcal{U}$
for $k = 0, 1, 2, \dots$ do
    $u_g^k := \tau u^k + (1-\tau)u_f^k$
    $u^{k+\frac{1}{2}} := (1+\eta\alpha)^{-1}(u^k - \eta(\nabla G(u_g^k) - \alpha u_g^k + z^k))$
    $z^{k+1} := z^k + \theta \mathbf{K}^\top(\mathbf{K}u^{k+\frac{1}{2}} - \mathbf{b}')$
    $u^{k+1} := (1+\eta\alpha)^{-1}(u^k - \eta(\nabla G(u_g^k) - \alpha u_g^k + z^{k+1}))$
    $u_f^{k+1} := u_g^k + \frac{2\tau}{2-\tau}(u^{k+1} - u^k)$
end for
```

APAPC algorithm formulates as Algorithm 1, and its convergence properties are given in Proposition 1.

Proposition 1 (Salim et al. (2022a), Proposition 1)**.** *Assume that the matrix* $\mathbf{K}$ *in* (20) *satisfies* $\mu_{\mathbf{K}} > 0$ *and* $\mathbf{b}' \in \operatorname{range}\mathbf{K}$*, and denote* $\kappa_{\mathbf{K}} = \frac{L_{\mathbf{K}}}{\mu_{\mathbf{K}}}$*. Also assume that the function* G *is* L_G*-smooth and* μ_G*-strongly convex. Set the parameter values of Algorithm 1 as* $\tau = \min\left\{1, \frac{1}{2}\sqrt{\frac{\kappa_{\mathbf{K}}}{\kappa_G}}\right\}$*,* $\eta = \frac{1}{4\tau L_G}$*,* $\theta = \frac{1}{\eta L_{\mathbf{K}}}$ *and* $\alpha = \mu_G$*. Denote by* u^* *the solution of problem* (20) *and by* z^* *the solution of its dual problem satisfying* $z^* \in \operatorname{range}\mathbf{K}$*. Then the iterates* u^k, z^k *of Algorithm 1 satisfy*

$$\begin{aligned} &\frac{1}{\eta}\left\|u^k - u^\star\right\|^2 + \frac{\eta\alpha}{\theta(1+\eta\alpha)}\left\|(\mathbf{K}^\top)^\dagger z^k - z^\star\right\|^2 \\ &+ \frac{2(1-\tau)}{\tau}\mathrm{D}_G(u_f^k, u^\star) \leq \left(1 + \frac{1}{4}\min\left\{\frac{1}{\sqrt{\kappa_G \kappa_{\mathbf{K}}}}, \frac{1}{\kappa_{\mathbf{K}}}\right\}\right)^{-k} C, \end{aligned} \tag{21}$$

where $C := \frac{1}{\eta}\left\|u^0 - u^\star\right\|^2 + \frac{1}{\theta}\|z^0 - z^\star\|^2 + \frac{2(1-\tau)}{\tau}\mathrm{D}_G(u_f^0, u^\star)$*, and* D_G *denotes the Bregman divergence of* G*, defined by* $D_G(u', u) = G(u') - G(u) - \langle \nabla G(u), u' - u\rangle$*.*

Algorithm 3 $\mathbf{mulW'}(y)$: Multiplication by $\mathbf{W}'$

```
Parameters: $y$
$\rho := \left(\sqrt{L_{\mathbf{W}}} - \sqrt{\mu_{\mathbf{W}}}\right)^2/16$
$\nu := \left(\sqrt{L_{\mathbf{W}}} + \sqrt{\mu_{\mathbf{W}}}\right)/2$
$\delta^0 := -\nu/2,\ n := \lceil\sqrt{\kappa_{\mathbf{W}}}\rceil$
$p^0 := -\mathbf{W}y/\nu,\ y^1 := y + p^0$
for $i = 1, \ldots, n-1$ do
    $\beta^{i-1} := \rho/\delta^{i-1}$
    $\delta^i := -(\nu + \beta^{i-1})$
    $p^i := \left(\mathbf{W}y^i + \beta^{i-1}p^{i-1}\right)/\delta^i$
    $y^{i+1} := y^i + p^i$
end for
Output: $y - y^n$
```

Algorithm 4 $\mathbf{grad\_G}(u)$: Computation of $\nabla G(u)$

```
Parameters: $u = (x, y)$
$z := r\left(\mathbf{A}x + \gamma \cdot \mathbf{mulW'}(y) - \mathbf{b}\right)$
Output: $\begin{pmatrix} \nabla F(x) + \mathbf{A}^\top z \\ \gamma \cdot \mathbf{mulW'}(z) \end{pmatrix}$
```

Algorithm 5 $\mathbf{K\_Chebyshev}(u)$: Computation of $\mathbf{K}^\top(\mathbf{K}u - \mathbf{b}')$

```
Parameters: $u = (x, y)$
$\rho := (L_{\mathbf{B}} - \mu_{\mathbf{B}})^2/16$
$\nu := (L_{\mathbf{B}} + \mu_{\mathbf{B}})/2$
$\delta^0 := -\nu/2,\ n := \lceil\sqrt{\kappa_{\mathbf{B}}}\rceil$
$q^0 := \mathbf{A}x + \gamma \cdot \mathbf{mulW'}(y) - \mathbf{b}$
$p^0 := -\frac{1}{\nu}\begin{pmatrix} \mathbf{A}^\top q^0 \\ \gamma \cdot \mathbf{mulW'}(q^0) \end{pmatrix}$
$u^1 := u + p^0$
for $i = 1, \ldots, n-1$ do
    $\beta^{i-1} := \rho/\delta^{i-1}$
    $\delta^i := -(\nu + \beta^{i-1})$
    $(x^i, y^i) = u^i$
    $q^i := \mathbf{A}x^i + \gamma \cdot \mathbf{mulW'}(y^i) - \mathbf{b}$
    $p^i := \frac{1}{\delta^i}\begin{pmatrix} \mathbf{A}^\top q^i \\ \gamma \cdot \mathbf{mulW'}(q^i) \end{pmatrix} + \beta^{i-1}p^{i-1}/\delta^i$
    $u^{i+1} := u^i + p^i$
end for
Output: $u - u^n$
```

5 MAIN RESULTS

5.1 ALGORITHM

As stated in Lemma 4, problem (20) is equivalent to problem (1). Due to Lemma 1, its objective is strongly convex, allowing us to apply Algorithm 1 to it. Using Lemma 3, we obtain that the condition numbers of $\mathbf{W}'$ and $\mathbf{K}$ are bounded as $O(1)$, but a single multiplication by $\mathbf{W}'$ and $\mathbf{K}, \mathbf{K}^\top$ translates to $O(\sqrt{\kappa_{\mathbf{W}}})$ multiplications by $\mathbf{W}$ and $O(\sqrt{\kappa_{\mathbf{B}}})$ multiplications by $\mathbf{B}, \mathbf{B}^\top$ respectively.

Algorithm 2 Main algorithm

```
Parameters: $x^0 \in \mathbb{R}^d$ $\eta, \theta, \alpha > 0$, $\tau \in (0, 1)$
Set $y^0 := 0 \in (\mathbb{R}^m)^n$, $u^0 := (x^0, y^0)$,
    $u_f^0 := u^0$, $z^0 := 0 \in \mathbb{R}^d \times (\mathbb{R}^m)^n$
for $k = 0, 1, 2, \ldots$ do
    $u_g^k := \tau u^k + (1-\tau)u_f^k$
    $g^k := \mathbf{grad\_G}(u_g^k) - \alpha u_g^k$
    $u^{k+\frac{1}{2}} := (1+\eta\alpha)^{-1}(u^k - \eta(g^k + z^k))$
    $z^{k+1} := z^k + \theta \cdot \mathbf{K\_Chebyshev}(u^{k+\frac{1}{2}})$
    $u^{k+1} := (1+\eta\alpha)^{-1}(u^k - \eta(g^k + z^{k+1}))$
    $u_f^{k+1} := u_g^k + \frac{2\tau}{2-\tau}(u^{k+1} - u^k)$
end for
```

We implement multiplications by $\mathbf{W}'$ and $\mathbf{K}, \mathbf{K}^\top$ through numerically stable Chebyshev iteration procedures given in Algorithms 3 and 5, which only use decentralized communications and multiplications by $\mathbf{A}, \mathbf{A}^\top$. Lemmas 1 to 3 allow us to express the complexity of Algorithm 1 in terms of the parameters of the initial problem given in Assumptions 1 to 3. All this leads us to the following Theorem 1, a detailed proof of which is provided in Appendix A.3, as well as the derivation of Algorithms 3 and 5 and values of the parameters of Algorithm 2.

Theorem 1. *Set the parameter values of Algorithm 2 as* $\tau = \min\left\{1, \frac{1}{2}\sqrt{\frac{19}{44\max\{1+\kappa_f, 6\}}}\right\}$, $\eta = \frac{1}{4\tau\max\{L_f+\mu_f, 6\mu_f\}}$, $\theta = \frac{15}{19\eta}$ *and* $\alpha = \frac{\mu_f}{4}$. *Denote by* x^* *the solution of problem* (1). *Then, for every* $\varepsilon > 0$, *Algorithm 2 finds* x^k *for which* $\|x^k - x^*\|^2 \le \varepsilon$ *using* $O(\sqrt{\kappa_f}\log(1/\varepsilon))$ *objective's gradient computations,* $O(\sqrt{\kappa_f}\sqrt{\kappa_{\mathbf{A}}}\log(1/\varepsilon))$ *multiplications by* $\mathbf{A}$ *and* $\mathbf{A}^\top$, *and* $O(\sqrt{\kappa_f}\sqrt{\kappa_{\mathbf{A}}}\sqrt{\kappa_{\mathbf{W}}}\log(1/\varepsilon))$ *communication rounds (multiplications by* $\mathbf{W}$*).*

5.2 Lower bounds

Let us formulate the lower complexity bounds for decentralized optimization with affine constraints. To do that, we formalize the class of the algorithms of interest. In the literature, approaches with continuous time Scaman et al. (2017) and discrete time Kovalev et al. (2021) are used. We use the latter discrete time formalization. We assume that the method works in synchronized rounds of three types: local objective's gradient computations, local matrix multiplications and communications. At each time step, algorithm chooses one of the three step types.

Since the devices may have different dimensions d_i of locally held vectors x_i, they cannot communicate these vectors directly. Instead, the nodes exchange quantities $\mathbf{A}_i x_i \in \mathbb{R}^m$. For this reason, we introduce two types of memory $\mathcal{M}_i(k)$ and $\mathcal{H}_i(k)$ for node i at step k. Set $\mathcal{M}_i(k)$ stands for the local memory that the node does not share and $\mathcal{H}_i(k)$ denotes the memory that the node exchanges with neighbors. The interaction between $\mathcal{M}_i(k)$ and $\mathcal{H}_i(k)$ is performed via multiplications by $\mathbf{A}_i$ and $\mathbf{A}_i^\top$.

Below we describe how the sets $\mathcal{M}_i(k), \mathcal{H}_i(k)$ are updated.

1. Algorithm performs local gradient comutation round at step k. Gradient updates only operate in $\mathcal{M}_i(k)$ and do not affect $\mathcal{H}_i(k)$. For all $i \in \mathcal{V}$ we have

$$\mathcal{M}_i(k+1) = \mathrm{Span}\left\{x, \nabla f_i(x), \nabla f_i^*(x) :\ x \in M_i(k)\right\},\ \mathcal{H}_i(k+1) = \mathcal{H}_i(k),$$

where f_i^* is the Fenchel conjugate of f_i.
2. Algorithm performs local matrix multiplication round at step k. Sets $\mathcal{H}_i(k)$ and $\mathcal{M}_i(k)$ make mutual updates via multiplication by $\mathbf{A}_i$ and $\mathbf{A}_i^\top$. For all $i \in \mathcal{V}$ we have

$$\mathcal{M}_i(k+1) = \mathrm{Span}\left\{\mathbf{A}_i^\top b_i,\ \mathbf{A}_i^\top y :\ y \in \mathcal{H}_i(k)\right\},\ \mathcal{H}_i(k+1) = \mathrm{Span}\left\{b_i, \mathbf{A}_i x :\ x \in \mathcal{M}_i(k)\right\}.$$

3. Algorithm performs a communication round at step k. The non-shared local memory $\mathcal{M}_i(k)$ stays unchanged, while the shared memory $\mathcal{H}_i(k+1)$ is updated via interaction with neighbors. For all $i \in \mathcal{V}$ we have

$$\mathcal{M}_i(k+1) = \mathcal{M}_i(k),\ \mathcal{H}_i(k+1) = \mathrm{Span}\left\{\mathcal{H}_j(k) :\ (i,j) \in \mathcal{E}\right\}.$$

Under given memory and computation model, we formulate the lower complexity bounds.

Theorem 2. *For any $L_f > \mu_f > 0$, $\kappa_{\mathbf{A}}, \kappa_{\mathbf{W}} > 0$ there exist L_f-smooth μ_f-strongly convex functions $\{f_i\}_{i=1}^n$, matrices $\mathbf{A}_i$ such that $\kappa_{\mathbf{A}} = L_{\mathbf{A}}/\mu_{\mathbf{A}}$ (where $L_{\mathbf{A}}, \mu_{\mathbf{A}}$ are defined in* (4)*), and a communication graph $\mathcal{G}$ with a corresponding gossip matrix $\mathbf{W}$ such that $\kappa_{\mathbf{W}} = \lambda_{\max}(\mathbf{W})/\lambda_{\min}^+(\mathbf{W})$, for which any first-order decentralized algorithm on problem* (1) *to reach accuracy ε requires at least*

$$N_f = \Omega\left(\sqrt{\kappa_f}\log\left(\frac{1}{\varepsilon}\right)\right) \textit{ gradient computations},$$

$$N_{\mathbf{A}} = \Omega\left(\sqrt{\kappa_f}\sqrt{\kappa_{\mathbf{A}}}\log\left(\frac{1}{\varepsilon}\right)\right) \textit{ multiplications by } \mathbf{A} \textit{ and } \mathbf{A}^\top,$$

$$N_{\mathbf{W}} = \Omega\left(\sqrt{\kappa_f}\sqrt{\kappa_{\mathbf{A}}}\sqrt{\kappa_{\mathbf{W}}}\log\left(\frac{1}{\varepsilon}\right)\right) \textit{ communication rounds (multiplications by } \mathbf{W}\textit{).}$$

A proof of Theorem 2 is provided in Appendix B.

6 Experiments

The experiments were run on CPU Intel(R) Core(TM) i9-7980XE, with 62.5 GB RAM.

• **Synthetic linear regression.** In this section we perform numerical experiments on a synthetic linear regression problem with ℓ_2-regularization:

$$\min_{x_1,\ldots,x_n \in \mathbb{R}^{d_i}} \sum_{i=1}^n \left(\frac{1}{2}\|C_i x_i - d_i\|_2^2 + \frac{\theta}{2}\|x_i\|_2^2\right) \quad \text{s.t.} \quad \sum_{i=1}^n (\mathbf{A}_i x_i - b_i) = 0, \tag{22}$$

where we randomly generate matrices $C_i \in \mathbb{R}^{d_i \times d_i}$, $\mathbf{A}_i \in \mathbb{R}^{m \times d_i}$ and vectors $d_i \in \mathbb{R}^{d_i}$, $b_i \in \mathbb{R}^m$ from the standard normal distribution. Local variables $x_i \in \mathbb{R}^{d_i}$ have the same dimension d_i, equal for all devices. Regularization parameter θ is 10^{-3}. In the Fig. 1 we demonstrate the performance of the our method on the problem, that has the following parameters: $\kappa_f = 3140$, $\kappa_{\mathbf{A}} = 27$, $\kappa_{\mathbf{W}} = 89$. There we use Erdős–Rényi graph topology with $n = 20$ nodes. Local variables dimension is $d_i = 3$ and number of linear constraints is $m = 10$. We compare performance of Algorithm 2 with Tracking-ADMM algorithm Falsone et al. (2020) and DPMM algorithm Gong and Zhang (2023). Note that Tracking-ADMM and DPMM are proximal algorithms that solve a subproblem at each iteration. The choice of objective function in our simulations (linear regression) makes the corresponding proximal operator effectively computable via Conjugate Gradient algorithm Nesterov (2004) that uses gradient computations. Therefore, we measure the computational complexity of these methods in the number of gradient computations, not the number of proximal operator computations.

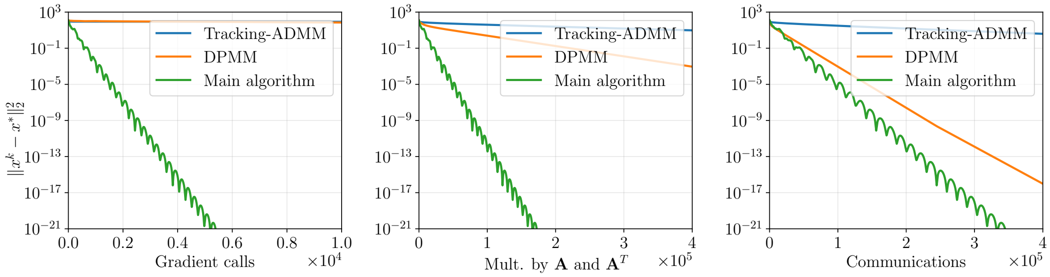

Figure 1: Synthetic, Erdős–Rényi graph, $n = 20$, $d_i = 3$, $m = 10$

• **VFL linear regression on real data.** Now we return to the problem, that we have announced in the introduction section. We apply VFL in the linear regression problem: ℓ is a typical mean squared loss function, that is $\ell(z, l) = \frac{1}{2}\|z - l\|_2^2$, and r_i are ℓ_2-regularizers, *i.e.* $r_i(x_i) = \lambda\|x_i\|_2^2$. To adapt this from (2) to (1), we redefine $x_1 := \binom{x_1}{z}$ and $x_2 := x_2, \ldots, x_n := x_n$. Thus, we can derive constraints matrices as in the (1):

$$\mathbf{A}_1 = \left(\mathbf{F}_1 \quad -\mathbf{I}\right), \qquad \mathbf{A}_1 x_1 = \mathbf{F}_1 w_1 - z, \tag{23}$$

$$\mathbf{A}_i = \mathbf{F}_i, \quad i = 2, \ldots, n, \qquad \sum_{i=1}^{n} \mathbf{A}_i x_i = \sum_{i=1}^{n} \mathbf{F}_i w_i - z. \tag{24}$$

For numerical simulation, we use `mushrooms` dataset from LibSVM library Chang and Lin (2011). We split $m = 100$ samples subset vertically between $n = 7$ devices. Regularization parameter $\lambda = 10^{-2}$. The results are in the Fig. 2.

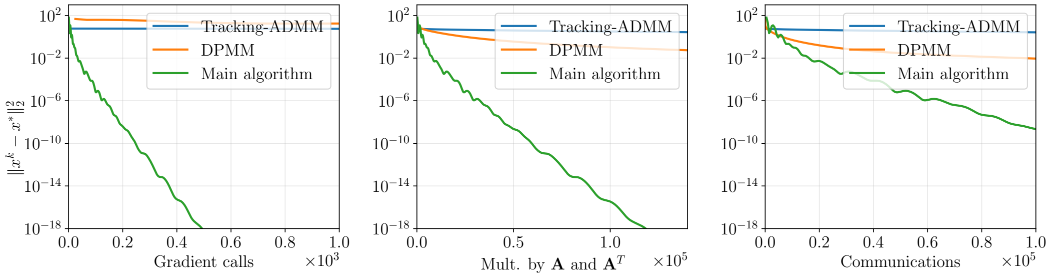

Figure 2: VFL, Erdős–Rényi graph, $n = 7$, $m = 100$

Our algorithm exhibits the best convergence rates, as evidenced by the steepest slopes. The slopes vary for gradient calls, matrix multiplications, and communications. This is due to the fact that Algorithm 2 involves many communications per iteration, in contrast to DPMM and Tracking-ADMM, which make numerous gradient calls per iteration.

REFERENCES

Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Found. Trends Mach. Learn.*, 3(1):1–122, January 2011. ISSN 1935-8237. doi:10.1561/2200000016. URL `http://dx.doi.org/10.1561/2200000016`.

Angelia Nedić, Alex Olshevsky, and Wei Shi. Improved convergence rates for distributed resource allocation. In *2018 IEEE Conference on Decision and Control (CDC)*, pages 172–177. IEEE, 2018.

Kenneth J Arrow and Gerard Debreu. Existence of an equilibrium for a competitive economy. *Econometrica: Journal of the Econometric Society*, pages 265–290, 1954.

Alejandro D Dominguez-Garcia, Stanton T Cady, and Christoforos N Hadjicostis. Decentralized optimal dispatch of distributed energy resources. In *2012 IEEE 51st IEEE conference on decision and control (CDC)*, pages 3688–3693. IEEE, 2012.

Yamin Wang, Lei Wu, and Shouxiang Wang. A fully-decentralized consensus-based admm approach for dc-opf with demand response. *IEEE Transactions on Smart Grid*, 8(6):2637–2647, 2016.

Haixiang Zhang, Ying Chen, and Javad Lavaei. Geometric analysis of matrix sensing over graphs. *Advances in Neural Information Processing Systems*, 36, 2024.

Jingwei Yang, Ning Zhang, Chongqing Kang, and Qing Xia. A state-independent linear power flow model with accurate estimation of voltage magnitude. *IEEE Transactions on Power Systems*, 32 (5):3607–3617, 2016.

Kenneth Van den Bergh, Erik Delarue, and William D'haeseleer. Dc power flow in unit commitment models. *no. May*, 2014.

Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. *Foundations and trends® in machine learning*, 14(1–2):1–210, 2021.

Eduard Gorbunov, Alexander Rogozin, Aleksandr Beznosikov, Darina Dvinskikh, and Alexander Gasnikov. Recent theoretical advances in decentralized distributed convex optimization. In *High-Dimensional Optimization and Probability*, pages 253–325. Springer, 2022.

Kevin Scaman, Francis Bach, Sébastien Bubeck, Yin Tat Lee, and Laurent Massoulié. Optimal algorithms for smooth and strongly convex distributed optimization in networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 3027–3036. JMLR. org, 2017.

Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10(2):1–19, 2019.

Angelia Nedić and Asuman Ozdaglar. Distributed subgradient methods for multi-agent optimization. *IEEE Transactions on Automatic Control*, 54(1):48–61, 2009.

John Nikolas Tsitsiklis. Problems in decentralized decision making and computation. Technical report, Massachusetts Inst of Tech Cambridge Lab for Information and Decision Systems, 1984.

Dimitri P Bertsekas and John N Tsitsiklis. *Parallel and distributed computation: numerical methods*, volume 23. Prentice hall Englewood Cliffs, NJ, 1989.

Stephen Boyd, Arpita Ghosh, Balaji Prabhakar, and Devavrat Shah. Randomized gossip algorithms. *IEEE transactions on information theory*, 52(6):2508–2530, 2006.

Alex Olshevsky and John N Tsitsiklis. Convergence speed in distributed consensus and averaging. *SIAM Journal on Control and Optimization*, 48(1):33–55, 2009.

Huan Li and Zhouchen Lin. Accelerated gradient tracking over time-varying graphs for decentralized optimization. *arXiv preprint arXiv:2104.02596*, 2021.

Dmitry Kovalev, Elnur Gasanov, Alexander Gasnikov, and Peter Richtarik. Lower bounds and optimal algorithms for smooth and strongly convex decentralized optimization over time-varying networks. *Advances in Neural Information Processing Systems*, 34, 2021.

Darina Dvinskikh and Alexander Gasnikov. Decentralized and parallel primal and dual accelerated methods for stochastic convex programming problems. *Journal of Inverse and Ill-posed Problems*, 29(3):385–405, 2021.

Aleksandr Beznosikov, Samuel Horváth, Peter Richtárik, and Mher Safaryan. On biased compression for distributed learning. *Journal of Machine Learning Research*, 24(276):1–50, 2023.

Anastasiia Koloskova. Optimization algorithms for decentralized, distributed and collaborative machine learning. Technical report, EPFL, 2024.

Alexander Rogozin, Alexander Beznosikov, Darina Dvinskikh, Dmitry Kovalev, Pavel Dvurechensky, and Alexander Gasnikov. Decentralized distributed optimization for saddle point problems. *arXiv preprint arXiv:2102.07758*, 2021.

Aleksandr Beznosikov, Eduard Gorbunov, and Alexander Gasnikov. Derivative-free method for composite optimization with applications to decentralized distributed optimization. *IFAC-PapersOnLine*, 53(2):4038–4043, 2020.

Angelia Nedić. Distributed gradient methods for convex machine learning problems in networks: Distributed optimization. *IEEE Signal Processing Magazine*, 37(3):92–101, 2020.

Sundhar Srinivasan Ram, Venugopal V Veeravalli, and Angelia Nedic. Distributed non-autonomous power control through distributed convex optimization. In *IEEE INFOCOM 2009*, pages 3001–3005. IEEE, 2009.

Xiangru Lian, Ce Zhang, Huan Zhang, Cho-Jui Hsieh, Wei Zhang, and Ji Liu. Can decentralized algorithms outperform centralized algorithms? a case study for decentralized parallel stochastic gradient descent. In *Advances in Neural Information Processing Systems*, pages 5330–5340, 2017.

Angelia Nedic, Asuman Ozdaglar, and Pablo A Parrilo. Constrained consensus and optimization in multi-agent networks. *IEEE Transactions on Automatic Control*, 55(4):922–938, 2010.

Minghui Zhu and Sonia Martinez. On distributed convex optimization under inequality and equality constraints. *IEEE Transactions on Automatic Control*, 57(1):151–164, 2011.

Ion Necoara, Valentin Nedelcu, and Ioan Dumitrache. Parallel and distributed optimization methods for estimation and control in networks. *Journal of Process Control*, 21(5):756–766, 2011.

Ion Necoara and Valentin Nedelcu. Distributed dual gradient methods and error bound conditions. *arXiv preprint arXiv:1401.4398*, 2014.

Ion Necoara and Valentin Nedelcu. On linear convergence of a distributed dual gradient algorithm for linearly constrained separable convex problems. *Automatica*, 55:209–216, 2015.

Jianzheng Wang and Guoqiang Hu. Distributed optimization with coupling constraints in multi-cluster networks based on dual proximal gradient method. *arXiv preprint arXiv:2203.00956*, 2022.

Shu Liang, George Yin, et al. Distributed smooth convex optimization with coupled constraints. *IEEE Transactions on Automatic Control*, 65(1):347–353, 2019.

Kai Gong and Liwei Zhang. Decentralized proximal method of multipliers for convex optimization with coupled constraints. *arXiv preprint arXiv:2310.15596*, 2023.

Bingru Zhang, Chuanye Gu, and Jueyou Li. Distributed convex optimization with coupling constraints over time-varying directed graphs. *Journal of Industrial and Management Optimization*, 17(4): 2119–2138, 2021.

Xuyang Wu, He Wang, and Jie Lu. Distributed optimization with coupling constraints. *IEEE Transactions on Automatic Control*, 68(3):1847–1854, 2022.

Thinh T Doan and Alex Olshevsky. Distributed resource allocation on dynamic networks in quadratic time. *Systems & Control Letters*, 99:57–63, 2017.

Alessandro Falsone, Ivano Notarnicola, Giuseppe Notarstefano, and Maria Prandini. Tracking-admm for distributed constraint-coupled optimization. *Automatica*, 117:108962, 2020.

Tsung-Hui Chang. A proximal dual consensus admm method for multi-agent constrained optimization. *IEEE Transactions on Signal Processing*, 64(14):3719–3734, 2016.

Huaqing Li, Qingguo Lü, Xiaofeng Liao, and Tingwen Huang. Accelerated convergence algorithm for distributed constrained optimization under time-varying general directed graphs. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 50(7):2612–2622, 2018.

Angelia Nedic, Alex Olshevsky, and Wei Shi. Achieving geometric convergence for distributed optimization over time-varying graphs. *SIAM Journal on Optimization*, 27(4):2597–2633, 2017.

Adil Salim, Laurent Condat, Dmitry Kovalev, and Peter Richtárik. An optimal algorithm for strongly convex minimization under affine constraints. In *International conference on artificial intelligence and statistics*, pages 4482–4498. PMLR, 2022a.

Winfried Auzinger and J Melenk. Iterative solution of large linear systems. *Lecture notes, TU Wien*, 2011.

Dmitry Kovalev, Adil Salim, and Peter Richtárik. Optimal and practical algorithms for smooth and strongly convex decentralized optimization. *Advances in Neural Information Processing Systems*, 33, 2020.

Adil Salim, Laurent Condat, Konstantin Mishchenko, and Peter Richtárik. Dualize, split, randomize: Toward fast nonsmooth optimization algorithms. *Journal of Optimization Theory and Applications*, 195(1):102–130, 2022b.

Yurii Nesterov. *Introductory Lectures on Convex Optimization: a basic course*. Kluwer Academic Publishers, Massachusetts, 2004.

Chih-Chung Chang and Chih-Jen Lin. Libsvm: A library for support vector machines. *ACM Trans. Intell. Syst. Technol.*, 2(3), may 2011. ISSN 2157-6904. doi:10.1145/1961189.1961199. URL `https://doi.org/10.1145/1961189.1961199`.

Martin H Gutknecht and Stefan Röllin. The chebyshev iteration revisited. *Parallel Computing*, 28(2): 263–283, 2002.

APPENDIX / SUPPLEMENTAL MATERIAL

A MISSING PROOFS FROM SECTION 4

A.1 PROOF OF LEMMA 1

Proof. Let $D_G(x',y';x,y)$ denote the Bregman divergence of G:

$$D_G(x',y';x,y) = G(x',y') - G(x,y) - \langle \nabla_x G(x,y), x'-x\rangle - \langle \nabla_y G(x,y), y'-y\rangle. \tag{25}$$

The value of μ_G can be obtained as follows:

$$\begin{aligned}
\mathrm{D}_G(x',y';x,y) &= \mathrm{D}_F(x';x) + \frac{r}{2}\|\mathbf{A}(x'-x) + \gamma\mathbf{W}'(y'-y)\|^2 \\
&\overset{(a)}{\geq} \frac{\mu_f}{2}\|x'-x\|^2 + \frac{r}{2}\|\mathbf{A}(x'-x) + \gamma\mathbf{W}'(y'-y)\|^2 \\
&= \frac{\mu_f}{2}\|x'-x\|^2 + \frac{r}{2}\|\mathbf{A}(x'-x)\|^2 + r\langle \mathbf{A}(x'-x), \gamma\mathbf{W}'(y'-y)\rangle \\
&\quad + \frac{r}{2}\|\gamma\mathbf{W}'(y'-y)\|^2 \\
&\overset{(b)}{\geq} \frac{\mu_f}{2}\|x'-x\|^2 + \frac{r}{4}\|\gamma\mathbf{W}'(y'-y)\|^2 - \frac{r}{2}\|\mathbf{A}(x'-x)\|^2 \\
&\overset{(c)}{\geq} \frac{\mu_f}{2}\|x'-x\|^2 + \frac{r\gamma^2\mu_{\mathbf{W}'}}{4}\|y'-y\|^2 - \frac{rL_{\mathbf{A}}}{2}\|x'-x\|^2 \\
&\overset{(d)}{=} \frac{\mu_f}{4}\|x'-x\|^2 + \frac{\mu_f\gamma^2\mu_{\mathbf{W}'}}{8L_{\mathbf{A}}}\|y'-y\|^2, \\
&\overset{(e)}{\geq} \frac{\mu_f}{2}\min\left\{\frac{1}{2}, \frac{\mu_{\mathbf{A}} + L_{\mathbf{A}}}{4L_{\mathbf{A}}}\right\}\left\|\begin{pmatrix} x'-x \\ y'-y \end{pmatrix}\right\|^2,
\end{aligned}$$

where (a) is due to Assumption 1; (b) is due to Young's inequality; (c) is due to Assumption 2, $y'-y \in \mathcal{L}_m^\perp$, eq. (8) and eq. (5); (d) and (e) is due to eq. (12).

The value of L_G can be obtained as follows:

$$\begin{aligned}
\mathrm{D}_G(x',y';x,y) &= \mathrm{D}_F(x';x) + \frac{r}{2}\|\mathbf{A}(x'-x) + \gamma\mathbf{W}'(y'-y)\|^2 \\
&\overset{(a)}{\leq} \frac{L_f}{2}\|x'-x\|^2 + \frac{r}{2}\|\mathbf{A}(x'-x) + \gamma\mathbf{W}'(y'-y)\|^2 \\
&\overset{(b)}{\leq} \frac{L_f}{2}\|x'-x\|^2 + r\|\gamma\mathbf{W}'(y'-y)\|^2 + r\|\mathbf{A}(x'-x)\|^2 \\
&\overset{(c)}{\leq} \frac{L_f}{2}\|x'-x\|^2 + r\gamma^2 L_{\mathbf{W}'}\|y'-y\|^2 + rL_{\mathbf{A}}\|x'-x\|^2 \\
&\overset{(d)}{=} \frac{L_f + \mu_f}{2}\|x'-x\|^2 + \frac{\mu_f\gamma^2 L_{\mathbf{W}'}}{2L_{\mathbf{A}}}\|y'-y\|^2, \\
&\overset{(e)}{\leq} \frac{1}{2}\max\left\{L_f + \mu_f, \mu_f\frac{\mu_{\mathbf{A}} + L_{\mathbf{A}}}{L_{\mathbf{A}}}\frac{L_{\mathbf{W}'}}{\mu_{\mathbf{W}'}}\right\}\left\|\begin{pmatrix} x'-x \\ y'-y \end{pmatrix}\right\|^2,
\end{aligned}$$

where (a) is due to Assumption 1; (b) is due to Young's inequality; (c) is due to Assumption 2 and eq. (5); (d) and (e) is due to eq. (12). □

A.2 PROOF OF LEMMA 2

Proof. To obtain the formula for $L_{\mathbf{B}}$, consider an arbitrary $z \in (\mathbb{R}^m)^n$:

$$\begin{aligned}
\|\mathbf{B}^\top z\|^2 &= \|\mathbf{A}^\top z\|^2 + \|\gamma\mathbf{W}'z\|^2 \\
&\overset{(a)}{\leq} (L_{\mathbf{A}} + \gamma^2 L_{\mathbf{W}'})\|z\|^2 \\
&\overset{(b)}{=} \left(L_{\mathbf{A}} + (L_{\mathbf{A}} + \mu_{\mathbf{A}})\frac{L_{\mathbf{W}'}}{\mu_{\mathbf{W}'}}\right)\|z\|^2,
\end{aligned}$$

where (a) is due to Assumption 2 and eq. (5); (b) is due to eq. (12).

To derive the formula for $\mu_{\mathbf{B}}$, first of all, note that by eq. (8)

$$(\ker \mathbf{B}^\top)^\perp = \operatorname{range} \mathbf{B} = \operatorname{range} \mathbf{A} + \operatorname{range} \mathbf{W}' = \operatorname{range} \mathbf{A} + \mathcal{L}_m^\perp. \tag{26}$$

Let $z \in (\ker \mathbf{B}^\top)^\perp = u + v$, where $u = (u_1, \ldots, u_n), v = (v_0, \ldots, v_0) \in (\mathbb{R}^m)^n$ such that $u \in \mathcal{L}_m^\perp$ and $v \in \mathcal{L}_m$.

We can show that $v_0 \in \operatorname{range} \mathbf{S}$. In order to do that, let us show that $\langle v_0, w_0 \rangle = 0$ for all $w_0 \in \ker \mathbf{S}$. Let $w = (w_0, \ldots, w_0) \in \mathcal{L}_m$. The fact that $w_0 \in \ker \mathbf{S}$ and $w \in \mathcal{L}_m$ implies $w \in \ker \mathbf{A}\mathbf{A}^\top = \ker \mathbf{A}^\top$. Hence, it is easy to show that $w \in \ker \mathbf{B}^\top = (\operatorname{range} \mathbf{B})^\perp$. Then, we obtain

$$n\langle v_0, w_0 \rangle \overset{(a)}{=} \langle v, w \rangle \overset{(b)}{=} \langle u + v, w \rangle = \langle z, w \rangle \overset{(c)}{=} 0,$$

where (a) follows from the definition of v and w; (b) follows from the fact that $u \in \mathcal{L}_m^\perp$ and $w \in \mathcal{L}_m$; (c) follows from the fact that $z \in \operatorname{range} \mathbf{B}$ and $w \in (\operatorname{range} \mathbf{B})^\perp$. Hence, $v_0 \in \operatorname{range} \mathbf{S}$.

Further, we get

$$\begin{aligned}
\|\mathbf{B}^\top z\|^2 &\overset{(a)}{=} \|\mathbf{A}^\top (u+v)\|^2 + \|\gamma \mathbf{W}'(u+v)\|^2 \\
&\overset{(b)}{=} \|\mathbf{A}^\top (u+v)\|^2 + \|\gamma \mathbf{W}' u\|^2 \\
&\overset{(c)}{\geq} \|\mathbf{A}^\top (u+v)\|^2 + \gamma^2 \mu_{\mathbf{W}'} \|u\|^2 \\
&= \|\mathbf{A}^\top u\|^2 + \|\mathbf{A}^\top v\|^2 + 2\langle \mathbf{A}^\top u, \mathbf{A}^\top v \rangle + \gamma^2 \mu_{\mathbf{W}'} \|u\|^2 \\
&\overset{(d)}{\geq} -\|\mathbf{A}^\top u\|^2 + \frac{1}{2}\|\mathbf{A}^\top v\|^2 + \gamma^2 \mu_{\mathbf{W}'} \|u\|^2 \\
&\overset{(e)}{=} -\|\mathbf{A}^\top u\|^2 + \frac{1}{2}\langle v_0, n\mathbf{S} v_0 \rangle + \gamma^2 \mu_{\mathbf{W}'} \|u\|^2 \\
&\overset{(f)}{\geq} -L_{\mathbf{A}}\|u\|^2 + \frac{n\mu_{\mathbf{A}}}{2}\|v_0\|^2 + \gamma^2 \mu_{\mathbf{W}'} \|u\|^2 \\
&= -L_{\mathbf{A}}\|u\|^2 + \frac{\mu_{\mathbf{A}}}{2}\|v\|^2 + \gamma^2 \mu_{\mathbf{W}'} \|u\|^2 \\
&\overset{(g)}{=} \frac{\mu_{\mathbf{A}}}{2}\|v\|^2 + \mu_{\mathbf{A}}\|u\|^2 \\
&\overset{(h)}{\geq} \frac{\mu_{\mathbf{A}}}{2}\|z\|^2,
\end{aligned}$$

where (a) and (h) is due to the definitions of u and v; (b) is due to the fact that $v \in \mathcal{L}_m$; (c) is due to eq. (5) and eq. (8); (d) uses Young's inequality; (e) is due to the definitions of v and $\mathbf{S}$, and $\|\mathbf{A}^\top v\|^2 = \left\| \begin{pmatrix} \mathbf{A}_1^\top v_0 \\ \vdots \\ \mathbf{A}_n^\top v_0 \end{pmatrix} \right\|^2 = \sum_{i=1}^n \|\mathbf{A}_i^\top v_0\|^2 = \langle v_0, \sum_{i=1}^n \mathbf{A}_i \mathbf{A}_i^\top v_0 \rangle = \langle v_0, n\mathbf{S} v_0 \rangle$; (f) is due to Assumption 2 and the definition of v; (g) is due to eq. (12).

□

A.3 Proof of Theorem 1

Lemma 5 (Salim et al. (2022a), Section 6.3.2)**.** *Let* $\mathbf{M}$ *be a matrix with* $\mu_{\mathbf{M}} > 0$*,* $\mathbf{r} \in \operatorname{range} \mathbf{M}$ *and* $\mathbf{M} v_0 = \mathbf{r}$*. Then* $\mathcal{P}_{\mathbf{M}}(\mathbf{M}^\top \mathbf{M})(v - v_0) = v - \mathbf{Chebyshev}(v, \mathbf{M}, \mathbf{r})$*, where* $\mathbf{Chebyshev}$ *is defined as Algorithm 6.*

Algorithm 6 $\mathbf{Chebyshev}(v, \mathbf{M}, \mathbf{r})$: Chebyshev iteration (Gutknecht and Röllin (2002), Algorithm 4)

```
Parameters: $v, \mathbf{M}, \mathbf{r}$.
$n \coloneqq \left\lceil \sqrt{\frac{L_{\mathbf{M}}}{\mu_{\mathbf{M}}}} \right\rceil$
$\rho \coloneqq \left(L_{\mathbf{M}} - \mu_{\mathbf{M}}\right)^2/16$, $\nu \coloneqq (L_{\mathbf{M}} + \mu_{\mathbf{M}})/2$
$\delta^0 \coloneqq -\nu/2$
$p^0 \coloneqq -\mathbf{M}^\top(\mathbf{M}v - \mathbf{r})/\nu$
$v^1 \coloneqq v + p^0$
for $i = 1, \ldots, n-1$ do
    $\beta^{i-1} \coloneqq \rho/\delta^{i-1}$
    $\delta^i \coloneqq -(\nu + \beta^{i-1})$
    $p^i \coloneqq \left(\mathbf{M}^\top(\mathbf{M}v^i - \mathbf{r}) + \beta^{i-1}p^{i-1}\right)/\delta^i$
    $v^{i+1} \coloneqq v^i + p^i$
end for
Output: $v^n$
```

Proof of Theorem 1

Proof. Applying Lemma 3 to $\mathbf{W}$ and $\mathbf{B}^\top\mathbf{B}$, we derive that, due to eq. (18), it holds

$$\lambda^2_{\max}(\mathbf{W}') \le L_{\mathbf{W}'} = (19/15)^2, \quad \lambda^2_{\min^+}(\mathbf{W}') \ge \mu_{\mathbf{W}'} = (11/15)^2, \tag{27}$$

and by eq. (6) the polynomial $\mathcal{P}_{\mathbf{W}}$ has a degree of $\lceil\sqrt{\kappa_{\mathbf{W}}}\rceil$. Similarly, due to eq. (19), it holds

$$\sigma^2_{\max}(\mathbf{K}) = \lambda_{\max}(\mathbf{K}^\top\mathbf{K}) \le L_{\mathbf{K}} = 19/15, \quad \sigma^2_{\min^+}(\mathbf{K}) = \lambda_{\min^+}(\mathbf{K}^\top\mathbf{K}) \ge \mu_{\mathbf{K}} = 11/15, \tag{28}$$

and since $\kappa_{\mathbf{B}} = \frac{L_{\mathbf{B}}}{\mu_{\mathbf{B}}}$, the polynomial $\mathcal{P}_{\mathbf{B}}$ has a degree of $\lceil\sqrt{\kappa_{\mathbf{B}}}\rceil$.

We implement computation of the term $\mathbf{K}^\top(\mathbf{K}u - \mathbf{b}')$ in line 6 of Algorithm 1 via Algorithm 5 by Lemma 5:

$$\begin{aligned}\mathbf{K}^\top(\mathbf{K}u - \mathbf{b}') &= \mathbf{K}^\top\mathbf{K}(u - u_0) = \mathcal{P}_{\mathbf{B}}(\mathbf{B}^\top\mathbf{B})(u - u_0)\\ &= u - \mathbf{Chebyshev}(u, \mathbf{B}, \mathbf{b}) = \mathbf{K\_Chebyshev}(u).\end{aligned}$$

Similarly, utilizing Lemma 5, we get

$$\mathbf{W}'y = \mathcal{P}_{\sqrt{\mathbf{W}}}(\mathbf{W})y = \mathcal{P}_{\sqrt{\mathbf{W}}}(\sqrt{\mathbf{W}}^\top\sqrt{\mathbf{W}})(y - 0) = y - \mathbf{Chebyshev}(y, \sqrt{\mathbf{W}}, 0) = \mathbf{mulW}'(y), \tag{29}$$

where $\mathbf{mulW}'$ is defined as Algorithm 3.

Therefore, Algorithm 2 is equivalent to Algorithm 1.

From eqs. (13) and (27), $\frac{\mu_{\mathbf{A}} + L_{\mathbf{A}}}{L_{\mathbf{A}}} \le 2$ and $(19/11)^2 \le 3$, we get

$$L_G = \max\left\{L_f + \mu_f, \mu_f \frac{\mu_{\mathbf{A}} + L_{\mathbf{A}}}{L_{\mathbf{A}}} \frac{L_{\mathbf{W}'}}{\mu_{\mathbf{W}'}}\right\} \le \mu_f \max\left\{1 + \kappa_f, 6\right\}, \tag{30}$$

$$\mu_G = \mu_f \min\left\{\frac{1}{2}, \frac{\mu_{\mathbf{A}} + L_{\mathbf{A}}}{4L_{\mathbf{A}}}\right\} \ge \frac{\mu_f}{4}, \tag{31}$$

$$\kappa_G = \frac{L_G}{\mu_G} \le 4\max\left\{1 + \kappa_f, 6\right\}. \tag{32}$$

From eqs. (15) and (27) we get

$$\kappa_{\mathbf{B}} = \frac{L_{\mathbf{B}}}{\mu_{\mathbf{B}}} \le 2\left(\kappa_{\mathbf{A}} + (19/11)^2(1 + \kappa_{\mathbf{A}})\right) \le 8\kappa_{\mathbf{A}} + 6. \tag{33}$$

From eq. (28) we obtain

$$\kappa_{\mathbf{K}} = \frac{L_{\mathbf{K}}}{\mu_{\mathbf{K}}} = 19/11, \tag{34}$$

and substituting eqs. (32) and (33) to Proposition 1, we obtain as its direct corollary that $k = O(\sqrt{\kappa_f}\log(1/\varepsilon))$. Each iteration of Algorithm 2 require $O(1)$ computations of ∇F, $O(\sqrt{\kappa_{\mathbf{B}}}) = O(\sqrt{\kappa_{\mathbf{A}}})$ multiplications by $\mathbf{A}, \mathbf{A}^\top$ and $O(\sqrt{\kappa_{\mathbf{A}}}\sqrt{\kappa_{\mathbf{W}}})$ multiplications by $\mathbf{W}$, which gives us the statement of Theorem 1. The values of the parameters $\tau, \eta, \theta, \alpha$ in Theorem 1 are derived from Proposition 1 as follows. We have $\tau = \min\left\{1, \frac{1}{2}\sqrt{\frac{\kappa_{\mathbf{K}}}{\kappa_G}}\right\} = \min\left\{1, \frac{1}{2}\sqrt{\frac{19}{44\max\{1+\kappa_f, 6\}}}\right\}$ due to eqs. (32) and (34); $\eta = \frac{1}{4\tau L_G} = \frac{1}{4\tau\max\{L_f+\mu_f, 6\mu_f\}}$ due to eq. (30); $\theta = \frac{15}{19\eta}$ due to eq. (28) and $\alpha = \mu_G = \frac{\mu_f}{4}$ due to eq. (31). □

B Proof of Theorem 2

B.1 Dual problem

Let us construct the lower bound for the problem dual to the initial one. Consider primal problem with zero r.h.s. in constraints.

$$\min_{x_1,\ldots,x_n\in\mathbb{R}^d} \sum_{i=1}^n f_i(x_i)$$
$$\text{s.t.} \sum_{i=1}^n \mathbf{A}_i x_i = 0$$

The dual problem has the form

$$\min_{x_1,\ldots,x_n\in\mathbb{R}^d} \max_w \left[\sum_{i=1}^n f_i(x_i) - \langle z, \mathbf{A}_i x_i\rangle\right] = \max_w \left[-\max_{x_1,\ldots,x_n\in\mathbb{R}^d} \sum_{i=1}^n \left\langle \mathbf{A}_i^\top z, x_i\right\rangle - f_i(x_i)\right]$$
$$= -\min_w \sum_{i=1}^n f_i^*(\mathbf{A}_i^\top z).$$

Introducing local copies of w at each node, we get

$$\min_{z_1,\ldots,z_n} \sum_{i=1}^n g_i(z_i) := \sum_{i=1}^n f_i^*(\mathbf{A}_i^\top z_i) \tag{35}$$
$$\text{s.t. } \mathbf{W}z = 0 \tag{36}$$

B.2 Example graph

We follow the principle of lower bounds construction introduced in Kovalev et al. (2021) and take the example graph from Scaman et al. (2017). Let the functions be held by the nodes be organized into a path graph with n vertices, where n is divisible by 3. The nodes of graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ are divided into three groups $\mathcal{V}_1 = \{1, \ldots, n/3\}, \mathcal{V}_2 = \{n/3+1, \ldots, 2n/3\}, \mathcal{V}_3 = \{2n/3+1, \ldots, n\}$ of $n/3$ vertices each.

Now we recall the construction from Scaman et al. (2017). Let $\gamma_n = \frac{1-\cos\left(\frac{\pi}{3n}\right)}{1+\cos\left(\frac{\pi}{3n}\right)}$. Since $\gamma_n \overset{n\to\infty}{\to} 0$, there exists $n \geq 1$ such that $\gamma_n \geq \frac{1}{\chi} > \gamma_{n+1}$. Introduce edge weights $w_{i,i+1} = 1 - a\mathbb{I}\{i=1\}$, take the corresponding weighed Laplacian W_a and denote its condition number χ. If $a = 1$, the network is disconnected and therefore $\chi(W_a) = \infty$. If $a = 0$, we have $\chi(W_a) = 1/\gamma_n$. By continuity of Laplacian spectra we obtain that for some $a \in [0, 1)$ it holds $\chi(W_a) = \chi$. Note that

$$\gamma_n = \frac{1-\cos\frac{\pi}{3n}}{1+\cos\frac{\pi}{3n}} \leq \frac{\pi^2}{9n^2} \quad\Rightarrow\quad \chi \geq \frac{9n^2}{\pi^2} \geq n^2. \tag{37}$$

B.3 Example functions

We let $e_1 = (1\ 0\ \ldots\ 0)^\top$ denote the first coordinate vector and define functions

$$f_i(p, t) = \frac{\mu_f}{2}\left\|p - \frac{\sqrt{\hat{L}_{\mathbf{A}}}}{2\mu_f}e_1\right\|^2 + \frac{L_f}{2}\|t\|^2.$$

Correspondingly,

$$f_i^*(u,v) = \frac{1}{2\mu_f}\|u\|^2 + \frac{1}{2L_f}\|v\|^2 - \frac{\sqrt{\hat{L}_{\mathbf{A}}}}{\mu_f}u_1.$$

To define matrices $\mathbf{A}_i$, we first introduce

$$\mathbf{E}_1 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & \dots \\ 0 & 1 & -1 & 0 & 0 & \dots \\ 0 & 0 & 0 & 0 & 0 & \dots \\ 0 & 0 & 0 & 1 & -1 & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix}, \quad \mathbf{E}_2 = \begin{pmatrix} 1 & -1 & 0 & 0 & 0 & \dots \\ 0 & 0 & 0 & 0 & 0 & \dots \\ 0 & 0 & 1 & -1 & 0 & \dots \\ 0 & 0 & 0 & 0 & 0 & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix}.$$

Let $\hat{L}_{\mathbf{A}} = \frac{1}{2}L_{\mathbf{A}} - \frac{3}{4}\mu_{\mathbf{A}},\ \hat{\mu}_{\mathbf{A}} = \frac{3}{2}\mu_{\mathbf{A}}$ and introduce

$$\mathbf{A}_i = \begin{cases} [\sqrt{\hat{L}_{\mathbf{A}}}\mathbf{E}_1^\top \ \ \sqrt{\hat{\mu}_{\mathbf{A}}}\mathbf{I}], & i \in \mathcal{V}_1 \\ [\ \mathbf{0} \qquad\quad \mathbf{0}\ \ ], & i \in \mathcal{V}_2 \\ [\sqrt{\hat{L}_{\mathbf{A}}}\mathbf{E}_2^\top \ \ \sqrt{\hat{\mu}_{\mathbf{A}}}\mathbf{I}], & i \in \mathcal{V}_3 \end{cases}$$

Let us make sure that the choice of $\mathbf{A}_i$ guarantees constants $L_{\mathbf{A}}, \mu_{\mathbf{A}}$ from (4).

$$\max_i \lambda_{\max}(\mathbf{A}_i\mathbf{A}_i^\top) = \lambda_{\max}\left(\hat{L}_{\mathbf{A}}\mathbf{E}_1^\top\mathbf{E}_1 + \hat{\mu}_{\mathbf{A}}\mathbf{I}\right) = 2\hat{L}_{\mathbf{A}} + \hat{\mu}_{\mathbf{A}} = L_{\mathbf{A}},$$

$$\begin{aligned} \lambda_{\min}^+\left(\frac{1}{n}\sum_{i=1}^n \mathbf{A}_i\mathbf{A}_i^\top\right) &= \lambda_{\min}^+\left(\frac{1}{3}(\hat{L}_{\mathbf{A}}\mathbf{E}_1^\top\mathbf{E}_1 + \hat{\mu}_{\mathbf{A}}\mathbf{I}) + \frac{1}{3}(\hat{L}_{\mathbf{A}}\mathbf{E}_2^\top\mathbf{E}_2 + \hat{\mu}_{\mathbf{A}}\mathbf{I})\right) \\ &= \frac{2}{3}\hat{\mu}_{\mathbf{A}} = \mu_{\mathbf{A}}. \end{aligned}$$

Let $\widetilde{\mathbf{M}} = \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}$ and

$$\mathbf{M}_1 = \mathbf{E}_1^\top\mathbf{E}_1 = \begin{pmatrix} 1 & 0 & 0 & \dots \\ 0 & \widetilde{\mathbf{M}} & 0 & \dots \\ 0 & 0 & \widetilde{\mathbf{M}} & \dots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix}, \quad \mathbf{M}_2 = \mathbf{E}_2^\top\mathbf{E}_2 = \begin{pmatrix} \widetilde{\mathbf{M}} & 0 & 0 & \dots \\ 0 & \widetilde{\mathbf{M}} & 0 & \dots \\ 0 & 0 & \widetilde{\mathbf{M}} & \dots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix}$$

The dual functions take the form

$$\begin{aligned} g_i(z) = f_i^*(\mathbf{A}_i^\top z) &= \begin{cases} \frac{1}{2\mu_f}\|\sqrt{\hat{L}_{\mathbf{A}}}\mathbf{E}_1 z\|^2 + \frac{1}{2L_f}\|\sqrt{\hat{\mu}_{\mathbf{A}}}z\|^2 - \frac{\hat{L}_{\mathbf{A}}}{2\mu_f}z_1, & i \in \mathcal{V}_1 \\ 0, & i \in \mathcal{V}_2 \\ \frac{1}{2\mu_f}\|\sqrt{\hat{L}_{\mathbf{A}}}\mathbf{E}_2 z\|^2 + \frac{1}{2L_f}\|\sqrt{\hat{\mu}_{\mathbf{A}}}z\|^2 - \frac{\hat{L}_{\mathbf{A}}}{2\mu_f}z_1, & i \in \mathcal{V}_3 \end{cases} \\ &= \begin{cases} \frac{1}{2}z^\top\left(\frac{\hat{L}_{\mathbf{A}}}{\mu_f}\mathbf{M}_1 + \frac{\hat{\mu}_{\mathbf{A}}}{L_f}\mathbf{I}\right)z - \frac{\hat{L}_{\mathbf{A}}}{2\mu_f}z_1, & i \in \mathcal{V}_1 \\ 0, & i \in \mathcal{V}_2 \\ \frac{1}{2}z^\top\left(\frac{\hat{L}_{\mathbf{A}}}{\mu_f}\mathbf{M}_2 + \frac{\hat{\mu}_{\mathbf{A}}}{L_f}\mathbf{I}\right)z - \frac{\hat{L}_{\mathbf{A}}}{2\mu_f}z_1, & i \in \mathcal{V}_3 \end{cases} \end{aligned} \tag{38}$$

Therefore, we have

$$\begin{aligned} \sum_{i=1}^n g_i(z) &= \frac{n}{3}\left[\frac{\hat{L}_{\mathbf{A}}}{2\mu_f}z^\top(\mathbf{M}_1 + \mathbf{M}_2)z + \frac{\hat{\mu}_{\mathbf{A}}}{L_f}z^\top z - \frac{\hat{L}_{\mathbf{A}}}{\mu_f}z_1\right] \\ &= \frac{n}{3}\frac{\hat{L}_{\mathbf{A}}}{\mu_f}\left[\frac{1}{2}z^\top\mathbf{M}z - z_1 + \frac{\hat{\mu}_{\mathbf{A}}\mu_f}{\hat{L}_{\mathbf{A}}L_f}z^\top z\right], \end{aligned}$$

where

$$\mathbf{M} = \mathbf{M}_1 + \mathbf{M}_2 = \begin{pmatrix} 2 & -1 & 0 & 0 & 0 & \dots \\ -1 & 2 & -1 & 0 & 0 & \dots \\ 0 & -1 & 2 & -1 & 0 & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix}.$$

Now we formulate the lower complexity bounds for $\sum_{i=1}^n g_i(z)$, where $g_i(z)$ are defined in (38).

B.4 Deriving the lower bound

Let us introduce the local memory $\mathcal{Z}_i(k)$ which is updated as follows.
1. If the algorithm performs a local gradient computation, then

$$\mathcal{Z}_i(k+1) = \mathrm{Span}\left\{x, \nabla f_i(x), \nabla f_i^*(x) :\ x \in \mathcal{Z}_i(k)\right\}.$$

2. If the algorithm performs a communication rounds, then

$$\mathcal{Z}_i(k+1) = \mathrm{Span}\left\{x :\ x \in \mathcal{Z}_j(k),\ (i,j) \in \mathcal{E}\right\}.$$

Lemma 6. *Function $\sum_{i=1}^n g_i(z)$ attains its minimum at $z^* = \left\{\rho^k\right\}_{k=1}^{\infty}$, where*

$$\rho = \frac{\sqrt{\frac{2}{3}\frac{L_A L_f}{\mu_A \mu_f} + 1} - 1}{\sqrt{\frac{2}{3}\frac{L_A L_f}{\mu_A \mu_f} + 1} + 1}.$$

Proof. In the lower bound example in (Lemma 1 in Appendix C) Kovalev et al. (2021) it was shown that function

$$h(z) = \frac{1}{2} z^\top \mathbf{M} z + \frac{3\mu}{L-\mu}\|z\|^2 - z_1$$

attains its minimum at $z_k^* = \rho^k$, where

$$\rho = \frac{\sqrt{\frac{2L}{3\mu} + \frac{1}{3}} - 1}{\sqrt{\frac{2L}{3\mu} + \frac{1}{3}} + 1}$$

Let us deduce the expression for L, μ in terms of L_A, L_f, μ_A, μ_f. We enforce $h(z) = \sum_{i=1}^n g_i(z)$ and get

$$\frac{3\mu}{L-\mu} = \frac{\mu_A \mu_f}{L_A L_f} \Rightarrow \frac{L}{\mu} = 1 + \frac{L_A L_f}{\mu_A \mu_f}.$$

Therefore, for ρ we obtain

$$\rho = \frac{\sqrt{\frac{2}{3}\frac{L_A L_f}{\mu_A \mu_f} + 1} - 1}{\sqrt{\frac{2}{3}\frac{L_A L_f}{\mu_A \mu_f} + 1} + 1}.$$

□

Let us first show the lower bound on the number of communications.

Lemma 7. *Let $s_i(k)$ denote the maximum index of a nonzero component of vector held by i-th node at step k, i.e.*

$$s_i(k) = \begin{cases} 0, & \mathcal{Z}_i(k) \subseteq \{0\} \\ \min\left\{s \in \{1, 2, \ldots\} : \mathcal{Z}_i(k) \subseteq \mathrm{Span}\left\{e_1, \ldots, e_s\right\}\right\}, & \textit{else}. \end{cases}$$

Let k_q denote the the number of algorithm step by which exactly q communication steps have been performed, where $q \geq 0$. For any $k \in \{1, \ldots, k_q\}$ we have

$$\max_i s_i(k) \leq 2 + \left\lfloor \frac{q}{\frac{n}{3} + 1} \right\rfloor \tag{39}$$

Proof. Note that from the structure of $g_i(z)$, if the method performs a computation step, then

$$s_i(k+1) \leq s_i(k) + \begin{cases} 1 - (s_i(k) \mod 2), & i \in \mathcal{V}_1 \\ 0, & i \in \mathcal{V}_2 \\ (s_i(k) \mod 2), & i \in \mathcal{V}_3 \end{cases}$$

Due to the structure of network, if the method makes a communication round, it follows

$$s_i(k+1) \leq \begin{cases} \max\left(s_{i-1}(k), s_i(k), s_{i+1}(k)\right), & i \in \{2, \dots, n-1\} \\ \max\left(s_1(k), s_2(k)\right), & i = 1 \\ \max\left(s_{n-1}(k), s_n(k)\right), & i = n \end{cases} \tag{40}$$

We will prove that
1. For $q = 2\ell(n/3+1),\ \ell \in \{0, 1, \dots\}$ we have

$$s_i(k_q) \leq \begin{cases} 1+2\ell, & i \in \mathcal{V}_1 \\ 1+2\ell, & i \in \mathcal{V}_2 \\ 2+2\ell, & i \in \mathcal{V}_3 \end{cases} \tag{41}$$

2. For $q = (2\ell+1)(n/3+1),\ \ell \in \{0, 1, \dots\}$ we have

$$s_i(k_q) \leq \begin{cases} 2+(2\ell+1), & i \in \mathcal{V}_1 \\ 1+(2\ell+1), & i \in \mathcal{V}_2 \\ 1+(2\ell+1), & i \in \mathcal{V}_3 \end{cases} \tag{42}$$

The proof follows by induction.

Induction basis. Let $q = 0$. From definitions of $g_i(z)$ it follows that

$$s_i(k_0) \leq \begin{cases} 1, & i \in \mathcal{V}_1 \\ 0, & i \in \mathcal{V}_2 \\ 2, & i \in \mathcal{V}_3 \end{cases}$$

Therefore, for $q = 0$ our statement holds.

Induction step for $q = (2\ell+1)(n/3+1)$. Consider $q_- = q - n/3 = 2\ell(n/3+1)$. From (41) we have that for the spread of nonzero components from $\mathcal{V}_3$ to $\mathcal{V}_1$ it requires $n/3$ communication rounds to reach node $n/3+1$. After one more communication round, the information reaches node $n/3$.

Induction step for $q = (2\ell+1)(n/3+1)$. The proof follows by the same argument as for $q = (2\ell+1)(n/3+1)$.

We just proved the statement of lemma, i.e. relation (39), for q divisible by $(n/3+1)$. Between such checkpoints, the information (i.e. the number of nonzero components) traverses nodes of $\mathcal{V}_2$ and therefore $\max_i s_i(k)$ stays unchanged. Thus the statement of the lemma is proven. □

Now we estimate the distance to optimum.

$$\|z_i(k) - z^*\|_2^2 \geq \sum_{\ell=s_i(k)+1}^{\infty} (z_i(k) - z^*)^2 = \sum_{\ell=s_i(k)+1}^{\infty} \rho^{2\ell} = \frac{\rho^{2s_i(k)+2}}{1-\rho^2} = \frac{\rho^{6+2\left\lfloor \frac{q}{n/3+1} \right\rfloor}}{1-\rho^2}$$
$$\overset{(a)}{\geq} \frac{\rho^{6+\frac{2q}{2n/3}}}{1-\rho^2} = \frac{\rho^6}{1-\rho^2} \cdot \rho^{\frac{3q}{n}} \overset{(b)}{=} \frac{\rho^6}{1-\rho^2} \cdot \rho^{\frac{3q}{\sqrt{\chi}}},$$

where (a) holds since $n/3 \geq 1$; (b) holds due to (37).

Following Kovalev et al. (2021), we obtain that

$$\rho \geq \max\left(0, 1 - \sqrt{\frac{6\mu_A \mu_f}{L_A L_f}}\right).$$

Therefore,

$$\|z_i(k) - z^*\|_2^2 \geq \frac{\rho^6}{1-\rho^2} \left(\max\left(0, 1 - \sqrt{\frac{6\mu_A \mu_f}{L_A L_f}}\right)\right)^{\frac{3q}{\sqrt{\chi}}}.$$

It follows that the number of communications is lower bounded as

$$N_{\mathbf{W}} \geq \Omega\left(\sqrt{\chi}\sqrt{\frac{L_A L_f}{\mu_A \mu_f}} \log\left(\frac{1}{\varepsilon}\right)\right)$$

and the number of oracle calls of $g_i(z)$ at each node (that is, the number of local matrix multiplications by A_i) is lower bounded as

$$N_{\mathbf{A}} \geq \Omega\left(\sqrt{\frac{L_A L_f}{\mu_A \mu_f}} \log\left(\frac{1}{\varepsilon}\right)\right).$$

B.5 Lower bound on the number of gradient computations

To get the lower bound on local gradient calls, let us consider a problem

$$\begin{aligned} \min_{x_1,\ldots x_n \in \mathbb{R}^d} \quad & \sum_{i=1}^n f_i(x_i) + \sum_{i=1}^n v_i(u_i) \\ \text{s.t.} \quad & \sum_{i=1}^n A_i x_i = 0 \end{aligned}$$

where all $v_i(u)$ are the same and $v_i(u_i) = g(u) = \sum_{j=1}^n g_j(u)$ and $g_j(u)$ are defined in (38). All $v_i(u)$ are the same and are defined as

$$v_i(u_i) = \frac{1}{2} u_i^\top \mathbf{M} u_i + \frac{\mu_f}{L_f} u^\top u - u_1.$$

Since there is no communication constraint on u_i, each node runs optimization process individually. Following the same arguments as for function $g(z)$, we get the lower bound on the number of oracle calls

$$N_f \geq \Omega\left(\sqrt{\frac{L_f}{\mu_f}} \log\left(\frac{1}{\varepsilon}\right)\right).$$