Adaptive Stepsize Selection in Decentralized Convex Optimization

Ilya Kuruzov
Innopolis University
kuruzov.ia@phystech.edu.

Xiaokai Chen
Purdue University
chen4373@purdue.edu.

Gesualdo Scutari
Purdue University
gscutari@purdue.edu.

Alexander Gasnikov
Innopolis University
gasnikov@yandex.ru

May 10, 2025

Abstract

We study decentralized optimization where multiple agents minimize the average of their (strongly) convex, smooth losses over a communication graph. Convergence of the existing decentralized methods generally hinges on an apriori, proper selection of the stepsize. Choosing this value is notoriously delicate: (i) it demands global knowledge from all the agents of the graph's connectivity and every local smoothness/strong-convexity constants–information they rarely have; (ii) even with perfect information, the worst-case tuning forces an overly small stepsize, slowing convergence in practice; and (iii) large-scale trial-and-error tuning is prohibitive. This work introduces a decentralized algorithm that is fully adaptive in the choice of the agents' stepsizes, without any global information and using only neighbor-to-neighbor communications–agents need not even know whether the problem is strongly convex. The algorithm retains strong guarantees: it converges at *linear* rate when the losses are strongly convex and at *sublinear* rate otherwise, matching the best-known rates of (nonadaptive) parameter-dependent methods.

1 Introduction

Consider the following multiagent optimization problem:

$$\min_{x\in\mathbb{R}^d} \frac{1}{m}\sum_{i=1}^{m} f_i(x), \tag{P}$$

where each $f_i : \mathbb{R}^d \to \mathbb{R}$ is the loss function accessible only to agent i, assumed to be (strongly) convex and smooth (i.e., with Lipschitz continuous gradient). Agents communicate through a fixed, undirected, and connected mesh network, represented as a graph $\mathcal{G}$, and interactions occur exclusively among immediate neighbors without a central coordinating server.

Problem (P) arises in numerous fields including signal processing, multiagent systems, communications, and decentralized machine learning, where data are distributed across multiple locations.

The literature includes various decentralized methods for solving (P) on mesh networks; we refer readers to the overview papers [28, 4, 27, 39] and book [35] for comprehensive reviews. Typically, existing methods rely on conservative stepsize choices, whose values depend on global parameters such as the Lipschitz constants of agents'

functions and network spectral properties. Since this global information is inaccessible to the agents, manual tuning is often employed, leading to unpredictable, conservative, and problem-dependent performance. This suggests the following question:

Can one design decentralized algorithms that adaptively select agents' stepsizes using only neighbor-to-neighbor communications, without any knowledge of the problem structure (not even whether the local functions are convex or strongly convex) or the network topology?

Despite recent attempts to introduce adaptivity into decentralized methods [26, 5, 19, 40, 17, 7, 1, 12], fully addressing this challenge remains an open problem, as documented next.

1.1 Related works

Adaptive centralized methods: Recent years have seen increased interest in adaptive optimization algorithms for centralized and federated setups. These methods include established techniques such as line-search approaches [30], Polyak's stepsize [31], Barzilai-Borwein's stepsize [2], as well as recent advancements that estimate the local curvature of the cost function [24, 25, 18, 40]. Additional adaptive gradient methods tailored for machine learning include AdaGrad [9], Adam [16], AMSGrad [34], NSGD-M [8], and their variants [21, 38]. Some methods have been adapted to federated architectures [33, 20, 6], but remain unsuitable for mesh networks due to reliance on central servers.

Adaptive decentralized methods: The landscape of adaptive *decentralized* methods remains limited [26, 5, 19, 40, 17, 7, 1, 12]. Approaches like [26, 5, 19] address stochastic, online, and smooth problems, typically employing gradient normalization. These methods generally assume global Lipschitz continuity–hence, they are not applicable to strongly convex instances of (P))–simplifying parameter-free convergence via standard stepsizes like $\mathcal{O}(1/\sqrt{k})$, and often still rely on some problem-specific parameter knowledge. The authors in [1] proposes a Port-Hamiltonian framework ensuring parameter-free convergence in *centralized* settings. Convergence (global asymptotic stability) of decentralized implementations is ensured for specific graph structures (e.g., cycle or fully connected graphs) or under *graph-dependent* stepsize constraints, calling for local knowledge of some network-dependent parameters. No explicit convergence rate expression is provided.

The works [10, 13, 11] adapt the Barzilai-Borwein (BB) stepsize to gradient tracking algorithms [37, 23, 29]. These methods have limited convergence guarantees: (i) agents' losses must be quadratic, as the BB rule may otherwise fail to converge *even in the centralized setting*; (ii) the BB-generated stepsizes must remain *uniformly bounded*, a condition that generally is not satisfied or hard to check; (iii) approaches by [11, 13] necessitate multiple communication rounds per iteration and global knowledge of network and optimization parameters, making them non-adaptive. More recently, [7, 12] developed an adaptive stepsize procedure for gradient-tracking algorithms, inspired by the curvature estimation approach in [24] for centralized methods. The method in [7] converges only to a neighborhood of the optimum and is numerically unstable on sparse networks [12]. These limitations were addressed in [12], which ensures exact convergence provided the adaptive stepsize remains below a given threshold depending on network and optimization parameters. Since this threshold is not known to the agents, the method is not practically implementable in decentralized settings where global information is unavailable. Additionally, both methods [7, 12] lacks a convergence rate analysis, leaving their theoretical competitiveness uncertain.

Recently, [17] proposed an adaptive decentralized method for strongly convex instances of (P), achieving linear convergence and demonstrating improved theoretical and practical performance over existing non-adaptive algorithms. This method employs a backtracking procedure at each agent to adaptively select stepsizes *locally*, followed by a *global* min-consensus step synchronizing agents' local stepsizes. This global consensus is implemented using flooding protocols based on low-power wide-area network (LPWAN) technologies such as LoRa (low power, long-range) [15, 14]. Alternatively, a *local* min-consensus rule is proposed, which exchanges information only with *immediate*

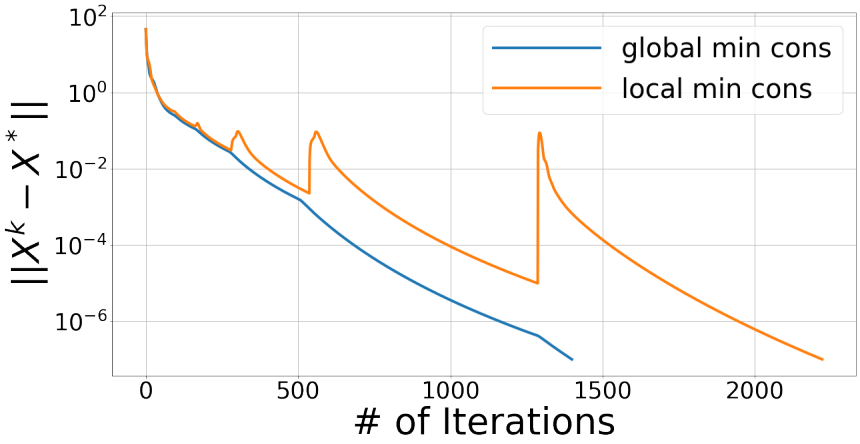


Figure 1: Algorithm from [17] using local and global min-consensus, applied to ridge regression.

neighbors. While practically more feasible, this local approach results in *weaker* theoretical guarantees, requiring stronger assumptions like bounded iterates, and yields *non-monotonic* convergence trajectories. Consequently, decentralized termination criteria become hard to implement, and error trajectories frequently show transient fluctuations or "spikes"–see Fig. 1.

1.2 Major contributions

The literature review above shows that existing adaptive decentralized algorithms cannot tackle the question raised in this paper. We fill this gap. Our major contributions are as follows.

Algorithm design: We propose a *fully* adaptive decentralized algorithm that operates using only neighbor-to-neighbor communication. Each agent tunes its primal and dual stepsizes via a local backtracking rule and then synchronizes them with its immediate neighbors through a lightweight local-minimum consensus that passes only *scalar* (hence negligible) messages. The procedure requires *zero* prior knowledge of optimization parameters (e.g., smoothness or strong-convexity constants) or network characteristics (degree, diameter, spectral gap). Agents need not even know whether their cost function is merely convex or strongly convex. Consequently, the algorithm runs out-of-the-box–without any manual tuning or user intervention.

On the technical side, our method departs substantially from [17]. That algorithm adapts stepsizes with a *primal-only* backtracking rule followed by a *global* (network-wide) min-consensus, yet it lacks any mechanism to tame the dual-space imbalances that heterogeneous primal stepsizes create. We address this shortcoming by introducing explicit *dual* stepsize variables and a *joint primal–dual tracking scheme* that continuously re-aligns the two sets of stepsizes, compensating in real time for imbalances produced by independently chosen primal stepsizes, yet preserving adaptivity. This architecture eliminates the costly flooding step required in [17] while preserving its convergence guarantees using only neighbor-to-neighbor communication. A further dividend of our design is a novel *running-diameter tracker* that estimates the "effective" graph diameter on the fly. Updated via a scalar local-min consensus, it delivers the synchronizing power of a diameter-based global consensus practically at a fraction of the communication overhead.

Convergence analysis: For strongly convex f_i's, the algorithm converges linearly, matching the best nonadaptive rates. Crucially, the smoothness and strong-convexity constants appearing in the bound are evaluated only over the *convex hull* of the iterates–a direct benefit of the stepsize adaptivity–so the dependence on problem parameters is far milder than in worst-case global bounds in the literature. For merely convex losses, we guarantee sublinear

convergence. To our knowledge, this is the first adaptive decentralized method with proven guarantees in the purely convex setting.

1.3 Notation

Throughout the paper we use the following notation. Capital letters denote matrices. Bold capital letters represent matrices where each row is an agent's variable, e.g., $\mathbf{X} = [x_1, \dots, x_m]^\top$. For such matrices, the i-th row is denoted by the corresponding lowercase letter with the subscript i; e.g., for $\mathbf{X}$, we write x_i (as column vector). Let $\mathbb{S}^m$, $\mathbb{S}^m_+$, and $\mathbb{S}^m_{++}$ be the set of $m \times m$ (real) symmetric, symmetric positive semidefinite, and symmetric positive definite matrices, respectively; $A^\dagger$ denotes the Moore-Penrose pseudoinverse of A. The eigenvalues of $W \in \mathbb{S}^m$ are ordered in nonincreasing order, and denoted by $\lambda_1(W) \geq \cdots \geq \lambda_m(W)$. We denote: $[m] = \{1, \dots, m\}$, for any integer $m \geq 1$; $[x]_+ := \max(x, 0)$, $x \in \mathbb{R}$; $1_m \in \mathbb{R}^m$ is the vector of all ones; I_m (resp. 0_m) is the $m \times m$ identity (resp. the $m \times m$ zero) matrix; the information on the dimension is omitted when not necessary; $\mathtt{null}(A)$ (resp. $\mathtt{span}(A)$) is the nullspace (resp. range space) of the matrix A. Let $\langle X, Y \rangle := \mathtt{tr}(X^\top Y)$, for any X and Y of suitable size ($\mathtt{tr}(\bullet)$) is the trace operator; and $\|X\|_M := \sqrt{\langle MX, X \rangle}$, for any symmetric, positive definite M and X of suitable dimensions. We still use $\|X\|_M$ when M is positive semidefinite and $X \in \mathtt{span}(M)$.

2 Setup and Preliminaries

We study Problem (P) under the following assumptions.

Assumption 1. **(i)** *Each function f_i in (P) is L-smooth and μ-strong convex on $\mathbb{R}^d$, for some $L \in (0, \infty)$ and $\mu \in [0, \infty)$;* **(ii)** *each agent i has access only to its own function f_i, but μ and L; and* **(iii)** *when $\mu = 0$, Problem (P) has a nonempty solution set.*

Notice that when $\mu = 0$, we simply require convexity. We remark that for simplicity we assumed global smoothness (and strong convexity), however our results extend to functions that are only locally smooth (and strongly convex).

Assumption 2. *Agents are embedded in a communication network modeled as an undirected, static, connected graph $\mathcal{G} = ([m], \mathcal{E})$, where $\mathcal{V} = [m]$ is the set of agents and $(i, j) \in \mathcal{E}$ if there is communication link (edge) between i and j. Let $\mathcal{N}_i := \{j : | (i, j) \in \mathcal{E}$, for some $i \in [m]\} \cup \{i\}$ denote the set of immediate neighbors of agent i (including agent i itself).*

2.1 The decentralized adaptive method [17]

For strongly convex instances of (P) ($\mu > 0$), [17] introduced the first *adaptive* decentralized algorithm that achieves provable convergence without requiring agents to know optimization or network parameters. Each agents keeps a local replica $x_i \in \mathbb{R}^d$ of the global variable $x \in \mathbb{R}^d$ along with dual variables $y_i \in \mathbb{R}^d$. Let us stack the primal and dual variables into $\mathbf{X} := [x_1, \dots, x_m]^\top \in \mathbb{R}^{m \times d}$ and $\mathbf{Y} := [y_1, \dots, y_m]^\top \in \mathbb{R}^{m \times d}$; let $\nabla F(\mathbf{X}) := [\nabla f_1(x_1), \cdots, \nabla f_m(x_m)]^\top$; and define the diagonal matrices of the agents' stepsizes $\Theta^k = \mathrm{diag}(\theta_1^k, \dots, \theta_m^k)$. The algorithm updates $\mathbf{X}^k$, $\mathbf{Y}^k$ and θ_i^k as follows: starting from $(\mathbf{X}^0, \mathbf{Y}^0) \in \mathbb{R}^{m \times d}$ and $\Theta^0 > 0$, for any $k \geq 0$,

$$\begin{aligned} \mathbf{X}^{k+1/2} &= W\,\mathbf{X}^k, \quad \mathbf{Y}^{k+1/2} = W\left(\mathbf{Y}^k + \nabla F(\mathbf{X}^{k+1/2})\right), \\ \mathbf{X}^{k+1} &= \mathbf{X}^{k+1/2} - \Theta^k \cdot \mathbf{Y}^{k+1/2}, \\ \mathbf{Y}^{k+1} &= \mathbf{Y}^{k+1/2} + (I - W)\left(\Theta^k\right)^{-1}\mathbf{X}^k - \nabla F(\mathbf{X}^{k+1/2}). \end{aligned} \tag{1}$$

The stepsizes are updated via backtracking followed by one of the two min-consensus schemes:

$$\theta_i^k = \begin{cases} \min\limits_{j\in[m]} \overline{\theta}_j^k & (\texttt{global min-consensus}); \\ \min\limits_{j\in\mathcal{N}_i} \overline{\theta}_j^k & (\texttt{local min-consensus}); \end{cases} \tag{2}$$

with

$$\overline{\theta}_j^k = \texttt{Backtracking}\left(\theta^{k-1}, f_i, x_i^{k+1/2}, -y_i^{k+1/2}, \gamma^{k-1}, \delta\right),$$

and the backtracking procedure detailed below, wherein $\delta > 0$ and $\gamma > 1$ are given constants.

Backtracking $(\theta,\ f,\ x,\ y,\ \gamma,\ \delta)$

$\theta^+ := \gamma\theta$; $x^+ := x + \theta^+ y$; and set $t = 1$;
while $f(x^+) > f(x) + \langle \nabla f(x), x^+ - x\rangle + \frac{\delta}{2\theta^+}\|x^+ - x\|^2$ **do**
  $\theta^+ \leftarrow (1/2)\theta^+$ and $x^+ := x + \theta^+ y$; set $t \leftarrow t + 1$;
return θ^+.

In (1), $W \in \mathbb{R}^{m\times m}$ represents the gossip matrix, satisfying the following standard property in the decentralized optimization literature [28, 35, 27]: $W \in \mathcal{W}_{\mathcal{G}}$, where $\mathcal{W}_{\mathcal{G}}$ is defined as follows.

Definition 3 (Gossip matrices)**.** *Let $\mathcal{W}_{\mathcal{G}}$ denote the set of matrices $\widetilde{W} = [\widetilde{W}_{ij}]_{i,j=1}^m$ that satisfy the following properties:* **(i)** *(compliance with $\mathcal{G}$) $\widetilde{W}_{ij} > 0$ if $(i,j) \in \mathcal{E}$; otherwise $\widetilde{W}_{ij} = 0$. Furthermore, $\widetilde{W}_{ii} > 0$, for all $i \in [m]$; and* **(ii)** *(doubly stochastic) $\widetilde{W} \in \mathbb{S}^m$ and $\widetilde{W} 1_m = 1_m$.*

Algorithm (1) operates as follows. Each agent: (i) performs two rounds of communication to compute intermediate variables $\mathbf{X}^{k+1/2}$ and $\mathbf{Y}^{k+1/2}$ via gossiping; (ii) applies local backtracking on its function f_i at $x_i^{k+1/2}$ along $-y_i^{k+1/2}$ to compute $\overline{\theta}_i^k$; and (iii) executes a min-consensus protocol to establish the final stepsize θ_i^{k+1}. In the *global* min-consensus mode, agents execute a *network-wide* min-operation and adopt the common stepsize value $\min_{j\in[m]} \overline{\theta}_j^k$. In contrast, the *local* min-consensus rule $\min_{j\in\mathcal{N}_i} \overline{\theta}_j^k$ exchanges information only with immediate neighbors, but resulting in much weaker convergence guarantees–see discussion in Sec. 1.1 and Fig. 1.

The next section addresses these shortcomings, devising an adaptive scheme relying solely on one-hop communications yet preserving convergence guarantees of Algorithm (1) with global min-consensus.

3 A New Fully-Decentralized Design

Using (1) equipped with the global min-consensus as a benchmark, our first step is to understand the reasons behind the performance gap between its local and global min-consensus variants.

We compare the one-step updates of Algorithm (1) from an arbitrary iterate $(\mathbf{X}^k, \mathbf{Y}^k)$ with $\mathbf{Y}^k \in \texttt{range}(I - W)$, $(\mathbf{X}^k, \mathbf{Y}^k)$, using two distinct step-size strategies. The first one, denoted $(\mathbf{X}_{\texttt{lc}}^{k+1}, \mathbf{Y}_{\texttt{lc}}^{k+1})$, uses *heterogeneous* step sizes $\Theta^k = \text{diag}(\theta_1^k \dots \theta_m^k) > 0$, with $\theta_{\min}^k (:= \min_{i\in[m]} \theta_i^k) \neq \theta_{\max}^k (:= \max_{i\in[m]} \theta_i^k)$. The second instance, denoted $(\mathbf{X}_{\texttt{gb}}^{k+1}, \mathbf{Y}_{\texttt{gb}}^{k+1})$, uses *uniform* step sizes $\Theta^k = \theta_{\min}^k I$. The constructions $(\mathbf{X}_{\texttt{lc}}^{k+1}, \mathbf{Y}_{\texttt{lc}}^{k+1})$ and $(\mathbf{X}_{\texttt{gb}}^{k+1}, \mathbf{Y}_{\texttt{gb}}^{k+1})$ capture precisely the updates in Algorithm (1) using the local-and global-min consensus strategies (2), respectively. Let $(\mathbf{X}^\star, \mathbf{Y}^\star)$ denote an arbitrary fixed point of the algorithm, using global min-consensus; it must be $\mathbf{X}^\star = 1(x^\star)^\top$, with $x^\star$ being an optimal solution of (P) [17], and $1 \in \mathbb{R}^m$ is the vector of all ones.

Lemma 4. *In the setting above and Assumptions 1-2, the following holds:*

$$\|\mathbf{X}_{\texttt{gb}}^{k+1} - \mathbf{X}_{\texttt{lc}}^{k+1}\| + \|\mathbf{Y}_{\texttt{gb}}^{k+1} - \mathbf{Y}_{\texttt{lc}}^{k+1}\| = \mathcal{O}\left(\left(\theta_{\max}^k - \theta_{\min}^k\right) \max\left(\|\mathbf{X}^\star\|, R^k\right)\right), \tag{3}$$

where $R^k := \|\mathbf{X}^k - \mathbf{X}^\star\| + \|\mathbf{Y}^k - \mathbf{Y}^\star\|$. *Furthermore, if* $\|\mathbf{X}^\star\| \geq 2m^{1.5}R^k$, *then*

$$\|\mathbf{Y}_{\texttt{gb}}^{k+1} - \mathbf{Y}_{\texttt{lc}}^{k+1}\| = \Omega\left(\left(\theta_{\max}^k - \theta_{\min}^k\right)\max\left(\|\mathbf{X}^\star\|, R^k\right)\right). \tag{4}$$

Eq. (4) indicates that whenever $\theta_{\min}^k \neq \theta_{\max}^k$–the typical scenario when employing the local min-consensus–the dual trajectory $\mathbf{Y}_{\texttt{lc}}^{k+1}$ can deviate from $\mathbf{Y}_{\texttt{gb}}^{k+1}$, even if $R^k = 0$, i.e., when the algorithm has reached the optimal solution at iteration k. Enforcing $\theta_{\min}^k = \theta_{\max}^k$ for every k using only neighbor-to-neighbor communications is generally infeasible and would, in any case, significantly reduce the algorithm's adaptivity. This issue arises because of the presence of the term $\|\mathbf{X}^\star\|$ in the max expression in (3)-(4), which persists even if R^k approaches zero (or is sufficiently small).

To overcome this limitation and preserve stepsize adaptivity, we propose using *separate* stepsizes in the dual variable updates in (1), denoted by $\Pi^k = \mathrm{diag}(\pi_1^k, \ldots, \pi_m^k)$. The modification of (1) reads:

$$\begin{aligned}
\mathbf{X}^{k+1/2} &= W\,\mathbf{X}^k, \quad \mathbf{Y}^{k+1/2} = W\left(\mathbf{Y}^k + \nabla F(\mathbf{X}^{k+1/2})\right),\\
\mathbf{X}^{k+1} &= \mathbf{X}^{k+1/2} - \Theta^k \cdot \mathbf{Y}^{k+1/2},\\
\mathbf{Y}^{k+1} &= \mathbf{Y}^{k+1/2} + (I - W)\left(\Pi^k\right)^{-1}\mathbf{X}^k - \nabla F(\mathbf{X}^{k+1/2}).
\end{aligned} \tag{5}$$

We proceed designing neighbor-only primal/dual stepsize updates.

We begin deriving the counterpart of Lemma 4. Starting from $(\mathbf{X}^k, \mathbf{Y}^k)$, compare one step of (5) with *heterogeneous* stepsizes Θ^k and Π^k to Algorithm (1) using the uniform stepsize $\Theta^k = \theta_{\min}^k I$, yielding iterates $(\mathbf{X}_\pi^{k+1}, \mathbf{Y}_\pi^{k+1})$ and $(\mathbf{X}_{\texttt{gb}}^{k+1}, \mathbf{Y}_{\texttt{gb}}^{k+1})$, respectively. We have the following, where $(\mathbf{X}^\star, \mathbf{Y}^\star)$ denotes an arbitrary fixed point of Algorithm (1) using uniform stepsizes.

Lemma 5. *In the setting above and Assumptions 1-2, the following holds:*

$$\Delta^{k+1} = \mathcal{O}\left(\left(\left|\theta_{\min}^k - \pi_{\min}^k\right| + \left(\theta_{\max}^k - \theta_{\min}^k\right)\right)R^k + \left(\pi_{\max}^k - \pi_{\min}^k\right)\max\left(\|\mathbf{X}^\star\|, R^k\right)\right), \tag{6}$$

where $\Delta^{k+1} := \|\mathbf{X}_{\texttt{gb}}^{k+1} - \mathbf{X}_\pi^{k+1}\| + \|\mathbf{Y}_{\texttt{gb}}^{k+1} - \mathbf{Y}_\pi^{k+1}\|$, $\pi_{\min}^k := \min_i \pi_i^k$, *and* $\pi_{\max}^k := \max_i \pi_i^k$.

Eq. (6) highlights that the discrepancy Δ^{k+1} is now controllable even when $\theta_{\min}^k \neq \theta_{\max}^k$, provided $\pi_{\min}^k \approx \pi_{\max}^k$. This suggests the following desideratum for the updates of Θ^k and Π^k:

(c1) Maintain $\Pi^k \approx \pi^k I$ for almost all iterations, and some suitably adjusted scalar π^k; and

(c2) Ensure that $|\pi_{\min}^k - \theta_{\min}^k|$ and $\theta_{\max}^k - \theta_{\min}^k$ remain uniformly bounded, possibly sufficiently small across all iterations k.

(c1) guarantees that the second term in (6) remains negligible while **(c2)** promotes adherence of iterates (5) to those of the benchmark (1) that uses global min-consensus.

Based on the above, we propose the following updating procedures for the primal and dual stepsizes.

Primal stepsize updates: Given θ_i^k, each agent i tunes $\overline{\theta}_i^k$ via the backtracking procedure (Sec. 2.1), then runs the *local* min-consensus step (2) on $\overline{\theta}_i^k$, producing θ_i^{k+1}. This method adapts stepsizes to local curvatures of the losses f_i, still controlling the discrepancy $\theta_{\max}^k - \theta_{\min}^k$.

Dual stepsize updates: Ideally, $\Pi^k = \theta_{\min}^k I$, for all k, would fulfill **(c1)** and **(c2)**, but $\theta_{\min}^k$ is *unknown* locally. To address this, we design a neighbor-only protocol tracking $\theta_{\min}^k$. We hinge on the following properties of the backtracking search, which hold irrespectively of the algorithm trajectory.

Lemma 6. *Given an L-smooth function* f, *arbitrary sequences* $\{x^k, y^k\}_k \subseteq \mathbb{R}^d \times \mathbb{R}^d$ *and* $\{\gamma^k\}_k \subset \mathbb{R}_+$, *and* $\theta^0 > 0$ *and* $\delta > 0$, *let* $\theta^{k+1} = \texttt{Backtracking}(\theta^k, f, x^k, y^k, \gamma^k, \delta)$, $k \in \mathbb{N}_0$. *The following hold, for every* k: *(a) either*

$\theta^{k+1} < \theta^k$ *or* $\theta^{k+1} = \gamma^k \theta^k$*; and (b) the number of decreases of stepsize value,* $\mathcal{I}_\theta(k) = \{j \leq k \,|\, \theta^{j+1} < \theta^j\}$*, is bounded by* $|\mathcal{I}_\theta(k)| = \mathcal{O}(\sum_{j=1}^k \log \gamma^j)$.

Initially, assume each agent knows the graph diameter $d_\mathcal{G}$ and receives $\theta^k_{\min}$ precisely every $d_\mathcal{G}$ iterations, that is, at $k \equiv 0 \mod d_\mathcal{G}$. The horizon of $d_\mathcal{G}$ is because the min-consensus protocol converges in at most $d_\mathcal{G}$ steps. In the intermediate iterations $k' = k+1, \ldots, k+d_\mathcal{G}-1$, if $\{\theta^{k'}_{\min}\}$ does not decrease, by Lemma 6(a), each agent can compute $\theta^{k'}_{\min}$ directly without communicating, by $\theta^{k'}_{\min} = \gamma^{k'-1}\theta^{k'-1}_{\min}$. This suggests the following update for Π^k:

$$\Pi^k = \theta^k_{\min} I, \;\; \forall k \equiv 0 \mod d_\mathcal{G}, \quad \text{and} \quad \Pi^{k'} = \gamma^{k'-1}\Pi^{k'-1}, \;\; \forall k' = k+1, \ldots, k+d_G-1. \tag{7}$$

This ensures **(c1)** always, and **(c2)** except for the iterations where $\theta^{k'}_{\min}$ decreases; Lemma 6(b) shows that such drops are rare–at most $\mathcal{O}(\sum_{j=1}^k \log \gamma^j)$ within $\{0, \ldots, k\}$, for any $k \in \mathbb{N}_0$. We will choose $\{\gamma^k\}$ exactly to minimize such occurrences, guided by the convergence analysis (Sec. 3).

The rule (7) subsumes that every agent obtains $\theta^k_{\min}$, which would require a *global* min-consensus protocol. To avoid that, we propose tracking $\theta^k_{\min}$ introducing *local surrogates* collected in $\widetilde{\Theta}^k = \operatorname{diag}(\widetilde{\theta}^k_1, \ldots, \widetilde{\theta}^k_m)$. Each agent i updates $\widetilde{\theta}^k_i$ performing a *local* min-consensus step: $\forall k \in \mathbb{N}_0$, let

$$\widetilde{\theta}^k_i = \min_{j \in \mathcal{N}_i} \theta^k_j, \text{ if } k \equiv 1 \mod d_\mathcal{G}; \quad \text{and} \quad \widetilde{\theta}^{k'}_i = \min_{j \in \mathcal{N}_i} \gamma^{k'-1}\widetilde{\theta}^{k'-1}_j, \text{ if } k' = k+1, \ldots, k+d_G-1. \tag{8}$$

Every $d_\mathcal{G}$ iterations, at $k \equiv 1 \mod d_\mathcal{G}$, the min-consensus step (8) is "reset" with the current primal stepsize value θ^k_i, for each agent i. The above procedure has the following properties. **(i)** *Consensus*: after at most $d_\mathcal{G}$ hops, a consensus is reached, $\widetilde{\theta}^{k+d_\mathcal{G}-1}_i = \widetilde{\theta}^{k+d_\mathcal{G}-1}_j$, for all $i, j \in [m]$, $i \neq j$, and $k \in \mathbb{N}_0$. **(ii)** *Recovery of* $\theta^k_{\min}$*:* If $\{\theta^{k'}_{\min}\}$ does not decrease over $k' = k+1, \ldots, k+d_\mathcal{G}-1$, it must be $\widetilde{\theta}^{k+d_\mathcal{G}-1}_i = \theta^{k+d_\mathcal{G}-1}_{\min}$, for all $i \in [m]$–hence agents recover $\theta^{k+d_\mathcal{G}-1}_{\min}$ *locally*. If $\{\theta^{k'}_{\min}\}$ drops, $\widetilde{\theta}^k_i$ will be still consensual at $k+d_\mathcal{G}-1$, approximating $\theta^{k+d_\mathcal{G}-1}_{\min}$.

Using (8), we replace (7) with the following:

$$\Pi^k = \widetilde{\Theta}^k, \quad k \equiv 0 \mod d_\mathcal{G} \quad \text{and} \quad \Pi^{k'} = \gamma^{k'-1}\Pi^{k'-1}, \quad \forall k' = k+1, \ldots, k+d_G-1. \tag{9}$$

Since $\widetilde{\Theta}^k$ is consensual at $k+d_\mathcal{G}-1$, the re-initializations of Π^k at $k \equiv 0 \mod d_\mathcal{G}$ in (9) yield $\Pi^k = \pi^k I$, that is *consensual* dual stepsizes. If $\{\theta^{k'}_{\min}\}$ does not decrease in the previous $d_\mathcal{G}-1$ iterations (up to k), by the recovery property of (8), it follows $\pi^k = \theta^k_{\min}$, matching the ideal rule (7) *without requiring global min-consensus.*

Adaptive estimation of the "effective" diameter: Our rules (8)-(9) need a horizon long enough for the local-min consensus to converge. When the *true* graph diameter $d_\mathcal{G}$ is unknown, we introduce a fully-decentralized procedure whereby each agent i keeps an estimate d^k_i that replaces $d_\mathcal{G}$ in (8)-(9), and enlarges it only when strictly necessary, eventually converging to an *effective diameter*–the smallest integer that still guarantees (8) and (9) to work exactly as intended. Let the auxiliary variables $\widetilde{\theta}^k_i$ evolve as in (8), but with $d_\mathcal{G}$ replaced by the current guess d^k_i; the d^k_i's are updated for all $k \in \mathbb{N}_0$ and $i \in [m]$, as

$$d^{k+1}_i = \max_{j \in \mathcal{N}_i}(2d^k_j), \;\; \text{if } \; k \equiv 0 \mod d^k_i \quad \text{and} \quad \widetilde{\theta}^k_i \neq \min_{j \in \mathcal{N}_i} \widetilde{\theta}^k_j; \quad \text{otherwise} \quad d^{k+1}_i = \max_{j \in \mathcal{N}_i} d^k_j. \tag{10}$$

In words, at every iteration that is a multiple of the current horizon, $k \equiv 0 \mod d^k_i$, agent i checks whether the local-min consensus has already converged, $\widetilde{\theta}^k_i \stackrel{?}{=} \min_{j \in \mathcal{N}_i} \widetilde{\theta}^k_j$. If the test fails, the diameter estimate is doubled. This value then feeds a one-step max-consensus for synchronization with the other agents'. **Why the test is sound:** Take two consecutive multiples of the current horizon, k' and $k''(> k')$, so that $k' \equiv 0 \mod d^{k'}_i$ and $k'' \equiv 0 \mod d^{k''}_i$; it is $k'' \geq k' + d^{k'}_i$. According to (8), no re-initialization of the $\widetilde{\theta}$-variables occurs in $[k'+1, k'']$. Hence, if the local consensus test fails at k'' then at least $d^{k'}_i$ rounds of local min-consensus proved insufficient, and doubling is required. **Guarantees: (i)** *Finite number of failures*–A consensual value $d^k_i = d_\mathcal{G}$, for all i, always passes the test;

specifically, one can show that each agent can fail it at most $\sim \log_2 d_{\mathcal{G}}$. **(ii)** *Horizon may be smaller than the true diameter*–The local–min consensus in (8) can converge on a common horizon $d_i^k = d_j^k$, $i \neq j \in [m]$, that is strictly smaller than $d_{\mathcal{G}}$. This facilitates condition **(c2)**, as it reduces the frequency of iterations where primal and dual step sizes differ, thus bringing the proposed fully-decentralized scheme closer to Algorithm (1) employing global-consensus communications.

The overall algorithm incorporating the primal-and dual-step procedures (8) and (9) as well as the adaptive estimation scheme in (10) of the effective diameter is summarized in Algorithm 1.

The simple example below highlights why an adaptive horizon can outperform the conservative choice $d_{\mathcal{G}}$.

Remark 7. *Consider a line graph with m agents, so $d_{\mathcal{G}} = m - 1$. Assume the cost functions alternate along the line: $f_{2n+1} \equiv f$ and $f_{2n} \equiv g$, for $n = 1 \dots \lfloor m/2 \rfloor$; where $f, g : \mathbb{R}^d \to \mathbb{R}$ are some convex and smooth functions. Additionally, suppose the backtracking yields identical stepsizes for equal functions, i.e., $\overline{\theta}_{2n+1}^k \equiv \theta_f^k$ and $\overline{\theta}_{2n}^k \equiv \theta_g^k$, for all k. This is the case, e.g., for $f(x) = x^2$ and $g(x) = ax^2, a > 1$. One round of local min-consensus then is enough to synchronize all agents: $\theta_i^k = \min(\theta_f^k, \theta_g^k)$, for all $i \in [m]$. Hence, all min-consensus procedures in Algorithm 1 converge in one round. This matches the behavior of Algorithm (1) with global min-consensus! In contrast, if every agent uses $d_i^k = d_{\mathcal{G}}$, it would introduce a $m-1$-fold delay, potentially slowing convergence.*

Algorithm 1

Data: (i) Initialization: $\mathbf{X}^0 \in \mathbb{R}^{m \times n}$ and $\mathbf{Y}^0 = 0$; $\theta_i^{-1} \in (0, \infty)$ (primal-stepsize), $d_i^0 := d^0$ for some $d^0 \in \mathbb{N}$ (local diameter estimate); $\pi_i^{-1}, \widetilde{\theta}_i^{-1} \in (0, \infty)$ (dual stepsizes and auxiliary variables). (ii) Backtracking parameters: $\delta \in (0, 1]$, and $\{\gamma^k\}_k \subseteq [1, \infty)$. (iv) Gossip matrix $W := (1-c)I_m + c\widetilde{W}$, with $\widetilde{W} \in \mathcal{W}_{\mathcal{G}}$, and $c \in (0, 1/2]$. Set the iteration index $k = 0$.

`(S.1) Communication step:` Agents updates primal and dual variables via gossiping:

$$\mathbf{X}^{k+1/2} = W\,\mathbf{X}^k \quad \text{and} \quad \mathbf{Y}^{k+1/2} = W\,\left(\mathbf{Y}^k + \nabla F(\mathbf{X}^{k+1/2})\right);$$

`(S.2) Update of the primal stepsizes`: Each agent $i \in [m]$ updates their primal step sizes using a local backtracking procedure followed by a min-consensus step:

$$\overline{\theta}_i^k = \texttt{Backtracking}\left(\theta_i^{k-1}, f_i, x_i^{k+1/2}, y_i^{k+1/2}, \gamma^{k-1}, \delta\right); \quad \theta_i^k = \min_{j \in \mathcal{N}_i} \overline{\theta}_j^k.$$

`(S.3) Update of the dual stepsizes and auxiliary variables`: Each agent $i \in [m]$ updates their axillary variables and dual stepsizes:

$$\widetilde{\theta}_i^k = \begin{cases} \min_{j \in \mathcal{N}_i} \theta_j^k, & \text{if } k \equiv 1 \bmod d_i^k, \\ \min_{j \in \mathcal{N}_i} \gamma^k \widetilde{\theta}_j^k, & \text{else,} \end{cases} \qquad \pi_i^k = \begin{cases} \widetilde{\theta}_i^k, & \text{if } k \equiv 0 \bmod d_i^k, \\ \gamma^k \pi_i^{k-1}, & \text{else.} \end{cases}$$

`(S.4) Graph diameter estimation:` Each agent i updates its local diameter estimation:

$$d_i^{k+1} = \begin{cases} \max_{j \in \mathcal{N}_i} (2\, d_j^k), & \text{if } k \equiv 0 \bmod d_i^k \text{ and } \widetilde{\theta}_i^k \neq \min_{j \in \mathcal{N}_i} \widetilde{\theta}_j^k, \\ \max_{j \in \mathcal{N}_i} d_j^k, & \text{otherwise.} \end{cases}$$

`(S.5) Local updates of the primal and dual variables:`

$$\mathbf{X}^{k+1} = \mathbf{X}^{k+1/2} - \Theta^k \cdot \mathbf{Y}^{k+1/2}, \quad \mathbf{X}_\Pi^{k+1/2} = W\,(\Pi^k)^{-1}\mathbf{X}^k,$$
$$\mathbf{Y}^{k+1} = \mathbf{Y}^{k+1/2} + (\Pi^k)^{-1}\mathbf{X}^k - \mathbf{X}_\Pi^{k+1/2} - \nabla F(\mathbf{X}^{k+1/2}).$$

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

---

4 Convergence Analysis

Before diving into the convergence analysis, we introduce some preliminary definitions and facts.

Lemma 8. *Consider Problem (P) under Assumption 1; let $(\mathbf{X}^\star, \mathbf{Y}^\star)$ be a fixed point of Algorithm 1; and let $F : \mathbb{R}^{m\times d} \to \mathbb{R}$ be the augmented function defined as*

$$F(\mathbf{X}) := \frac{1}{m}\sum_{i=1}^{m} f_i(x_i), \quad x_i \in \mathbb{R}^d,\ i \in [m].$$

Then, the following hold:

(i) $\mathbf{X}^\star = \mathbf{1}(x^\star)^\top$, *where $x^\star$ is a solution of (P);*

(ii) $\mathbf{Y}^\star \in \mathbf{range}(I - W)$ *and* $\nabla F(\mathbf{X}^\star) + \mathbf{Y}^\star = 0$.

Every fixed point of Algorithm 1 yields consensual rows of $\mathbf{X}^\star$, being a solution of (P).

4.1 The case of strongly convex f_i

We introduce the following merit function evaluated along the iterates $\{\mathbf{X}^k, \mathbf{Y}^k\}$ of Algorithm 1:

$$V^k := \|\mathbf{X}^k - \mathbf{X}^\star\|^2 + (\theta_{\min}^{k-1})^2 \|\mathbf{Y}^k - \mathbf{Y}^\star\|_M^2,$$

where $(\mathbf{X}^\star, \mathbf{Y}^\star)$ is a fixed point of the algorithm (with $\mathbf{X}^\star = 1(x^\star)^\top$ and $x^\star$ being the unique solution of (P)); $M := c^{-1}(I - \widetilde{W})^\dagger - I$, with $(I - \widetilde{W})^\dagger$ denoting the Moore-Penrose pseudoinverse of $I - \widetilde{W}$; $\|\cdot\|_M$ is the norm induced by the matrix M; and $\theta_{\min}^k = \min_{i\in[m]} \theta_i^k$. Notice that V^k is a valid merit function: $V^k = 0$ if and only if $\mathbf{X}^k = \mathbf{X}^\star$ and $\mathbf{Y}^k = \mathbf{Y}^\star$ (due to $\mathbf{Y}^k \in \mathbf{range}(I - W) = \mathbf{range}(I - \widetilde{W})$).

Linear convergence of Algorithm 1 applied to strongly convex instances of (P) is summarized next.

Theorem 9. *Given Problem (P) under Assumption 1, with $\mu > 0$ and unique solution $x^\star \in \mathbb{R}^d$; let $\{(\mathbf{X}^k, \mathbf{Y}^k)\}$ be the iterates generated by Algorithm 1. Additionally, choose $\{\gamma^k\}$ as $\gamma^k \le \big((k+\beta_1)/(k+1)\big)^{\beta_2}$, for all k and some $\beta_1 \ge 1, \beta_2 > 0$. Then $V^k \le \varepsilon$, for all $k \ge N_\varepsilon$, with*

$$N_\varepsilon = \mathcal{O}\left(\frac{1}{\delta}\frac{\kappa \log_2 d_{\mathcal{G}}}{(1 - c(1-\lambda_m(\widetilde{W})))^2 c(1-\lambda_2(\widetilde{W}))}\left(\log(\max\{\|\mathbf{X}^\star\|^2, V^0\}/\varepsilon) + d_{\mathcal{G}}\right)\right). \tag{11}$$

Here κ is the condition number of each f_i restricted to the convex hull of $\{x^\star, \{x_i^k, x_i^{k+1/2}\}_{k=0}^{N_\varepsilon}\}$, and $\mathcal{O}$ hides the dependence on β_1 and β_2.

Because $1 - c(1 - \lambda_m(\widetilde{W})) > 1 - 2c$, the linear rate (11) yields (omitting poly-log factors)

$$N_\varepsilon = \widetilde{\mathcal{O}}\left(\frac{\kappa}{1-\lambda_2(\widetilde{W})}\log\left(\frac{1}{\varepsilon}\right) + d_{\mathcal{G}}\right).$$

Quite remarkably, this matches the complexity of the adaptive method in [17], which requires *global* min-consensus, whereas Algorithm 1 uses only one-hop exchanges; the extra term $d_{\mathcal{G}}$ is somehow expected, due to the min-consensus procedures. The complexity bound above is also on par with non-adaptive gradient-tracking schemes such as [29], SONATA [37], and [32]. Yet our condition number κ is *local*–evaluated on the convex hull of the trajectory–typically much smaller than the *global* condition number that governs the performance of those methods.

The sequence $\{\gamma^k\}_{k=1}^\infty$ (each $\gamma^k \ge 1$) used in Line 1 of the backtracking procedure, encourages *non-monotone* growth, enabling larger stepsizes between consecutive line-searches. Any choice with $\gamma \downarrow 1$ and $\prod_{k=1}^\infty \gamma^k = \infty$ is

suitable. In Theorem 9 we suggested a practical option we found effective. For maximal simplicity, one may instead set $\gamma^k = 1$, for all k, foregoing this extra parameter.

4.2 The case of (nonstrongly) convex f_i

In this setting, we will use the following merit function to monitoring progresses of the algorithm towards optimality: for any given fixed point $(\mathbf{X}^\star, \mathbf{Y}^\star)$ of Algorithm 1, let

$$\mathcal{M}(\mathbf{X}) := \max\left(\delta\|\mathbf{X}\|_{I-W}^2, F(\mathbf{X}) - F(\mathbf{X}^\star) + \langle \mathbf{Y}^\star, \mathbf{X}\rangle\right),$$

where $\delta > 0$ is as defined in the algorithm, and $\mathbf{X}^\star = 1(x^\star)^\top$. Note that $\mathcal{M}(\mathbf{X}) = 0$ if and only if $\mathbf{X} = \mathbf{1}x^\top$, for some x such that $(1/m)\sum_{i=1}^m f_i(x) \leq (1/m)\sum_{i=1}^m f_i(x^*)$ (such an $\mathbf{X}$ is such that $\langle \mathbf{Y}^\star, \mathbf{X}\rangle = 0$, due to $\mathbf{Y}^\star \in \mathbf{range}\,(I-W)$–see Lemma 8).

Convergence of Algorithm 1 is summarized next, where $(\mathbf{X}^\star, \mathbf{Y}^\star)$ is any given fixed point.

Theorem 10. *Given Problem (P) under Assumption 1, with $\mu = 0$, let $\{(\mathbf{X}^k, \mathbf{Y}^k)\}$ be the iterates generated by Algorithm 1. Choose $\{\gamma^k\}$ as $\gamma^k \leq \left((k+\beta_1)/(k+1)\right)^{\beta_2}$, for all k and some $\beta_1 \geq 1, \beta_2 > 0$. Further, let $R > 0$ and $c_\theta > 0$ such that $R \geq \|\mathbf{X}^{k+1} - \mathbf{X}^\star\|$, $R^2 \geq \max(V^0, \|\mathbf{X}^\star\|^2)$; and $c_\theta \geq L\theta_{\min}^k$. Then,*

$$\mathcal{M}(\widehat{\mathbf{X}}^k) \leq \varepsilon, \quad \textit{with} \quad \widehat{\mathbf{X}}^k := \frac{1}{k}\sum_{t=1}^k \mathbf{X}^t,$$

for all $k \geq N_\varepsilon$, and

$$N_\varepsilon = \mathcal{O}\left(\frac{c_\theta d_{\mathcal{G}} \tilde{L} R^2 \log 1/\varepsilon}{\varepsilon} + \frac{d_{\mathcal{G}} \log d_{\mathcal{G}} \tilde{L} R^2}{\varepsilon}\right),$$

where $\tilde{L}$ is the Lipschitz constant of each function f_i restricted to the convex hull of $\{x^\star, \{x_i^k, x_i^{k+1/2}\}_{k=0}^\infty\}$, and $\mathcal{O}$ hides the dependence on β_1 and β_2.

Theorem 10 guarantees a complexity of $\mathcal{O}(1/\varepsilon)$, aligning with the complexity of nonaccelerated, nonadaptive (first-order) decentralized optimization methods, e.g., [22, 36] applied to (P) ($\mu = 0$). Notably, the proposed method eliminates the need for parameter tuning. Furthermore, similar to the strongly convex scenario, the complexity depends on parameters defined over the convex hull formed by the algorithm's trajectory. To our knowledge, this is the first *adaptive* decentralized method for (nonstrongly) convex problems with provable convergence.

On the compactness of $\{\mathbf{X}^k, \mathbf{Y}^k\}$: Notice that Theorem 10 requires compactness of the iterates $\{\mathbf{X}^k, \mathbf{Y}^k\}$. This can be ensured by suitable choices of the sequence $\{\gamma^k\}$–two options are the following.

(i) Choose $\gamma^k = 1$, for all $k \geq K$ and some finite $K \in \mathbb{N}_+$. In such a case, one can show that there exists some finite iteration index, after which all primal (and thus dual) stepsizes do not increase, resulting in constant stepsizes over time and uniform across the agents.

(ii) Given $\widetilde{R} \in \mathbb{R}_{++}$, let us introduce additional local binary variables $h_i^k \in \{0, 1\}$ (initialized $h_i^0 = 1$), defined as

$$h_i^k = \begin{cases} 0, & \text{if } \max\left(\|x_i^k - x_i^0\|, \theta_i^{k-1}\|y_i^k - y_i^0\|\right) \geq \widetilde{R}, \\ \min\limits_{j\in\mathcal{N}_i} h_i^{k-1}, & \text{otherwise.} \end{cases} \tag{12}$$

This extra variables monitor the condition $\max\left(\|x_i^k - x_i^0\|, \theta_i^{k-1}\|y_i^k - y_i^0\|\right) \leq \widetilde{R}$.

Equipped with these variables, Algorithm 1 changes as follows:

(a) Add step `(S.0)` (before `(S.1)`) wherein (12) is performed; and

(b) Define

$$\widetilde{\gamma}_i^{k-1} = 1 + h_i^{k-1}(\gamma^{k-1} - 1),$$

and call the `Backtracking` procedure (step `(S.2)`), replacing γ^{k-1} with $\widetilde{\gamma}_i^{k-1}$.

This will guarantee that all the iterates generated by this variant of the algorithm are bounded.

5 Numerical Results

In this section, we compare the proposed method against the adaptive decentralized algorithm from [17] (considering both local and global min-consensus implementations) and the well-known EXTRA algorithm [36], whose stepsize is fine tuned for the best practical performance on the problems and network simulated. EXTRA serves as a benchmark representing decentralized algorithms that rely on stepsizes selected with full knowledge of both network topology and optimization parameters.

5.1 Strongly convex quadratic

As strongly convex instance of (P), we consider a quadratic optimization problem, with each $f_i(x) = \|A_i x - b_i\|^2$. Elements of $A_i \in \mathbb{R}^{h\times n}$ and $b_i \in \mathbb{R}^h$, with $h = 110$ and $n = 100$, are generated as i.i.d. from the standard normal distribution. We simulated three different network graphs, of size $m = 20$, namely: (i) line graph; (ii) Erdős–Rényi graph with edge probability $p = 0.1$ (modeling a sparse network); and (iii) Erdős–Rényi graph with edge probability $p = 0.5$ (modeling a fairly connected network). In our algorithm, we choose $\gamma^k = (k+2)/(k+1)$.

The comparison among the algorithms described above is illustrated in Fig. 2, where we plot the distance of the iterates from the optimal solution versus the number of communications. Remarkably, the proposed method consistently compares favorably against the adaptive algorithm from [17] relying on *global* min-consensus communications. Notably, the performance gap diminishes as network connectivity improves. Additionally, our approach demonstrates significantly greater stability compared to the local min-consensus variant from [17] and outperforms the EXTRA algorithm, despite requiring no prior knowledge of network topology or optimization parameters.

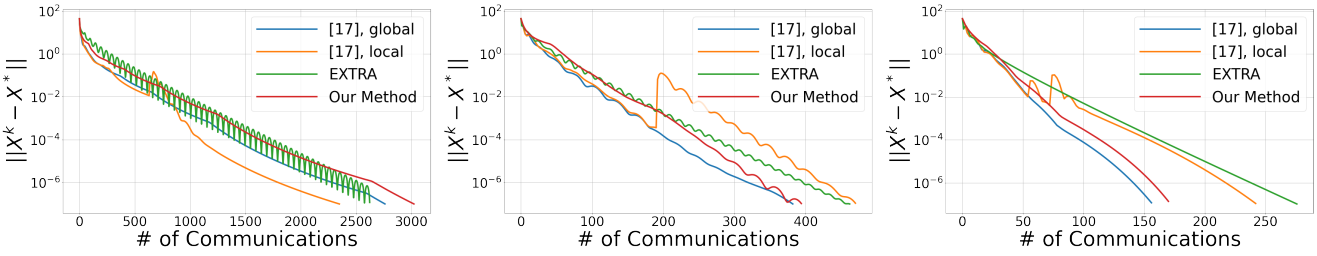
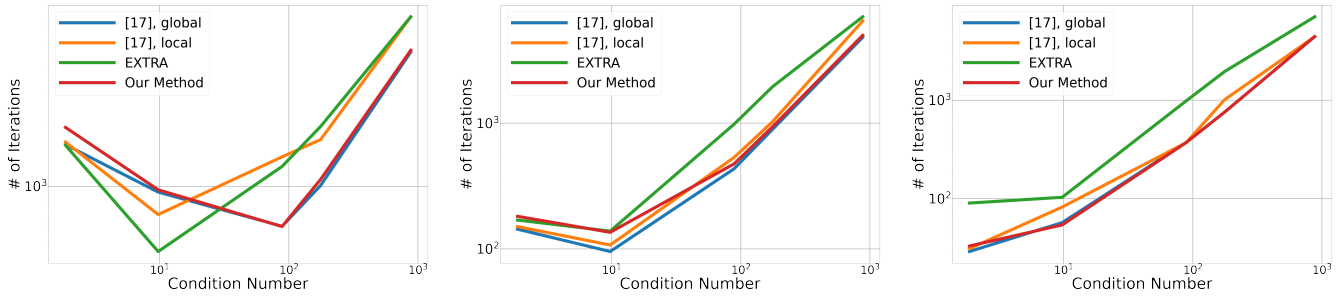


Figure 2: **Strongly convex quadratic** program on different graphs: (left) Line graph; (middle) Erdős-Rényi Graph with edge activation probability $p = 0.1$; (right) Erdős-Rényi Graph with $p = 0.5$

5.1.1 Dependence on the condition number

Let us consider now $f_i(x) = \|A_i x - b_i\|^2 + \lambda\|x\|^2/2$ so that changing λ one can control the condition number of the problem. Graphs and matrices A_i, b_i are generated as described in Figure 2, and so is the procedure followed to tune the simulated algorithms.



(a) Line Graph (b) Erdős-Rényi Graph, $p = 0.1$ (c) Erdős-Rényi Graph, $p = 0.5$

Figure 3: **Ridge regression**: Number of iterations N for $\|\mathbf{X}^N - \mathbf{X}^\star\| \leq 10^{-5}$ versus the condition number of agents' looses on different graphs; (a) Line graph; (b) Erdős-Rényi Graph, with edge activation probability $p = 0.1$; and (c) Erdős-Rényi Graph, with edge activation probability $p = 0.5$.

Results are reported in Figure 3. Quite remarkable, the proposed method outperforms EXTRA and [17] using local-min consensus and performs as good as [17] using *global* min-consensus, over all graphs simulated and mostly of the condition number values.

5.1.2 Dependence on the graph diameter

The algorithm complexity as stated in Theorems 9 and 10 contains some dependence on graph diameter $d_{\mathcal{G}}$. It is natural then to ask how the graph diameter value affects the numerical performance of the algorithm. We consider the same function $f_i(x) = \|A_i x - b_i\|^2$ with $A_i \in \mathbb{R}^{h \times n}$ with dimensions $h = 1$, $n = 100, and\ d = 100$. We simulated this problem over a line-graph with variable number of agents, hence variable diameter.

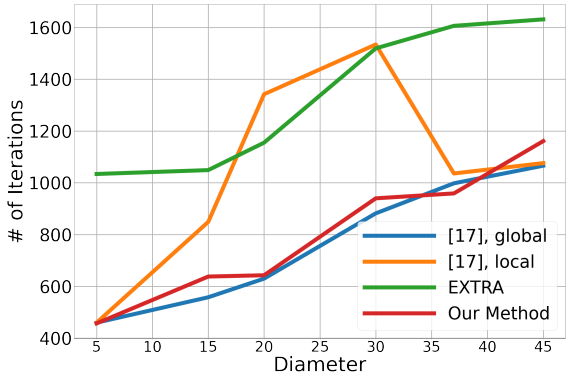


Figure 4: **Strongly convex quadratic** program on graph-line with different number of nodes: number of iterations N for $\|\mathbf{X}^N - \mathbf{X}^\star\| \leq 10^{-5}$ versus graph diameter

Figure 4 summarizes the results. Quite remarkably, the size of the graph seems not to affect the performance of the algorithm more than that of [17] employing global min synchronization of the agents' primal stepsizes. This is despite the additional dependence of the algorithm complexity on the graph diameter in Theorems 9 and 10.

5.2 Logistic regression

As (nonstrongly) convex instance of (P), we consider the decentralized logistic regression problem, corresponding to $f_i(x) = (1/h) \sum_{j=1}^{h} \log(1 + \exp(-b_{i,j} \langle f_{i,j}, a_{i,j} \rangle))$. Here, $b_{ij} \in \{0, 1\}$, $a_{ij} \in \mathbb{R}^{200}$ are data problem, taken from the

dataset a3a [3]. We distributed data across $m = 20$ nodes, each owning $h = 159$ samples. We simulated the same networks as in Fig. 2.

Results are summarized in Fig. 5; we plot the values of the merit function $\mathcal{M}$ along the ergodic iterates generated by all the algorithms versus the number of communications. Even on (nonstrongly) convex problems, the proposed method outperforms EXTRA and [17] using local min-consensus while being comparable with [17]. using global min-consensus. In such a setting however there is no convergence proof for [17]. Hence the proposed method is the first *adaptive* decentralized algorithm for (nonstrongly) convex instances of (P) with provable convergence.

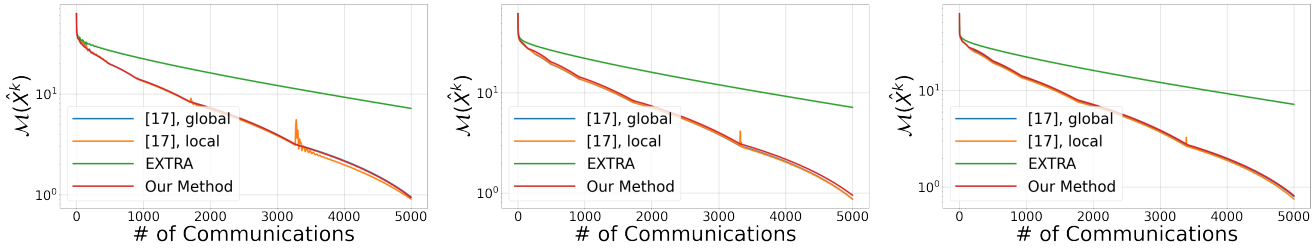


Figure 5: **Logistic regression** on different graphs: (left) Line graph; (middle) Erdős-Rényi Graph with edge activation probability $p = 0.1$; (right) Erdős-Rényi Graph with $p = 0.5$

References

[1] 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.

[2] J. Barzilai and J. M. Borwein. Two-point step size gradient methods. *IMA Journal of Numerical Analysis*, 8(1):141–148, 1988.

[3] C. Chang and C. Lin. Libsvm: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2, 07 2007.

[4] T. Chang, M. Hong, H. Wai, X. Zhang, and S. Lu. Distributed learning in the nonconvex world: From batch data to streaming and beyond. *IEEE Signal Processing Magazine*, 37(3):26–38, 2020.

[5] X. Chen, B. Karimi, W. Zhao, and P. Li. On the convergence of decentralized adaptive gradient methods. In *Asian Conference on Machine Learning*, pages 217–232. PMLR, 2023.

[6] X. Chen, X. Li, and P. Li. Toward communication efficient adaptive gradient method. In *Proceedings of the 2020 ACM-IMS on Foundations of Data Science Conference*, page 119–128, Virtual Event USA, October 2020. ACM.

[7] Ziqin Chen and Yongqiang Wang. Distributed optimization and learning with automated stepsizes. In *2024 IEEE 63rd Conference on Decision and Control (CDC)*, pages 3121–3126, 2024.

[8] A. Cutkosky and H. Mehta. Momentum improves normalized SGD. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 2260–2268. PMLR, 13–18 Jul 2020.

[9] 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*, pages 257–265, 2011.

[10] Iyanuoluwa E. and Chinwendu E. Q-linear convergence of distributed optimization with barzilai-borwein step sizes. In *58th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 1–8, 2022.

[11] J. Gao, XW. Liu, YH. Dai, HUang Y., and P. Yang. Achieving geometric convergence for distributed optimization with barzilai-borwein step sizes. *Sci. China Inf. Sci.*, 65:149–204, 2022.

[12] D. Ghaderyan and S. Werner. Fully adaptive stepsizes: Which system benefit more – centralized or decentralized? 2025. arXiv:2504.15196.

[13] J. Hu, X. Chen, L. Zheng, L. Zhang, and H. Li. (rectified version) the barzilai–borwein method for distributed optimization over unbalanced directed networks. *arXiv:2305.11469v3*, 2024.

[14] T. Janssen, N. BniLam, M. Aernouts, R. Berkvens, and M. Weyn. Lora 2.4 ghz communication link and range. *Sensors*, 20(16):4366, August 2020.

[15] D.H. Kim, J.Y. Lim, and J.D. Kim. Low-power, long-range, high-data transmission using wi-fi and lora. In *2016 6th International Conference on IT Convergence and Security (ICITCS)*, page 1–3, Prague, Czech Republic, September 2016. IEEE.

[16] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.

[17] I. Kuruzov, G. Scutari, and A. Gasnikov. Achieving linear convergence with parameter-free algorithms in decentralized optimization. In *Advances in Neural Information Processing Systems*, 2024.

[18] P. Latafat, A. Themelis, and P. Patrinos. Adaptive proximal algorithms for convex optimization under local lipschitz continuity of the gradient. *arXiv preprint arXiv:2301.04431*, 2023.

[19] J. Li, X. Chen, S. Ma, and M. Hong. Problem-parameter-free decentralized nonconvex stochastic optimization. *arXiv preprint arXiv:2402.08821*, 2024.

[20] X. Li, B. Karimi, and P. Li. On distributed adaptive optimization with gradient compression. In *International Conference on Learning Representations (ICLR)*, 2022.

[21] X. Li and F. Orabona. On the convergence of stochastic gradient descent with adaptive stepsizes. In *Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics (AISTAT)*. PMLR, 2019.

[22] Z. Li, W. Shi, and M. Yan. A decentralized proximal-gradient method with network independent step-sizes and separated convergence rates. *IEEE Transactions on Signal Processing*, 67(17):4494–4506, 2019.

[23] P. Di Lorenzo and G. Scutari. NEXT: In-network nonconvex optimization. *IEEE Transactions on Signal and Information Processing over Networks*, 2(2):120–136, 2016.

[24] Y. Malitsky and K. Mishchenko. Adaptive gradient descent without descent. In *International Conference on Machine Learning*, 2019.

[25] Y. Malitsky and K. Mishchenko. Adaptive proximal gradient method for convex optimization. *arXiv preprint arXiv:2308.02261*, 2024.

[26] P. Nazari, D.A. Tarzanagh, and G. Michailidis. Dadam: A consensus-based distributed adaptive gradient method for online optimization. *IEEE Transactions on Signal Processing*, 70:6065–6079, 2022.

[27] A. Nedic. Distributed gradient methods for convex machine learning problems in networks: Distributed optimization. *IEEE Signal Processing Magazine*, 37(3):92–101, 2020.

[28] A. Nedić, A. Olshevsky, and M. Rabbat. Network topology and communication-computation tradeoffs in decentralized optimization. *Proceedings of the IEEE*, 106:953–976, 2018.

[29] A. Nedić, A. Olshevsky, and W. Shi. Achieving geometric convergence for distributed optimization over time-varying graphs. *SIAM Journal on Optimization*, 27:2597–2633, July 2016.

[30] J. Nocedal and S. Wright. *Numerical Optimization*. Springer, 2 edition, 2006.

[31] B.T. Polyak. Minimization of unsmooth functionals. *USSR Computational Mathematics and Mathematical Physics*, 9(3):14–29, 1969.

[32] G. Qu and N. Li. Harnessing smoothness to accelerate distributed optimization. *IEEE Transactions on Control of Network Systems*, 5(3):1245–1260, Sept 2018.

[33] S. Reddi, Z. Burr Charles, M. Zaheer, Z. Garrett, K. Rush, J. Kone**v***cn*, S. Kumar, and B. McMahan. Adaptive federated optimization. In *International Conference on Learning Representations (ICLR)*, 2021.

[34] S. J. Reddi, S. Kale, and S. Kumar. On the convergence of adam and beyond. In *International Conference on Learning Representations (ICLR)*, 2018.

[35] A. H. Sayed. Adaptation, learning, and optimization over networks. *Foundations and Trends in Machine Learning*, 7:311–801, January 2014.

[36] W. Shi, Q. Ling, G. Wu, and W. Yin. EXTRA: An exact first-order algorithm for decentralized consensus optimization. *SIAM J. on Optimization*, 25(2):944–966, November 2015.

[37] Y. Sun, G. Scutari, and A. Daneshmand. Distributed optimization based on gradient-tracking revisited: Enhancing convergence rate via surrogation. *SIAM J. on Optimization*, 32:354–385, 2022.

[38] R. Ward, X. Wu, and L. Bottou. Adagrad stepsizes: Sharp convergence over nonconvex landscapes. *The Journal of Machine Learning Research*, 21:1–30, 2020.

[39] R. Xin, S. Pu, A. Nedic, and U. A. Khan. A general framework for decentralized optimization with first-order methods. *Proceedings of the IEEE*, 108(11):1869–1889, November 2020.

[40] D. Zhou, S. Ma, and J. Yang. Adabb: Adaptive barzilai-borwein method for convex optimization. *arXiv preprint arXiv:2401.08024*, 2024.