

A Parameter-free Decentralized Algorithm for Composite Convex Optimization

Xiaokai Chen, Ilya Kuruzov, Gesualdo Scutari, and Alexander Gasnikov

Abstract—The paper studies decentralized optimization over networks, where agents minimize a composite objective consisting of the sum of smooth convex functions—the agents’ losses—and an additional nonsmooth convex extended value function. We propose a decentralized algorithm wherein agents *adaptively* adjust their stepsize using local backtracking procedures that require *no global (network) information* or extensive inter-agent communications. Our adaptive decentralized method enjoys robust convergence guarantees, outperforming existing decentralized methods, which are not adaptive. Our design is centered on a three-operator splitting, applied to a reformulation of the optimization problem. This reformulation utilizes a proposed BCV metric, which facilitates decentralized implementation and local stepsize adjustments while guarantying convergence.

Index Terms—Adaptive stepsize, convex optimization, decentralized optimization, networks.

I. INTRODUCTION

Consider the following decentralized optimization

$$\min_{x \in \mathbb{R}^d} u(x) := f(x) + r(x), \quad (\text{P})$$

$$f(x) := \frac{1}{m} \sum_{i=1}^m f_i(x), \quad (1)$$

where $f_i : \mathbb{R}^d \rightarrow \mathbb{R}$ is the cost function of agent $i \in [m] := \{1, \dots, m\}$, assumed to be convex, *locally* smooth, and known only to the agent; $r : \mathbb{R}^d \rightarrow \mathbb{R} \cup \{-\infty, +\infty\}$ is a convex, nonsmooth function, known to all agents, which can be used to enforce shared constraints or specific structures on the solution (e.g., sparsity or low-rank). Agents are embedded in a *mesh* communication network, modeled as a fixed, undirected and connected graph $\mathcal{G}$ with no servers.

Problem (P) arises from several applications of interest, including signal processing, machine learning, multiagent systems, and communications. The literature abounds with decentralized solution methods for (P) (when f_i ’s are *globally* smooth); we refer to the recent tutorial (and reference therein) [1] and book [2] for a detailed review. These methods commonly rely on conservative stepsize bounds for convergence, dependent on parameters such as the Lipschitz constants of the agents’ gradients, the spectral gap of the graph adjacency matrix, or other topological properties. However, this crucial information is not locally available to the agents. Consequently, stepsize values are often set through manual tuning, leading to performance that is unpredictable, problem-dependent, and not reproducible.

Chen and Scutari are with the School of Industrial Engineering, Purdue University; emails: <chen4373, gscutari>@purdue.edu. Kuruzov and Gasnikov are with the Innopolis University; emails: kuruzov.ia@phystech.edu, gasnikov@yandex.ru.

Furthermore, these methods fail when agents’ losses are only *locally* smooth—refer to Sec. I-B for more information. This paper brings adaptivity to the decentralized setting (P).

A. Main contributions

1. *Algorithm design*: We introduce an adaptive decentralized algorithm tailored to the composite structure of Problem (P). To deal with the nonsmooth term, our design leverages a three-operator splitting decomposition. This is applied to a reformulation of (P) based on the BCV technique [3], [4] (named after Bertsekas, O’Connor, and Vandenberghe). A suitable metric is chosen in the BCV transformation, to facilitate a fully decentralized implementation of the splitting-based algorithm that uses only *neighboring communications* while enabling adaptive updates of the agents’ stepsize through local backtracking procedures. Remarkably, our algorithm does not require agents to know any global optimization or network parameters.

2. *Convergence guarantees*: We rigorously prove convergence of the proposed method, demonstrating a sublinear $\mathcal{O}(1/k)$ rate for a suitably defined optimality gap, where k denotes the iteration count. Preliminary numerical results highlight that our adaptive algorithm significantly outperforms existing decentralized algorithms applicable to (P), all of which rely on nonadaptive stepsize choices.

B. Related works

1. *Adaptive centralized methods*: In recent years, there has been an increasing interest in developing adaptive (a.k.a. parameter-free) optimization algorithms for centralized and federated settings. These methods include established techniques such as line-search approaches [3], Polyak’s stepsize [5], Barzilai-Borwein’s stepsize [6], as well as more recent advancements that estimate the local curvature of the cost function [7]–[9]. Additionally, other adaptive gradient methods tailored specifically to machine learning challenges include AdaGrad [10], Adam [11], AMSGrad [12], and their variants [13], [14]. Some of these algorithms have been adapted for federated architectures (server-client systems). However, none of them is suitable for mesh networks, because they rely on a central server to aggregate local model updates and tune the common stepsize—a process impractical in decentralized mesh setups. Among these methods, only [5], [8]–[10] can deal with nonsmooth convex terms, as in Problem (P).

2. *Adaptive decentralized methods*: The landscape of parameter-free *decentralized* methods is limited, with notable proposals including [9], [15]–[19]. Specifically, [15]–

[17] study stepsize adaptivity for stochastic (non)convex/online and *smooth* optimization, typically using gradient normalization from past iterates. Except for [17], these methods assume globally Lipschitz continuous losses, enabling convergence with standard stepsizes of the order $\mathcal{O}(1/\sqrt{k})$ (here, k is the iteration index). Additionally, [15], [16] require prior knowledge of certain problem-dependent parameters. The work [19] introduces a Port-Hamiltonian System framework for smooth, strongly convex, and unconstrained problems. While parameter-free convergence holds in the centralized setting, convergence (global asymptotic stability) in decentralized settings is ensured only under specific graph structures or graph-dependent stepsize constraints, which rely on global network information. No explicit convergence rate expression is provided. Recently, [18] proposed a parameter-free decentralized method for smooth, strongly convex instances of (P), establishing linear convergence and demonstrating superior theoretical and practical performance compared to non-adaptive decentralized algorithms.

None of the decentralized methods described above handle *composite* optimization problems as (P), nor do they account for *locally* smooth losses. While proximal adaptive methods for composite optimization have been studied in centralized settings [20], [21], their adaptations to decentralized problems exhibit significant practical limitations. Specifically, the decentralization of [20] requires at *every* line-search step (i) transmission of *ambient-sized* vectors—causing prohibitive communication costs—and (ii) computation of the smallest agents' line-search stepsize across the *entire* network. The method from [21], while avoiding line-search procedures, still necessitates computing and propagating *global* scalars at each iteration for stepsize and relies on knowledge of a *global network quantity* unavailable locally. Resorting to conservative local overestimates of this parameter reduces adaptivity, slows convergence, and makes the algorithm sensitive to critical tuning parameters. These practical drawbacks are confirmed by numerical results in Sec. IV.

In summary, existing adaptive decentralized methods cannot solve composite optimization problems (P) *using only neighbor communications*. To the best of our knowledge, the parameter-free decentralized algorithm presented here is the first to address this critical challenge.

II. ALGORITHM DESIGN

We study (P) under the following assumptions.

Assumption 1. (i) Each $f_i : \mathbb{R}^d \rightarrow \mathbb{R}$ is convex on $\mathbb{R}^d$ and locally smooth; (ii) $r : \mathbb{R}^d \rightarrow \mathbb{R} \cup \{-\infty, \infty\}$ is convex, proper and lower-semicontinuous; and (iii) $u : \mathbb{R}^d \rightarrow \mathbb{R}$ is lower bounded.

Assumption 2. The network is modeled as an undirected, connected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = [m]$ and $(i, j) \in \mathcal{E}$ if and only if there is a link between agent i and j .

Assumption 3 (Gossip matrices). Let $W_{\mathcal{G}}$ denote the set of symmetric, doubly stochastic, gossip matrices $\tilde{W} :=$

$(\tilde{w}_{ij})_{i,j=1}^m$ that are compliant with $\mathcal{G}$, i.e., $\tilde{w}_{ii} > 0$, for all $i \in [m]$; $\tilde{w}_{ij} > 0$ for all $(i, j) \in \mathcal{E}$; and $\tilde{w}_{ij} = 0$ otherwise.

Assumption 1 requires only *local* smoothness of the losses, which significantly enlarges the class of covered functions and applications. Examples include covariance estimation problems and Poisson inverse problems [22]–[24] arising from image processing. Assumption 3 is standard in the literature of gossip-based algorithms—see, e.g., [1], [2].

A. A three-operator splitting-based algorithm

The starting point of our new algorithm design is the following reformulation of (P) *leveraging the BCV technique*: introducing local copies $x_i \in \mathbb{R}^d$ of x and slack variables $\tilde{x}_i \in \mathbb{R}^d$, along with the row-wise stack matrices $\mathbf{X} := [x_1, \dots, x_m]^\top$ and $\tilde{\mathbf{X}} := [\tilde{x}_1, \dots, \tilde{x}_m]^\top$, the following optimization problem

$$\min_{\mathbf{X}, \tilde{\mathbf{X}} \in \mathbb{R}^{m \times d}} \underbrace{F(\mathbf{X})}_{:=\tilde{F}(\mathbf{X}, \tilde{\mathbf{X}})} + \underbrace{R(\mathbf{X}) + \delta_{\{0\}}(\tilde{\mathbf{X}})}_{:=\tilde{R}(\mathbf{X}, \tilde{\mathbf{X}})} + \underbrace{\delta_{\{0\}}(\mathbf{L}\mathbf{X} + M\tilde{\mathbf{X}})}_{:=\tilde{G}(\mathbf{X}, \tilde{\mathbf{X}})}. \quad (\text{P}')$$

is equivalent to the original Problem (P). In (P'), we defined

$$F(\mathbf{X}) := \sum_{i=1}^m f_i(x_i), \quad R(\mathbf{X}) := \sum_{i=1}^m r(x_i);$$

$\delta_{\{0\}} : \mathbb{R}^{m \times d} \rightarrow \mathbb{R} \cup \{\infty\}$ is the indicator function of $\{0\}$; $\mathbf{L} \in \mathbb{S}^m$ satisfies $\text{null}(\mathbf{L}) = \text{span}(\mathbf{1}_m)$, and $M \in \mathbb{S}_{++}^m$ is the BCV metric. The matrix $\mathbf{L}$ enforces consensus among the agents' variables x_i 's via $\mathbf{L}\mathbf{X} = \mathbf{0}$. The choice of M will be shown to be crucial to enable a fully decentralized implementation of the proposed algorithm as well as adaptive local stepsize selection. This justifies the presence of $\tilde{G}$ in (P'), which is unconventional in the classical decentralized optimization literature. Here, $\mathbb{S}^m$ (resp. $\mathbb{S}_{++}^m$) denotes the set of $m \times m$ real symmetric (resp. positive definite) matrices.

Problem (P') is equivalent to solving the monotone inclusion based on the following three-operator splitting [25]:

$$\text{find } \mathbf{T} := \begin{bmatrix} \mathbf{X} \\ \tilde{\mathbf{X}} \end{bmatrix} \in \mathbb{R}^{2m \times d} \text{ s.t. } 0 \in (\mathbb{A} + \mathbb{B} + \mathbb{C})\mathbf{T}, \quad (2)$$

where $\mathbb{A} := \partial \tilde{G}$, $\mathbb{B} := \partial \tilde{R}$ and $\mathbb{C} := \nabla \tilde{F}$. Invoking the Davis-Yin three-operator splitting [26], (2) is equivalent to

$$\text{find } \mathbf{T}, \mathbf{Z} \in \mathbb{R}^{2m \times d} \text{ s.t. } \mathbb{D}_\alpha \mathbf{Z} = \mathbf{Z} \text{ and } \mathbf{T} = \mathbb{J}_{\alpha \mathbb{B}} \mathbf{Z}, \quad (3)$$

where

$$\mathbb{D}_\alpha := \mathbb{I} - \mathbb{J}_{\alpha \mathbb{B}} + \mathbb{J}_{\alpha \mathbb{A}} \circ (2\mathbb{J}_{\alpha \mathbb{B}} - \mathbb{I} - \alpha \mathbb{C} \circ \mathbb{J}_{\alpha \mathbb{B}}), \quad (4)$$

$\mathbb{J}_\mathbb{A} := (\mathbb{I} + \mathbb{A})^{-1}$ is the resolvent of a given operator $\mathbb{A}$; and the constant $\alpha > 0$ plays the role of the stepsize. Notice that $\mathbb{J}_{\alpha \partial \tilde{R}}$ (resp. $\mathbb{J}_{\alpha \partial \tilde{G}}$) is equivalent to the proximal operator $\text{prox}_{\alpha \tilde{R}}$ (resp. $\text{prox}_{\alpha \tilde{G}}$) applied row-wise to its argument.

One can solve (3) through the standard fixed-point Krasnosel'skii-Mann iteration [26]: given $\mathbf{Z}^k$ and $\alpha^k > 0$,

$$\begin{aligned} \mathbf{T}_\mathbb{B}^k &= \text{prox}_{\alpha^k \tilde{R}} \mathbf{Z}^k, \\ \mathbf{T}_\mathbb{A}^{k+1} &= \text{prox}_{\alpha^k \tilde{G}} \left(2\mathbf{T}_\mathbb{B}^k - \mathbf{Z}^k - \alpha^k \nabla \tilde{F}(\mathbf{T}_\mathbb{B}^k) \right), \\ \mathbf{Z}^{k+1} &= \mathbf{Z}^k + \mathbf{T}_\mathbb{A}^{k+1} - \mathbf{T}_\mathbb{B}^k. \end{aligned} \quad (5)$$

Next, we develop (5) further to include a backtracking procedure that allows for adaptive selection of the stepsize.

This development necessitates defining a suitable search direction and a proper merit function. The backtracking procedure and the direction are established based on the saddle-point reformulation of Problem (P'): $\min_{\mathbf{T} \in \mathbb{R}^{2m \times d}} \max_{\mathbf{S} \in \mathbb{R}^{2m \times d}} \mathcal{L}(\mathbf{T}, \mathbf{S})$, where

$$\mathcal{L}(\mathbf{T}, \mathbf{S}) := \tilde{F}(\mathbf{T}) + \tilde{G}(\mathbf{T}) + \langle \mathbf{S}, \mathbf{T} \rangle - \tilde{R}^*(\mathbf{S}), \quad (6)$$

is the Lagrangian function, with $\mathbf{T}$ as the primal variables and $\mathbf{S}$ as the dual variables, and $\tilde{R}^*$ denoting the conjugate of $\tilde{R}$. Introducing the intermediate variable

$$\mathbf{S}^k := \frac{1}{\alpha^k} (\mathbf{Z}^k - \mathbf{T}_{\mathbb{B}}^k),$$

allows us to express $\mathbf{T}_{\mathbb{A}}^{k+1}$ in (5) as

$$\mathbf{T}_{\mathbb{A}}^{k+1} = \text{prox}_{\alpha^k \tilde{G}}(\mathbf{T}_{\mathbb{B}}^k - \alpha^k \mathbf{S}^k - \alpha^k \nabla \tilde{F}(\mathbf{T}_{\mathbb{B}}^k)). \quad (7)$$

If the stepsize $\alpha^k > 0$ satisfies the following backtracking condition: given $\delta \in (0, 1)$, find the largest $\alpha^k > 0$ such that

$$\begin{aligned} \tilde{F}(\mathbf{T}_{\mathbb{A}}^{k+1}) &\leq \tilde{F}(\mathbf{T}_{\mathbb{B}}^k) + \langle \nabla \tilde{F}(\mathbf{T}_{\mathbb{B}}^k), \mathbf{T}_{\mathbb{A}}^{k+1} - \mathbf{T}_{\mathbb{B}}^k \rangle \\ &\quad + \frac{\delta}{2\alpha^k} \|\mathbf{T}_{\mathbb{A}}^{k+1} - \mathbf{T}_{\mathbb{B}}^k\|^2, \end{aligned} \quad (8)$$

it ensures a sufficient decrease of the Lagrangian function (6) at $\mathbf{S}^k$ along the direction $\mathbf{T}_{\mathbb{A}}^{k+1} - \mathbf{T}_{\mathbb{B}}^k$, that is:

$$\mathcal{L}(\mathbf{T}_{\mathbb{A}}^{k+1}, \mathbf{S}^k) \leq \mathcal{L}(\mathbf{T}_{\mathbb{B}}^k, \mathbf{S}^k) - \left(1 - \frac{\delta}{2}\right) \frac{1}{\alpha^k} \|\mathbf{T}_{\mathbb{A}}^{k+1} - \mathbf{T}_{\mathbb{B}}^k\|^2.$$

Thus, $\mathbf{T}_{\mathbb{A}}^{k+1} - \mathbf{T}_{\mathbb{B}}^k$ becomes our viable primal search direction, and $\mathbf{S}^k$ will serve as the dual variables.

Using the $\mathbf{S}$ -variable and absorbing the $\mathbf{Z}$ -variables, (5) can be rewritten as: given the largest $\alpha^k > 0$ satisfying (8),

$$\mathbf{T}_{\mathbb{A}}^{k+1} = \text{prox}_{\alpha^k \tilde{G}}(\mathbf{T}_{\mathbb{B}}^k - \alpha^k \mathbf{S}^k - \alpha^k \nabla \tilde{F}(\mathbf{T}_{\mathbb{B}}^k)) \quad (9a)$$

$$\mathbf{T}_{\mathbb{B}}^{k+1} = \text{prox}_{\alpha^k \tilde{R}}(\mathbf{T}_{\mathbb{A}}^{k+1} + \alpha^k \mathbf{S}^k), \quad (9b)$$

$$\mathbf{S}^{k+1} = \mathbf{S}^k + \frac{1}{\alpha^k} (\mathbf{T}_{\mathbb{A}}^{k+1} - \mathbf{T}_{\mathbb{B}}^{k+1}). \quad (9c)$$

B. A decentralized implementation

The algorithm in (9) is not fully decentralized, due to the reliance on $\text{prox}_{\alpha^k \tilde{G}}$ and the backtracking procedure (8) involving the function $\tilde{F}$, which cannot be performed locally and independently by the agents. In this section we cope with these issues, and proposed a fully decentralized implementation of the algorithm in (9).

Let us partition the variables as

$$\mathbf{T}_{\mathbb{A}}^k := \begin{bmatrix} \mathbf{T}_{\mathbb{A},1}^k \\ \mathbf{T}_{\mathbb{A},2}^k \end{bmatrix}, \quad \mathbf{T}_{\mathbb{B}}^k := \begin{bmatrix} \mathbf{T}_{\mathbb{B},1}^k \\ \mathbf{T}_{\mathbb{B},2}^k \end{bmatrix}, \quad \mathbf{S}^k := \begin{bmatrix} \mathbf{S}_1^k \\ \mathbf{S}_2^k \end{bmatrix}. \quad (10)$$

The update of the $\mathbf{T}_{\mathbb{B}}$ -variables involving $\text{prox}_{\alpha^k \tilde{R}}$ reads

$$\mathbf{T}_{\mathbb{B}}^{k+1} = \begin{bmatrix} \text{prox}_{\alpha^k \tilde{R}}(\mathbf{T}_{\mathbb{A},1}^{k+1} + \alpha^k \mathbf{S}_1^k) \\ \mathbf{0} \end{bmatrix}. \quad (11)$$

The expression of $\text{prox}_{\alpha^k \tilde{G}}$ is obtained using the following **Fact** (e.g., [25, Eq. (2.6)]): Let $g(u) = f^*(A^\top u)$, with $f: \mathbb{R}^n \rightarrow \mathbb{R} \cup \{-\infty, \infty\}$ assumed to be convex, closed and

proper. Suppose, $\text{ridom } f^* \cap \text{range}(A^\top) \neq \emptyset$. Then $v = \text{prox}_{\alpha, g}(u)$ if and only if there exists $x \in \mathbb{R}^n$ such that

$$\begin{aligned} x &\in \text{argmin}_y \{f(y) - \langle u, Ay \rangle + \frac{\alpha}{2} \|Ay\|^2\} \quad \text{and} \\ v &= u - \alpha Ax. \end{aligned} \quad (12)$$

Notice that $\tilde{G} = (\delta_{\{0\}}^*)^* \circ [\mathbf{L}, M]$, with $\delta_{\{0\}}^*$ denoting the conjugate of $\delta_{\{0\}}$. Since $\delta_{\{0\}}^*$ is convex, closed and proper and $\text{ridom } \delta_{\{0\}}^* \cap \text{Range}([\mathbf{L}, M]) \neq \emptyset$, we can use (12) to express $\text{prox}_{\alpha^k \tilde{G}}$. Hence, (9a) can be rewritten as

$$\mathbf{T}_{\mathbb{A}}^{k+1} = \begin{bmatrix} \mathbf{T}_{\mathbb{B},1}^k - \alpha^k \mathbf{S}_1^k - \alpha^k \nabla F(\mathbf{T}_{\mathbb{B},1}^k) \\ \mathbf{T}_{\mathbb{B},2}^k - \alpha^k \mathbf{S}_2^k \end{bmatrix} - \alpha^k \begin{bmatrix} \mathbf{L} \\ M \end{bmatrix} \mathbf{Y}^{k+1}, \quad (13)$$

where

$$\begin{aligned} \mathbf{Y}^{k+1} &\in \text{argmin}_{\mathbf{Y}} \{ \delta_{\{0\}}^*(\mathbf{Y}) - \langle \mathbf{T}_{\mathbb{B},2}^k - \alpha^k \mathbf{S}_2^k, M\mathbf{Y} \rangle \\ &\quad - \langle \mathbf{T}_{\mathbb{B},1}^k - \alpha^k \mathbf{S}_1^k - \alpha^k \nabla F(\mathbf{T}_{\mathbb{B},1}^k), \mathbf{L}\mathbf{Y} \rangle \\ &\quad + \frac{\alpha^k}{2} (\|\mathbf{L}\mathbf{Y}\|^2 + \|M\mathbf{Y}\|^2) \}. \end{aligned} \quad (14)$$

We proceed choosing M so that (14) admits a closed form solution, computable locally at the agents' sides. Combining (9c), (11) and (13), it holds $\mathbf{S}_2^k = -M\mathbf{Y}^k$. Substituting this expression in (14) and using the fact

$$\delta_{\{0\}}^*(\mathbf{Y}) = \sup_{\mathbf{Z} \in \mathbb{R}^{m \times d}} \{ \langle \mathbf{Y}, \mathbf{Z} \rangle - \delta_{\{0\}}(\mathbf{Z}) \} = \langle \mathbf{Y}, \mathbf{0} \rangle = 0,$$

yields

$$\begin{aligned} \mathbf{Y}^{k+1} &\in \text{argmin}_{\mathbf{Y}} \{ \alpha^k \|\mathbf{Y} - \mathbf{Y}^k\|_{\mathbf{L}^2 + M^2}^2 \\ &\quad - 2 \langle \mathbf{L}(\mathbf{T}_{\mathbb{B},1}^k - \alpha^k \mathbf{S}_1^k - \alpha^k \mathbf{L}\mathbf{Y}^k - \alpha^k \nabla F(\mathbf{T}_{\mathbb{B},1}^k)), \mathbf{Y} \rangle \}. \end{aligned}$$

Choosing M to linearize the quadratic term in the objective function above $-M := \sqrt{I - \mathbf{L}^2}$ -yields

$$\mathbf{Y}^{k+1} = \mathbf{Y}^k + \frac{1}{\alpha^k} \mathbf{L}(\mathbf{T}_{\mathbb{B},1}^k - \alpha^k \mathbf{S}_1^k - \alpha^k \mathbf{L}\mathbf{Y}^k - \alpha^k \nabla F(\mathbf{T}_{\mathbb{B},1}^k)). \quad (15)$$

Introducing the variable $\mathbf{D}^k := \mathbf{L}\mathbf{Y}^k$, and setting $\mathbf{L}^2 = I - W$, with $W = (1 - c)I + c\tilde{W} \in W_G$ for some $c \in (0, 1/2)$, (15) can be rewritten in terms of the $\mathbf{D}$ -variables as

$$\mathbf{D}^{k+1} = \mathbf{D}^k + \frac{1}{\alpha^k} (I - W)(\mathbf{T}_{\mathbb{B},1}^k - \alpha^k (\mathbf{S}_1^k + \mathbf{D}^k + \nabla F(\mathbf{T}_{\mathbb{B},1}^k))), \quad (16)$$

and so the update of the $\mathbf{T}_{\mathbb{A}}$ -variables (using (13)):

$$\mathbf{T}_{\mathbb{A},1}^k = W\mathbf{T}_{\mathbb{B},1}^k + W(\nabla F(\mathbf{T}_{\mathbb{B},1}^k) + \mathbf{S}_1^k + \mathbf{D}^k). \quad (17)$$

In summary, the proposed algorithm is given by (11), (16), and (17). The algorithm is now fully decentralized in the updates of $\mathbf{T}$ - and $\mathbf{D}$ -variables.

We are left to address the implementability of the line-search procedure (8) over the network. To this end, we introduce a local stepsize α_i^k for each agent i . This stepsize is determined by performing backtracking on the local loss function f_i . Specifically, for $t_{\mathbb{A},i}^{k+1}$ and $t_{\mathbb{B},i}^k$, which are the i -th rows of $\mathbf{T}_{\mathbb{A},1}^{k+1}$ and $\mathbf{T}_{\mathbb{B},1}^k$ respectively, each agent finds the largest α_i^k that satisfies:

$$\begin{aligned} f_i(t_{\mathbb{A},i}^{k+1}) &\leq f_i(t_{\mathbb{B},i}^k) + \langle \nabla f_i(t_{\mathbb{B},i}^k), t_{\mathbb{A},i}^{k+1} - t_{\mathbb{B},i}^k \rangle \\ &\quad + \frac{\delta}{2\alpha_i^k} \|t_{\mathbb{A},i}^{k+1} - t_{\mathbb{B},i}^k\|^2. \end{aligned} \quad (18)$$

Setting $\alpha^k := \min_{i \in [m]} \alpha_i^k$ ensures that the global line-search condition (8) is also met.

The proposed decentralized algorithm is outlined in Algorithm 1, with the detailed backtracking line-search procedure provided in Algorithm 2. For clarity in presentation, we refer to $\mathbf{T}_{\mathbb{B},1}^k$ and $\mathbf{S}_1^k$ as $\mathbf{X}^k$ and $\mathbf{S}^k$, respectively.

On the global-min consensus: Step (S.3) entails executing a min-consensus operation across the network to synchronize a common stepsize, $\alpha^k = \min_{i \in [m]} \alpha_i^k$, among the agents. This min-consensus protocol adapts seamlessly to contemporary wireless mesh network technologies. In fact, current network infrastructures support diverse communication interfaces, in particular WiFi and LoRa (Low-Range) [27], [28]. WiFi provides high-speed, short-range communication capabilities whereas LoRa supports long-range communication over low rates, perfectly suited for broadcasting minimal information to *all nodes* across the network in a single hop. In such multi-interface environments, Algorithm 1 is perfectly implementable: it employs WiFi for transmitting vector variables in Steps (S.1) and uses LoRa for executing the min-consensus in Step (S.3). Additionally, before transmission, the α_i^k values can be quantized to lower approximations using a few bits.

Local-min operations leveraging only neighboring communications will be considered in Sec. III.

Algorithm 1 Decentralized Adaptive Three Operator Splitting (global_DATOS)

Data: (i) Initialization: $\alpha^{-1} \in (0, \infty)$, $\mathbf{X}^0 \in \mathbb{R}^{m \times n}$, $\mathbf{S}^0 \in \mathbb{R}^{m \times n}$ and $\mathbf{D}^0 = \mathbf{0}$; (ii) Backtracking parameters: $\delta \in (0, 1]$; (iii) Gossip matrix $W = (1 - c)I + c\bar{W}$, $c \in (0, 1/2)$.

1: (S.1) Communication Step:

$$\mathbf{X}^{k+1/2} = W\mathbf{X}^k, \quad \mathbf{D}^{k+1/2} = W(\nabla F(\mathbf{X}^k) + \mathbf{S}^k + \mathbf{D}^k);$$

2: (S.2) Decentralized line-search: Each agent updates $\bar{\alpha}_i^k$ according to:

$$\bar{\alpha}_i^k = \text{Linesearch}(\alpha^{k-1}, f_i, x_i^k, x_i^{k+1/2}, -d_i^{k+1/2}, \gamma_k, \delta);$$

3: (S.3) Global min-consensus:

$$\alpha^k = \min_{i \in [m]} \bar{\alpha}_i^k;$$

4: (S.4) Updates of the primal and dual variables:

$$\begin{aligned} \mathbf{X}^{k+1} &= \text{prox}_{\alpha^k R}(\mathbf{X}^{k+1/2} - \alpha^k \mathbf{D}^{k+1/2} + \alpha^k \mathbf{S}^k) \\ \mathbf{S}^{k+1} &= \mathbf{S}^k + \frac{1}{\alpha^k}(\mathbf{X}^{k+1/2} - \mathbf{X}^{k+1} - \alpha^k \mathbf{D}^{k+1/2}) \\ \mathbf{D}^{k+1} &= \mathbf{D}^{k+1/2} + \frac{1}{\alpha^k}(\mathbf{X}^k - \mathbf{X}^{k+1/2} - \alpha^k \nabla F(\mathbf{X}^k) - \alpha^k \mathbf{S}^k). \end{aligned}$$

5: (S.5) If a termination criterion is not met, $k \leftarrow k + 1$ and go to step (S.1).

C. Convergence guarantees

This section study convergence of Algorithm 1. For analytical convenience, we will refer to the updates written as in (9). In fact, since $\mathbf{D}^0 = \mathbf{0} \in \text{span}(I - W)$, Algorithm II-B is equivalent to (9), provided that the stepsize α^k therein is updated according to the backtracking procedure described

Algorithm 2 Linesearch($\alpha, f, x_1, x_2, d, \delta$)

```

1:  $\alpha^+ = \alpha$ ;
2:  $x^+ = x_2 + \alpha^+ d$ ; Set  $t = 1$ ;
3: while  $f(x^+) > f(x_1) + \langle \nabla f(x_1), x^+ - x_1 \rangle + \frac{\delta}{2\alpha^+} \|x^+ - x_1\|^2$  do
4:    $\alpha^+ \leftarrow \alpha^+ / 2$ ;
5:    $x^+ \leftarrow x_2 + \alpha^+ d$ ;
6:    $t \leftarrow t + 1$ ;
return  $\alpha^+$ .
```

in (S.3) of Algorithm 1. Henceforth, all references to (9) will implicitly assume this condition.

Given the reformulation of (P') based on the Lagrangian function $\mathcal{L}$ in (6) and strong duality (guaranteed by Assumption 1), a saddle point $(\mathbf{T}^*, \mathbf{S}^*)$ of $\mathcal{L}$ is such that $\mathbf{T}^*$ is a minimizer of (P'). For any given saddle-point $(\mathbf{T}^*, \mathbf{S}^*)$ of $\mathcal{L}$, let $\mathcal{B}_T \times \mathcal{B}_S$ be a bounded set containing $(\mathbf{T}^*, \mathbf{S}^*)$. We introduce the partial primal-dual gap function [29],

$$\mathcal{G}_{\mathcal{B}_T \times \mathcal{B}_S}(\mathbf{T}_A, \mathbf{S}) := \max_{\mathbf{S} \in \mathcal{B}_S} \mathcal{L}(\mathbf{T}_A, \mathbf{S}) - \min_{\mathbf{T} \in \mathcal{B}_T} \mathcal{L}(\mathbf{T}, \mathbf{S}). \quad (19)$$

Notice that $\mathcal{G}_{\mathcal{B}_T \times \mathcal{B}_S}(\mathbf{T}_A, \mathbf{S}) \geq 0$, for any $(\mathbf{T}_A, \mathbf{S}) \in \text{dom}(\mathcal{L})$. Yet, $\mathcal{G}_{\mathcal{B}_T \times \mathcal{B}_S}$ is not a valid measure of optimality because it might vanish for $(\mathbf{T}_A, \mathbf{S})$ not being a saddle-point of $\mathcal{L}$ [29]. However, if $\mathcal{G}_{\mathcal{B}_T \times \mathcal{B}_S}(\mathbf{T}_A, \mathbf{S}) = 0$ for $(\mathbf{T}_A, \mathbf{S}) \in \text{dom}(\mathcal{L})$ and, in addition, $(\mathbf{T}_A, \mathbf{S})$ lies in the interior of $\mathcal{B}_T \times \mathcal{B}_S$, then $(\mathbf{T}_A, \mathbf{S})$ is a saddle-point of $\mathcal{L}$. The following theorem shows that the iterates $(\mathbf{T}_A^k, \mathbf{S}^k) \in \text{dom}(\mathcal{L})$ remain bounded. Hence, one can choose a “sufficiently large” $\mathcal{B}_T \times \mathcal{B}_S$ such that $\mathcal{G}_{\mathcal{B}_T \times \mathcal{B}_S}(\mathbf{T}_A^k, \mathbf{S}^k)$ is a valid measure of optimality for $(\mathbf{T}_A^k, \mathbf{S}^k)$. We provide next the convergence rate decay of $\mathcal{G}_{\mathcal{B}_T \times \mathcal{B}_S}$ along the ergodic sequences of the iterates $\{\mathbf{T}_A^k\}$ and $\{\mathbf{S}^k\}$, defined as

$$\bar{\mathbf{T}}_A^k := \frac{1}{\theta^k} \sum_{t=1}^k \alpha^{t-1} \mathbf{T}_A^t, \quad \bar{\mathbf{S}}^k := \frac{1}{\theta^k} \sum_{t=1}^k \alpha^{t-1} \mathbf{S}^t, \quad \theta^k := \sum_{t=1}^k \alpha^{t-1}.$$

Theorem 1. Under Assumptions 1-3, the following holds along the iterates of Algorithm II-B:

- (a) The sequence $\{\mathbf{T}_A^k, \mathbf{S}^k\}_k$ is bounded; and
- (b) For any given saddle-point $(\mathbf{T}^*, \mathbf{S}^*)$ of $\mathcal{L}$, there exists a bounded set $\mathcal{B}_T \times \mathcal{B}_S$ [see (36) in the proof for the explicit expression] such that (i) $(\mathbf{T}^*, \mathbf{S}^*) \in \mathcal{B}_T \times \mathcal{B}_S$, (ii) $(\mathbf{T}_A^k, \mathbf{S}^k) \in \text{int}(\mathcal{B}_T \times \mathcal{B}_S)$, for all $k \geq 0$; and (iii)

$$\mathcal{G}_{\mathcal{B}_T \times \mathcal{B}_S}(\bar{\mathbf{T}}_A^k, \bar{\mathbf{S}}^k) \leq \frac{1}{k} \cdot \frac{8R_0^2}{\min(\alpha^{-1}, \delta/(2L^k))}, \quad (20)$$

where L^k is the Lipschitz constant of ∇F over the convex hull of $\cup_{t=1}^k [\mathbf{T}_{\mathbb{B},1}^{t-1}, \mathbf{T}_{\mathbb{A},1}^t]$, and

$$R_0^2 := \|\mathbf{T}_{\mathbb{B}}^0 - \mathbf{T}^*\|^2 + (\alpha^0)^2 \|\mathbf{S}^0 - \mathbf{S}^*\|^2. \quad (21)$$

Theorem 1 establishes the sublinear convergence rate of Algorithm II-B, matching $\mathcal{O}(1/k)$ rate of state-of-art algorithms solving decentralized convex optimization problems. Crucially, Algorithm II-B removes the requirement for prior knowledge of optimization and network parameters, which is common to all prior native decentralized

algorithms applicable to Problem (P), such as [30], [31]. Notably the convergence rate (20) depends on the *local* Lipschitz constants $\{L^k\}$, defined on the convex hull of the trajectory traveled by the algorithm. This local constant often proves much smaller than the *global* Lipschitz constant L , governing the rates of existing (non adaptive) decentralized algorithms. This advantage showcases the algorithm's ability to adapt to the local geometry of the optimization function. Finally, the constant R_0^2 in (20) is to be expected, it quantifies the "distance" of the initial point from the saddle-point.

D. Proof sketch of Theorem 1

Proof of statement (a): It can be shown that the line-search terminates in a finite number of iteration, with α^k satisfying

$$\min\left(\alpha^{k-1}, \frac{\delta}{2\tilde{L}^k}\right) \leq \alpha^k \leq \alpha^{k-1}, \quad (22)$$

where $\tilde{L}^k \in (0, \infty)$ is the Lipschitz constant of ∇F over the segment $[\mathbf{T}_{\mathbb{B},1}^k, \mathbf{T}_{\mathbb{A},1}^{k+1}]$. Furthermore,

$$\begin{aligned} \Delta^k := & F(\mathbf{T}_{\mathbb{B},1}^k) + \langle \nabla F(\mathbf{T}_{\mathbb{B},1}^k), \mathbf{T}_{\mathbb{A},1}^{k+1} - \mathbf{T}_{\mathbb{B},1}^k \rangle \\ & + \frac{\delta}{2\alpha^k} \|\mathbf{T}_{\mathbb{A},1}^{k+1} - \mathbf{T}_{\mathbb{B},1}^k\|^2 - F(\mathbf{T}_{\mathbb{A},1}^{k+1}) \geq 0. \end{aligned} \quad (23)$$

We study the descent of the gap function $\mathcal{G}^k$. By definition (19) of $\mathcal{L}$, we have: for any fixed $(\mathbf{T}, \mathbf{S}) \in \text{dom}(\mathcal{L})$,

$$\begin{aligned} & \mathcal{L}(\mathbf{T}_{\mathbb{A}}^{k+1}, \mathbf{S}) - \mathcal{L}(\mathbf{T}, \mathbf{S}^{k+1}) \\ & \leq \underbrace{\tilde{F}(\mathbf{T}_{\mathbb{A}}^{k+1}) - \tilde{F}(\mathbf{T}_{\mathbb{B}}^k)}_{\text{term I}} + \underbrace{\tilde{F}(\mathbf{T}_{\mathbb{B}}^k) - \tilde{F}(\mathbf{T})}_{\text{term II}} \\ & + \underbrace{\tilde{G}(\mathbf{T}_{\mathbb{A}}^{k+1}) - \tilde{G}(\mathbf{T})}_{\text{term III}} + \underbrace{\langle \mathbf{T}_{\mathbb{A}}^{k+1} - \mathbf{T}, \mathbf{S}^{k+1} \rangle}_{\text{term IV}} \\ & + \underbrace{\tilde{R}^*(\mathbf{S}^{k+1}) - \tilde{R}^*(\mathbf{S})}_{\text{term V}} + \underbrace{\langle \mathbf{T}_{\mathbb{A}}^{k+1}, \mathbf{S} - \mathbf{S}^{k+1} \rangle}_{\text{term VI}}. \end{aligned} \quad (24)$$

We proceed bounding each term separately. By (23)

$$\begin{aligned} \text{term I} & \leq \langle \nabla F(\mathbf{T}_{\mathbb{B},1}^k), \mathbf{T}_{\mathbb{A},1}^{k+1} - \mathbf{T}_{\mathbb{B},1}^k \rangle \\ & + \frac{\delta}{2\alpha^k} \|\mathbf{T}_{\mathbb{A},1}^{k+1} - \mathbf{T}_{\mathbb{B},1}^k\|^2. \end{aligned} \quad (25)$$

Using convexity of F , yields

$$\text{term II} \leq \langle \nabla \tilde{F}(\mathbf{T}_{\mathbb{B}}^k), \mathbf{T}_{\mathbb{B}}^k - \mathbf{T} \rangle. \quad (26)$$

From the update (9), we have

$$\frac{1}{\alpha^k} (\mathbf{T}_{\mathbb{B}}^k - \alpha^k \mathbf{S}^k - \alpha^k \nabla \tilde{F}(\mathbf{T}_{\mathbb{B}}^k) - \mathbf{T}_{\mathbb{A}}^{k+1}) \in \partial \tilde{G}(\mathbf{T}_{\mathbb{A}}^{k+1}),$$

which, together with convexity of $\tilde{G}$, yields

$$\begin{aligned} \text{term III} & \leq \frac{1}{\alpha^k} \langle \mathbf{T}_{\mathbb{B}}^k - \mathbf{T}_{\mathbb{A}}^{k+1}, \mathbf{T}_{\mathbb{A}}^{k+1} - \mathbf{T} \rangle \\ & - \langle \mathbf{S}^k + \nabla \tilde{F}(\mathbf{T}_{\mathbb{B}}^k), \mathbf{T}_{\mathbb{A}}^{k+1} - \mathbf{T} \rangle. \end{aligned} \quad (27)$$

Using (25), (26) and (27), we conclude

$$\begin{aligned} & \text{term I} + \text{term II} + \text{term III} + \text{term IV} \\ & \stackrel{(9c)}{\leq} \frac{\delta}{2\alpha^k} \|\mathbf{T}_{\mathbb{A},1}^{k+1} - \mathbf{T}_{\mathbb{B},1}^k\|^2 + \frac{1}{\alpha^k} \langle \mathbf{T}_{\mathbb{B}}^k - \mathbf{T}_{\mathbb{A}}^{k+1}, \mathbf{T}_{\mathbb{A}}^{k+1} - \mathbf{T} \rangle \\ & - \frac{1}{\alpha^k} \langle \mathbf{T}_{\mathbb{B}}^{k+1} - \mathbf{T}_{\mathbb{A}}^{k+1}, \mathbf{T}_{\mathbb{A}}^{k+1} - \mathbf{T} \rangle \\ & = \frac{1}{2\alpha^k} (\|\mathbf{T}_{\mathbb{B}}^k - \mathbf{T}\|^2 - \|\mathbf{T}_{\mathbb{B}}^k - \mathbf{T}_{\mathbb{A}}^{k+1}\|^2 + \delta \|\mathbf{T}_{\mathbb{B},1}^k - \mathbf{T}_{\mathbb{A},1}^{k+1}\|^2 \\ & - \|\mathbf{T}_{\mathbb{B}}^{k+1} - \mathbf{T}\|^2 + \|\mathbf{T}_{\mathbb{B}}^{k+1} - \mathbf{T}_{\mathbb{A}}^{k+1}\|^2) \\ & \leq \frac{1}{2\alpha^k} (\|\mathbf{T}_{\mathbb{B}}^k - \mathbf{T}\|^2 - \|\mathbf{T}_{\mathbb{B}}^{k+1} - \mathbf{T}\|^2 + \|\mathbf{T}_{\mathbb{B}}^{k+1} - \mathbf{T}_{\mathbb{A}}^{k+1}\|^2), \end{aligned} \quad (28)$$

where in the last inequality we used $0 < \delta < 1$.

Using (9a) and (9c), it follows

$$\mathbf{S}^{k+1} \in \partial \tilde{R}(\mathbf{T}_{\mathbb{B}}^{k+1}) \Rightarrow \mathbf{T}_{\mathbb{B}}^{k+1} \in \partial \tilde{R}^*(\mathbf{S}^{k+1}). \quad (29)$$

Leveraging convexity of $\tilde{R}^*$, we have

$$\text{term V} \leq \langle \mathbf{T}_{\mathbb{B}}^{k+1}, \mathbf{S}^{k+1} - \mathbf{S} \rangle. \quad (30)$$

Thus, we can bound term V + term VI as

$$\begin{aligned} & \text{term V} + \text{term VI} \leq \langle \mathbf{T}_{\mathbb{B}}^{k+1} - \mathbf{T}_{\mathbb{A}}^{k+1}, \mathbf{S}^{k+1} - \mathbf{S} \rangle \\ & \stackrel{(9c)}{=} \frac{1}{2\alpha^k} (\|\alpha^k (\mathbf{S}^k - \mathbf{S})\|^2 - \|\alpha^k (\mathbf{S}^{k+1} - \mathbf{S})\|^2 - \\ & \quad \|\mathbf{T}_{\mathbb{B}}^{k+1} - \mathbf{T}_{\mathbb{A}}^{k+1}\|^2). \end{aligned} \quad (31)$$

Combining (28) and (31) and using $\mathbf{T}_{\mathbb{B},2}^k = \mathbf{0}$, yields

$$\begin{aligned} & 2\alpha^k (\mathcal{L}(\mathbf{T}_{\mathbb{A}}^{k+1}, \mathbf{S}) - \mathcal{L}(\mathbf{T}, \mathbf{S}^{k+1})) \\ & \leq \|\mathbf{T}_{\mathbb{B}}^k - \mathbf{T}\|^2 + (\alpha^k)^2 \|\mathbf{S}^k - \mathbf{S}\|^2 \\ & - (\|\mathbf{T}_{\mathbb{B}}^{k+1} - \mathbf{T}\|^2 + (\alpha^k)^2 \|\mathbf{S}^{k+1} - \mathbf{S}\|^2). \end{aligned} \quad (32)$$

Telescoping (32) and leveraging $\alpha^k \leq \alpha^{k-1}$ yield

$$\begin{aligned} & \|\mathbf{T}_{\mathbb{B}}^k - \mathbf{T}\|^2 + (\alpha^k)^2 \|\mathbf{S}^k - \mathbf{S}\|^2 \\ & + 2 \sum_{t=1}^k \alpha^t (\mathcal{L}(\mathbf{T}_{\mathbb{A}}^t, \mathbf{S}) - \mathcal{L}(\mathbf{T}, \mathbf{S}^t)) \\ & \leq \|\mathbf{T}_{\mathbb{B}}^0 - \mathbf{T}\|^2 + (\alpha^0)^2 \|\mathbf{S}^0 - \mathbf{S}\|^2. \end{aligned} \quad (33)$$

Let $(\mathbf{T}^*, \mathbf{S}^*)$ be any saddle-point of $\mathcal{L}$; set in (33) $(\mathbf{T}, \mathbf{S}) = (\mathbf{T}^*, \mathbf{S}^*)$. We have $\mathcal{L}(\mathbf{T}_{\mathbb{A}}^{k+1}, \mathbf{S}^*) - \mathcal{L}(\mathbf{T}^*, \mathbf{S}^{k+1}) \geq 0$. Invoking the fact that $\alpha^k \geq 0$ for any k , we conclude

$$\begin{aligned} & \|\mathbf{T}_{\mathbb{B}}^k - \mathbf{T}^*\|^2 + (\alpha^k)^2 \|\mathbf{S}^k - \mathbf{S}^*\|^2 \\ & \leq \|\mathbf{T}_{\mathbb{B}}^0 - \mathbf{T}^*\|^2 + (\alpha^0)^2 \|\mathbf{S}^0 - \mathbf{S}^*\|^2, \quad \forall k \geq 0. \end{aligned} \quad (34)$$

Hence, $\{\mathbf{T}_{\mathbb{B}}^k, \mathbf{S}^k\}_k$ is bounded. And so is $\{\mathbf{T}_{\mathbb{A}}^k\}_k$, due to (9c). Specifically, given $R_0^2 > 0$, defined in (21), we have

$$\|\mathbf{T}_{\mathbb{A}}^k - \mathbf{T}^*\|^2 + (\alpha^0)^2 \|\mathbf{S}^k - \mathbf{S}^*\|^2 \leq 4R_0^2, \quad \forall k \geq 0. \quad (35)$$

Proof of statement (b): From (34)-(35) it follows that we can define the bounded set $\mathcal{B}_T \times \mathcal{B}_S$ containing $(\mathbf{T}^*, \mathbf{S}^*)$ as

$$\mathcal{B}_T \times \mathcal{B}_S = \{(\mathbf{T}, \mathbf{S}) \mid \|\mathbf{T} - \mathbf{T}^*\|^2 + (\alpha^0)^2 \|\mathbf{S} - \mathbf{S}^*\|^2 \leq 4R_0^2\}. \quad (36)$$

This also ensures $(\mathbf{T}_{\mathbb{A}}^k, \mathbf{S}^k) \in \text{int}(\mathcal{B}_T \times \mathcal{B}_S)$, for any $k \geq 0$.

Dividing both sides of (33) by $2\theta_k$ and leveraging convexity of $\mathcal{L}(\bullet, \mathbf{S}) - \mathcal{L}(\mathbf{T}, \bullet)$, yields

$$\mathcal{L}(\bar{\mathbf{T}}_A^k, \mathbf{S}) - \mathcal{L}(\mathbf{T}, \bar{\mathbf{S}}^k) \leq \frac{1}{2\theta^k} (\|\mathbf{T}_B^0 - \mathbf{T}\|^2 + (\alpha^0)\|\mathbf{S}^0 - \mathbf{S}\|^2). \quad (37)$$

Taking the maximum of $(\mathbf{T}, \mathbf{S})$ over $\mathcal{B}_T \times \mathcal{B}_S$ on both sides of (37) and using (19), we infer

$$\mathcal{G}_{\mathcal{B}_T \times \mathcal{B}_S}(\bar{\mathbf{T}}_A^k, \bar{\mathbf{S}}^k) \leq \frac{D(\mathcal{B}_T, \mathcal{B}_S)}{2\theta^k},$$

where $D(\mathcal{B}_T, \mathcal{B}_S) := \sup_{(\mathbf{T}, \mathbf{S}) \in \mathcal{B}_T \times \mathcal{B}_S} \|\mathbf{T}_B^0 - \mathbf{T}\|^2 + (\alpha^0)^2 \|\mathbf{S}^0 - \mathbf{S}\|^2$. Moreover,

$$\theta^k \geq \sum_{t=1}^k \min(\alpha^{t-2}, \delta/2\tilde{L}^{t-1}) \geq \min(k\alpha^{-1}, k\delta/(2L^k)).$$

Finally, it is not difficult to check that $D(\mathcal{B}_T, \mathcal{B}_S) = 8R_0^2$. This completes the proof. $\square$

III. FROM GLOBAL TO LOCAL MIN-CONSENSUS

In this section we introduce a variant of Algorithm II-B wherein the global min-consensus step (S.3) is replaced by a local one—see Algorithm 3. Step (S.3) therein requires now only *local* communications with neighboring nodes. The update (S.3) produces now different local stepsizes, collected in $\Lambda^k := \text{diag}(\alpha_1^k, \alpha_2^k, \dots, \alpha_m^k)$. To simplify the notation, we defined $\text{prox}_{\Lambda^k, R}(\mathbf{X}) := [\text{prox}_{\alpha_1^k, R}(x_1), \dots, \text{prox}_{\alpha_m^k, R}(x_m)]^\top$.

Notice that Algorithm 3 differs from Algorithm 2 also in the updates of the $\mathbf{D}$ -variables, due to the fact that $\Lambda^k \notin \text{span}(I)$, for all k . In fact, in such a setting, one can show that the update of the $\mathbf{D}$ -variables should be instead

$$\mathbf{D}^{k+1} = \mathbf{D}^k + \mathbf{L}(\Lambda^k)^{-1} \mathbf{L}(\mathbf{X}^k - \Lambda^k \mathbf{S}^k - \Lambda^k \mathbf{D}^k - \Lambda^k \nabla F(\mathbf{X}^k)). \quad (38)$$

However, this update is not implementable on a network because $\mathbf{L}(\Lambda^k)^{-1} \mathbf{L}$ is not compliant with the graph $\mathcal{G}$. The proposed approach is then to “approximate” (38) by

$$\mathbf{D}^{k+1} = \mathbf{D}^k + \mathbf{L}^2(\Lambda^k)^{-1}(\mathbf{X}^k - \Lambda^k \mathbf{S}^k - \Lambda^k \mathbf{D}^k - \Lambda^k \nabla F(\mathbf{X}^k)). \quad (39)$$

Setting $\mathbf{L}^2 = I - W$, (39) is computable on the network. The update (39) serves as an effective approximation of (38) in the sense that it ensures two critical properties for convergence: **i)** $\mathbf{D}^k \in \text{span}(I - W)$, for all k ; and **ii)** if $\Lambda^k \in \text{span}(I)$, (39) recovers (38). A key property of the local-min consensus coupled with the proposed line-search procedure is to ensure that Λ^k will fall within $\text{span}(I)$ after a finite number of iterations.

Convergence of Algorithm III is summarized in Theorem 2, whose proof is omitted because of space limit—see [32]. Similarly to Theorems 1, it is convenient to state Theorem 2 referring to the equivalent formulation of Algorithm 3, using the updates in terms of $\mathbf{T}_{A,1}^k$ and $\mathbf{T}_{B,2}^k$, as given in (17) and (11), respectively, with the stepsize $\alpha^k I$ therein replaced by Λ^k , and updated according to (S.2)-(S.3) in Algorithm 3. In Theorem 2, the $D(\mathcal{B}_T, \mathcal{B}_S)$ is defined as in Theorem 1.

Algorithm 3 Decentralized Adaptive Three Operator Splitting with local min-consensus (local_DATOS)

Data: (i) initialization: $\alpha^{-1} \in (0, \infty)$, $\mathbf{X}^0 \in \mathbb{R}^{m \times n}$, $\mathbf{S}^0 \in \mathbb{R}^{m \times n}$ and $\mathbf{D}^0 = 0$; (ii) Backtracking parameters $\delta \in (0, 1]$; (iii) Gossip matrix $W = (1 - c)I + c\tilde{W}$, $c \in (0, 1/2)$.

1: (S.1) Communication Step:

$$\mathbf{X}^{k+1/2} = W\mathbf{X}^k, \quad \mathbf{D}^{k+1/2} = W(\nabla F(\mathbf{X}^k) + \mathbf{S}^k + \mathbf{D}^k);$$

2: (S.2) Decentralized line-search: Each agent updates α_i^k according to:

$$\bar{\alpha}_i^k = \text{Linesearch}(\alpha_i^{k-1}, f_i, x_i^k, x_i^{k+1/2}, -d_i^{k+1/2}, \gamma_k, \delta);$$

3: (S.3) Local min-consensus: Each agent updates α_i^k according to:

$$\alpha_i^k = \min_{j \in \mathcal{N}_i} \bar{\alpha}_j^k;$$

4: (S.4) Extra scalar communication step
Let $\Lambda^k := \text{diag}(\alpha_1^k, \alpha_2^k, \dots, \alpha_m^k)$,

$$\mathbf{D}_\Lambda^k = (I - W)(\Lambda^k)^{-1} \mathbf{X}^k;$$

5: (S.5) Updates of the auxiliary, dual and primal variables:

$$\begin{aligned} \mathbf{X}^{k+1} &= \text{prox}_{\Lambda^k, R}(\mathbf{X}^{k+1/2} - \Lambda^k \mathbf{D}^{k+1/2} + \Lambda^k \mathbf{S}^k) \\ \mathbf{S}^{k+1} &= \mathbf{S}^k + (\Lambda^k)^{-1}(\mathbf{X}^{k+1/2} - \mathbf{X}^{k+1} - \mathbf{D}^{k+1/2}) \\ \mathbf{D}^{k+1} &= \mathbf{D}^{k+1/2} + \mathbf{D}_\Lambda^k - \nabla F(\mathbf{X}^k) - \mathbf{S}^k. \end{aligned}$$

6: (S.6) If a termination criterion is not met, $k \leftarrow k + 1$ and go to step (S.1).

Theorem 2. Consider Algorithm III under Assumptions 1-3. Further assume that sequence $\{\mathbf{T}_A^k, \mathbf{S}^k\}_k$ is bounded. Then, for any given saddle point $(\mathbf{T}^*, \mathbf{S}^*)$ of $\mathcal{L}$, there exists a bounded set $\mathcal{B}_T \times \mathcal{B}_S$ such that (i) $(\mathbf{T}^*, \mathbf{S}^*) \in \mathcal{B}_T \times \mathcal{B}_S$, and $(\mathbf{T}_A^k, \mathbf{S}^k) \in \text{int}(\mathcal{B}_T \times \mathcal{B}_S)$, for all $k \geq 0$; and (ii) for some finite $K \geq 0$, it holds

$$\mathcal{G}_{\mathcal{B}_T \times \mathcal{B}_S}(\bar{\mathbf{T}}_A^k, \bar{\mathbf{S}}^k) \leq \frac{1}{k} \cdot \frac{C_K + 8R_K^2}{\min(\alpha^{-1}, \delta/(2L^k))}, \quad \forall k \geq K, \quad (40)$$

where $C_K := 2 \sum_{t=1}^K \alpha^{t-1} \mathcal{G}_{\mathcal{B}_T \times \mathcal{B}_S}(\mathbf{T}_A^t, \mathbf{S}^t)$ and

$$R_K^2 := \|\mathbf{T}_B^K - \mathbf{T}^*\|^2 + (\alpha^K)^2 \|\mathbf{S}^K - \mathbf{S}^*\|^2.$$

Theorem 2 certifies convergence of Algorithm II-B at sub-linear rate. The primary distinction between the local-min and global-min consensus procedures is that the local-min does not guarantee a monotonically decreasing merit function during the initial K iterations. However, as demonstrated by the numerical results in the following section, in practice, Algorithm 3 performs comparably to Algorithm 1.

Convergence of Algorithm 3 relies on the boundedness of the generated iterates. This is trivially ensured, for instance, when the loss functions are globally smooth or when r is the indicator function of a compact (convex) set. Additional scenarios and algorithmic variants that guarantee boundedness are discussed in the extended version of this paper [32].

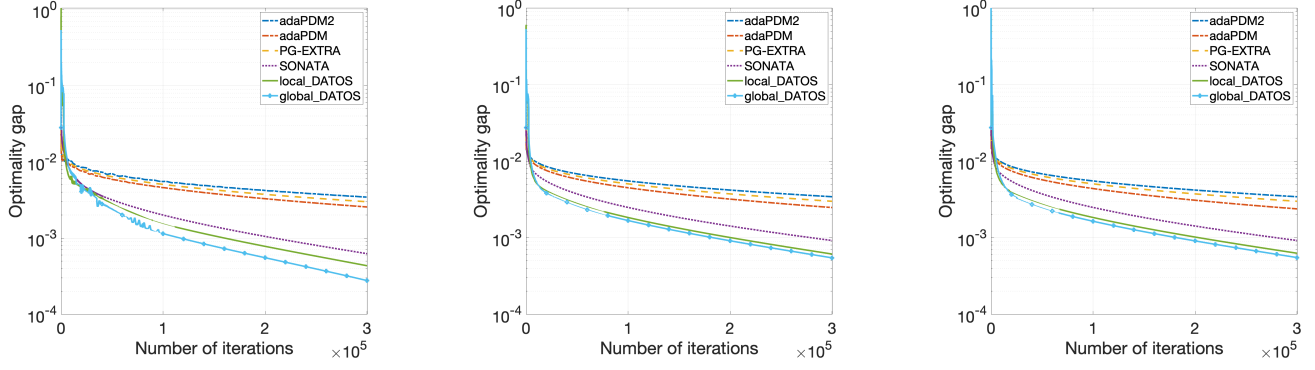


Fig. 1. Logistic regression with ℓ_1 -regularization: $\frac{1}{m} \sum_{i=1}^m u(x_i) - u(x^*)$ v.s. # iterations. Erdos-Renyi graphs with different edge-probability: $p = 0.1$ (left); $p = 0.5$ (middle); and $p = 0.9$ (right).

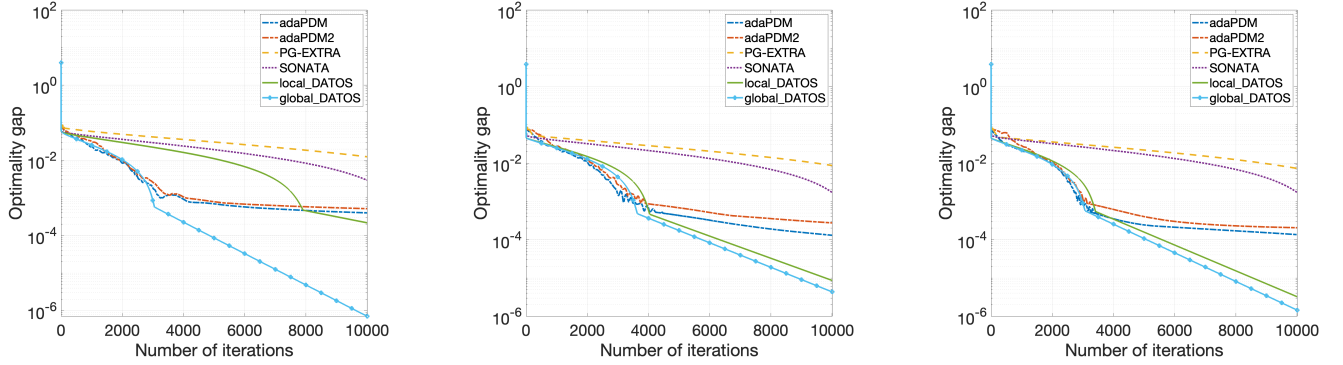


Fig. 2. Maximum likelihood covariance estimation: $\frac{1}{m} \sum_{i=1}^m u(x_i) - u(x^*)$ v.s. # iterations. Erdos-Renyi graphs with different edge-probability: $p = 0.1$ (left); $p = 0.5$ (middle); and $p = 0.9$ (right).

IV. NUMERICAL RESULTS

This section presents some preliminary numerical results, comparing Algorithms II-B and III against several benchmarks. Specifically, we consider the *nonadaptive* decentralized algorithms SONATA [33] and PG-EXTRA [30] as well as decentralized adaptations of the adaptive centralized method adaPDM [21]. SONATA and EXTRA require full knowledge of network and optimization parameters; for these methods, we perform a manual grid-search to identify the optimal stepsize ensuring fast convergence. Regarding adaPDM, we emphasize that it is not entirely parameter-free, since it requires the knowledge of the *global* network-related quantity $\|I - W\|$, unavailable in practice. For comparison, we simulate two versions of adaPDM: (i) one assuming exact knowledge of $\|I - W\| \leq 2$, termed adaPDM, and (ii) another using the conservative upper bound 2, resulting in a network-agnostic variant termed adaPDM2. In both cases, we manually tune the parameter t in adaPDM and adaPDM2 (as in [21]) to obtain the best possible practical convergence behavior. We anticipate that adaPDM and adaPDM2 are very sensitive to the choice of t .

We simulate Erdos-Renyi graphs, with $m = 20$ agents and edge-probability of $p = 0.1$, $p = 0.5$, and $p = 0.9$. The gossip weights used in all the algorithms are the

Metropolis-Hasting (see, e.g., [1]). For the implementation of Algorithm II-B and III, we initialize $\mathbf{X}^0$ and $\mathbf{S}^0$ randomly and set $\alpha^{-1} = 10$, $\delta = 0.9$, and $c = 1/3$.

A. Logistic regression with ℓ_1 -regularization

Consider the decentralized logistic regression problem with ℓ_1 -regularization, which is an instance of (P), with

$$f_i(x) = \frac{1}{n} \sum_{j=1}^n \log(1 + \exp(-b_{ij} \cdot \langle x, a_{ij} \rangle)), \quad r(x) = \lambda \|x\|_1,$$

where $a_{ij} \in \mathbb{R}^d$ and $b_{ij} \in \{-1, 1\}$. The data set $\{(a_{ij}, b_{ij})\}_{j=1}^n$ is owned by agent i . We use the MNIST dataset from LIBSVM [34], taking the first $N = 6000$ samples (hence $n = 300$). The feature dimension is $d = 784$. We set the ℓ_1 regularization parameter $\lambda = 10^{-5}$.

Figure 1 plots the optimality gap $(1/m) \sum_{i=1}^m u(x_i) - u^*$ versus the number of iterations, achieved by all the algorithms, where u is the objective function in (P) and u^* is its minimal value. The figures clearly show that both proposed methods consistently outperforms SONATA and PG-EXTRA, with the advantage that do not require any user's intervention for the tuning of the stepsize. Moreover, our methods significantly outperforms also adaPDM in both

of its implementations. Furthermore, the performance difference between Algorithm 3, which incorporates local-min consensus, and Algorithm 1—using global-min consensus, is remarkably negligible especially in networks with good connectivity.

B. Maximum likelihood estimate of the covariance matrix

Consider the decentralized estimation of the inverse of a covariance matrix, which is an instance of (P), with

$$f_i(X) = -n(\log(\det(X))) - \text{trace}(XY_i),$$

and $r(X) = \delta_C(X)$, where $C = \{X \in \mathbb{S}_{++}^d : aI \preceq X \preceq bI\}$, for some $0 < a \leq b$, and $Y_i = \frac{1}{n} \sum_{j=1}^n y_j^i (y_j^i)^\top$ for $\{y_j^i\}$ being the set of local samples at agent i 's site, drawn by a Gaussian distribution with covariance matrix $\Sigma \in \mathbb{S}_{++}^d$. Here we take $n = 100$ and $d = 5$. Notice that this is a problem where the loss function f is only locally smooth. Hence, SONATA and PG-EXTRA do not have theoretical convergence guarantees. We manually fine-tune their stepsizes for the best stable convergence behavior.

Figure 2 plots the optimality gap $(1/m) \sum_{i=1}^m u(x_i) - u^*$ versus the number of iterations, achieved by all the algorithms, where u is the objective function in (P) and u^* is its minimal value calculated by centralized proximal gradient method with linesearch, within the tolerance of 10^{-30} . The figures clearly illustrate that both proposed methods consistently outperform SONATA and PG-EXTRA, which require conservative stepsize selections to ensure stable global convergence. Additionally, our methods surpass adaPDM in both considered variants. Notice that the iterates generated by Algorithm 3 remain bounded, due to the compactness enforced by $r(X)$.

REFERENCES

- [1] A. Nedić, A. Olshevsky, and M. Rabbat, "Network topology and communication-computation tradeoffs in decentralized optimization," *Proceedings of the IEEE*, vol. 106, pp. 953–976, 2018.
- [2] A. H. Sayed, "Adaptation, learning, and optimization over networks," *Foundations and Trends in Machine Learning*, vol. 7, pp. 311–801, January 2014.
- [3] D. P. Bertsekas, *Nonlinear Programming*. Athena Scientific, third ed., 2016.
- [4] D. O'Connor and L. Vandenberghe, "On the equivalence of the primal-dual hybrid gradient method and douglas-rachford splitting," *Mathematical Programming*, vol. 179, no. 1, pp. 85–108, 2020.
- [5] B. Polyak, "Minimization of unsmooth functionals," *USSR Computational Mathematics and Mathematical Physics*, vol. 9, no. 3, pp. 14–29, 1969.
- [6] J. Barzilai and J. M. Borwein, "Two-point step size gradient methods," *IMA Journal of Numerical Analysis*, vol. 8, no. 1, pp. 141–148, 1988.
- [7] Y. Malitsky and K. Mishchenko, "Adaptive gradient descent without descent," in *International Conference on Machine Learning*, 2019.
- [8] Y. Malitsky and K. Mishchenko, "Adaptive proximal gradient method for convex optimization," *arXiv preprint arXiv:2308.02261*, 2024.
- [9] D. Zhou, S. Ma, and J. Yang, "Adabb: Adaptive barzilai-borwein method for convex optimization," *arXiv preprint arXiv:2401.08024*, 2024.
- [10] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," in *Proceedings of the 24th International Conference on Neural Information Processing Systems*, pp. 257–265, 2011.
- [11] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014.
- [12] S. J. Reddi, S. Kale, and S. Kumar, "On the convergence of adam and beyond," in *International Conference on Learning Representations (ICLR)*, 2018.
- [13] X. Li and F. Orabona, "On the convergence of stochastic gradient descent with adaptive stepsizes," in *The 22nd international conference on artificial intelligence and statistics*, pp. 983–992, PMLR, 2019.
- [14] R. Ward, X. Wu, and L. Bottou, "Adagrad stepsizes: Sharp convergence over nonconvex landscapes," *The Journal of Machine Learning Research*, vol. 21, pp. 1–30, 2020.
- [15] P. Nazari, D. Tarzanagh, and G. Michailidis, "Dadam: A consensus-based distributed adaptive gradient method for online optimization," *IEEE Transactions on Signal Processing*, vol. 70, p. 6065–6079, 2022.
- [16] X. Chen, B. Karimi, W. Zhao, and P. Li, "On the convergence of decentralized adaptive gradient methods," in *Asian Conference on Machine Learning*, pp. 217–232, PMLR, 2023.
- [17] J. Li, X. Chen, S. Ma, and M. Hong, "Problem-parameter-free decentralized nonconvex stochastic optimization," *arXiv preprint arXiv:2402.08821*, 2024.
- [18] I. Kuruzov, G. Scutari, and A. Gasnikov, "Achieving linear convergence with parameter-free algorithms in decentralized optimization," *Advances in Neural Information Processing Systems*, vol. 37, pp. 96011–96044, 2024.
- [19] R. Aldana-Lopez, A. Macchelli, G. Notarstefano, R. Aragues, and C. Sagues, "Towards parameter-free distributed optimization: a port-hamiltonian approach," *arXiv preprint arXiv:2404.13529*, 2024.
- [20] Y. Malitsky and T. Pock, "A first-order primal-dual algorithm with linesearch," *SIAM Journal on Optimization*, vol. 28, no. 1, pp. 411–432, 2018.
- [21] P. Latafat, A. Themelis, and P. Patrinos, "Adaptive proximal algorithms for convex optimization under local lipschitz continuity of the gradient," *Mathematical Programming, Series A*, 2024.
- [22] M. Bertero, P. Boccacci, G. Desiderà, and G. Vicidomini, "Image deblurring with poisson data: from cells to galaxies," *Inverse Problems*, vol. 25, no. 12, p. 123006, 2009.
- [23] S. Boyd, N. Parikh, and E. Chu, *Distributed optimization and statistical learning via the alternating direction method of multipliers*. Now Publishers Inc, 2011.
- [24] H. H. Bauschke, J. Bolte, and M. Teboulle, "A descent lemma beyond lipschitz gradient continuity: first-order methods revisited and applications," *Mathematics of Operations Research*, vol. 42, no. 2, pp. 330–348, 2017.
- [25] E. K. Ryu and W. Yin, *Large-scale convex optimization: algorithms & analyses via monotone operators*. Cambridge University Press, 2022.
- [26] D. Davis and W. Yin, "A three-operator splitting scheme and its optimization applications," *Set-valued and variational analysis*, vol. 25, pp. 829–858, 2017.
- [27] D. Kim, J. Lim, and J. Kim, "Low-power, long-range, high-data transmission using wi-fi and lora," in *2016 6th International Conference on IT Convergence and Security (ICITCS)*, (Prague, Czech Republic), p. 1–3, IEEE, Sept. 2016.
- [28] T. Janssen, N. BniLam, M. Aernouts, R. Berkvens, and M. Weyn, "Lora 2.4 ghz communication link and range," *Sensors*, vol. 20, p. 4366, Aug. 2020.
- [29] A. Chambolle and T. Pock, "A first-order primal-dual algorithm for convex problems with applications to imaging," *Journal of mathematical imaging and vision*, vol. 40, pp. 120–145, 2011.
- [30] W. Shi, Q. Ling, G. Wu, and W. Yin, "A proximal gradient algorithm for decentralized composite optimization," *IEEE Transactions on Signal Processing*, vol. 63, no. 22, pp. 6013–6023, 2015.
- [31] L. Guo, X. Shi, J. Cao, and Z. Wang, "Decentralized inexact proximal gradient method with network-independent stepsizes for convex composite optimization," *IEEE Transactions on Signal Processing*, vol. 71, pp. 786–801, 2023.
- [32] X. Chen, I. Kuruzov, G. Scutari, and A. Gasnikov, "Adaptive decentralized algorithms for composite convex optimization," in *preparation*, 2025.
- [33] Y. Sun, G. Scutari, and A. Daneshmand, "Distributed optimization based on gradient-tracking revisited: Enhancing convergence rate via surrogation," *SIAM J. on Optimization*, vol. 32, pp. 354–385, 2022.
- [34] C. Chih-Chung, "libsvm: a library for support vector machines," *acm transactions on intelligent systems and technology*, 2: 27: 1–27: 27, 2011, <http://www.csie.ntu.edu.tw/~cjlin/libsvm>, vol. 2, 1996.