DADAO: Decoupled Accelerated Decentralized Asynchronous Optimization for Time-Varying Gossips

Adel Nabli, Edouard Oyallon

DADAO: Decoupled Accelerated Decentralized Asynchronous Optimization for Time-Varying Gossips

Adel Nabli
ISIR, CNRS, Sorbonne University
nabli@isir.upmc.fr

Edouard Oyallon
ISIR, CNRS, Sorbonne University

July 25, 2022

Abstract

DADAO is a novel decentralized asynchronous stochastic algorithm to minimize a sum of L-smooth and μ-strongly convex functions distributed over a time-varying connectivity network of size n. We model the local gradient updates and gossip communication procedures with separate independent Poisson Point Processes, decoupling the computation and communication steps in addition to making the whole approach completely asynchronous. Our method employs primal gradients and do not use a multi-consensus inner loop nor other ad-hoc mechanisms as Error Feedback, Gradient Tracking or a Proximal operator. By relating spatial quantities of our graphs χ_1^*, χ_2^* to a necessary minimal communication rate between nodes of the network, we show that our algorithm requires $\mathcal{O}(n\sqrt{\frac{L}{\mu}}\log\epsilon)$ local gradients and only $\mathcal{O}(n\sqrt{\chi_1^*\chi_2^*}\sqrt{\frac{L}{\mu}}\log\epsilon)$ communications to reach a precision ϵ. If SGD with uniform noise σ^2 is used, we reach a precision ϵ with same speed, up to a bias term in $\mathcal{O}(\frac{\sigma^2}{\sqrt{\mu L}})$. This improves upon the bounds obtained with current state-of-the-art approaches, our simulations validating the strength of our relatively unconstrained method. Our source-code is released on a public repository.

1 Introduction

With the rise of highly-parallelizable and connected hardware, distributed optimization for machine learning is a topic of significant interest helding many promises. In a typical distributed training framework, the goal is to minimize a sum of functions $(f_i)_{i\leq n}$ splitted across n nodes of a compute network. A corresponding optimization procedure consists in alternating local computation

and communication rounds between the nodes. Spreading the compute load is done to ideally obtain a *linear speedup* in the number of nodes. In the decentralized setting, there is no central machine aggregating the information sent by the workers: nodes are only allowed to communicate with their neighbours in the network. In this setup, optimal methods [34, 17] have been derived for synchronous first-order algorithms, whose executions are blocked until a subset (or all) nodes have reached a predefined states: the instructions must be performed in a specific order (*e.g.*, all nodes must perform a local gradient step before the round of communication begins), which is one of the locks limiting their efficiency in practice.

This work attempts to address simultaneously multiple limitations of existing decentralized algorithms, while guaranteeing fast rates of convergence. To tackle the synchronous lock, we rely on the continuized framework [9], originally introduced to allow asynchrony in a fixed topology setting: iterates are labelled with a continuous-time index *(in opposition to a global iteration count)* and performed locally with no regards to a specific global ordering of events. This is more practical, while being theoretically grounded and simplifying the analysis. However, in [9], gradient and gossip operations are still coupled: each communication along an edge requires the computation of the gradients of the two functions locally stored on the corresponding nodes and vice-versa. As more communications than gradient computations are necessary to reach an ϵ precision, even in an optimal framework [17, 34], the coupling directly implies an overload in terms of gradient steps. Another limitation is the restriction to a fixed topology: in a more practical setting, connections between nodes should be allowed to disappear or new ones to appear over time. The procedures of [19, 23] are the first to obtain an optimal complexity in terms of gradient steps while being robust to topological change. Unfortunately, synchrony is mandatory in their frameworks as they either rely on the Error-Feedback mechanism [35] or the Gradient Tracking one [26]. Moreover, they both use an inner-loop to control the number of gradient steps, at the cost of a significant increase of the amount of activated communication edges. To our knowledge, we are the first work to tackle those locks simultaneously.

In this paper, we propose a novel algorithm (DADAO: Decoupled Accelerated Decentralized Asynchronous Optimization) based on a combination of similar formulations to [17, 10, 12] in the continuized framework of [9]. We study:

$$\inf_{x\in\mathbb{R}^d} \sum_{i=1}^{n} f_i(x)\,, \tag{1}$$

where each $f_i : \mathbb{R}^d \to \mathbb{R}$ is a μ-strongly convex and L-smooth function computed in one of the n nodes of a network. We derive a first-order optimization algorithm which only uses primal gradients and relies on a time-varying Point-wise Poisson Processe (P.P.P.s [20]) modeling of the communication and gradient occurrences, leading to accelerated communication and computation rates. Our framework is based on a simple fixed point iteration and kept minimal: it only involves primal computations with an extra momentum term and works in both

the Gradient and Stochastic Gradient Descent (SGD) settings. Thus, we do not add other cumbersome designs such as the Error Feedback or Forward Backward used in [17], which are intrinsically not amenable in the continuized framework as they require synchrony. While we do not take into account the delays bound to appear in practice (we assume instantaneous communications and gradient computations), we show that the ordering of the gradient and gossip steps can be variable, removing the coupling lock.

Our contributions are as follows: **(1)** first, we propose the first primal algorithm with provable guarantees in the context of asynchronous decentralized learning with time-varying connectivity. **(2)** Compared to any reference work, this algorithm reaches accelerated rates of communication and computations while not requiring ad-hoc mechanisms obtained from an inner-loop. **(3)** Our algorithm also leads to accelerated rate with SGD with a minor modification. **(4)** We propose a simple theoretical framework compared to concurrent works and **(5)** we demonstrate its optimality numerically.

Our paper is structured as follows: in Sec. 3.1, we describe our work hypothesis as well as our model of a decentralized environment while Sec. 3.2 describes our dynamic. Sec. 3.3 sketches the proof of our convergence guarantees which is fully detailed in the Appendix. Then, Sec. 4.1 compares our work with its competitors, Sec. 4.2 explains our implementation of this algorithm and finally Sec. 4.3 verifies numerically our claims. All our experiments are reproducible, using PyTorch [29] and our code can be found online: https://github.com/AdelNabli/DADAO/.

2 Related Work

Table 1: This table highlights the strength of our method compared to concurrent works. Here, n is the number of node, $|\mathcal{E}|$ the number of edges, $\frac{1}{\chi_1}$ is the smallest strictly positive eigenvalue of a fixed stochastic Gossip matrix, also $1 \leq \chi_2 \leq \chi_1$ for acceleration, and γ is the eigengap. Note that under reasonable assumptions $|\mathcal{E}|\sqrt{\gamma} \geq \sqrt{\chi_1\chi_2}n$. Async., Comm., Grad., M.-C. and Prox. stand respectively for Asynchrony, Communication steps and Gradient steps., Multi-consensus and Proximal operator.

Method	Async.	Varying Topology	Decoupled	No Inner Loop (M.-C. or Prox.)	Primal Oracle	Total # Comm.	Total # Grad.
MSDA [34]	✗	✗	✗	✗	✗	$\sqrt{\gamma}\|\mathcal{E}\|\sqrt{\frac{L}{\mu}}$	$n\sqrt{\frac{L}{\mu}}$
AGT [23]	✗	✓	✗	✗	✓	$\chi_1\|\mathcal{E}\|\sqrt{\frac{L}{\mu}}$	$n\sqrt{\frac{L}{\mu}}$
ADOM+ [17]	✗	✓	✗	✗	✓	$\chi_1\|\mathcal{E}\|\sqrt{\frac{L}{\mu}}$	$n\sqrt{\frac{L}{\mu}}$
Continuized [9]	✓	✗	✗	✓	✗	$\sqrt{\chi_1\chi_2}n\sqrt{\frac{L}{\mu}}$	$\sqrt{\chi_1\chi_2}n\sqrt{\frac{L}{\mu}}$
ADFS [14]	✗	✗	✓	✗	✗	$\sqrt{\gamma}\|\mathcal{E}\|\sqrt{\frac{L}{\mu}}$	$n\sqrt{\frac{L}{\mu}}$
TVR [12]	✓	✗	✓	✓	✓	$\gamma\|\mathcal{E}\|\frac{L}{\mu}$	$n\frac{L}{\mu}$
Ours	✓	✓	✓	✓	✓	$\sqrt{\chi_1\chi_2}n\sqrt{\frac{L}{\mu}}$	$n\sqrt{\frac{L}{\mu}}$

Tab. 1 compares our contribution with other references, in order to highlight the benefits of our method.

Continuized and asynchronous algorithms. We highly rely on the elegant continuized framework [9], which allows to obtain simpler proofs along with bringing the flexibility of asynchronous algorithms. However in our work, we significantly reduce the necessary amount of gradient steps compared to [9], while maintaining the same amount of activated edges. Another type of asynchronous algorithms can also be found in [21], yet it fails to obtain Nesterov's accelerated rates for a lack of momentum. We note that [22] studies the robustness to delays, yet requires a shared memory and thus applies to a different context than decentralized optimization. [12] is a promising approach for modeling random communication on graphs yet fails, for now, to obtain acceleration in a neat framework that does not use inner-loops.

Decentralized algorithms with fixed topology. [34] is the first work to derive an accelerated algorithm for decentralized optimization, and it links the convergence speed to the Laplacian eigen-gap. The corresponding algorithm uses a dual formulation as well as a Chebychev acceleration (which is synchronous and for a fixed topology), yet as stated in Tab. 2, it still requires a significant amount of edges activated. Furthermore, under a relatively flexible condition on the intensity of our P.P.P.s, we show that our work improves over bounds that depend on the spectral gap. An emerging line of work following this formulation employs the continuized framework [11, 9, 10], but are unfortunately not amenable to incorporate a time varying topology by essence, as they rely on a coordinate descent scheme in the dual [28]. We note that [10] incorporates delays in their model, using the same technique as our work, yet transferring this robustness to another setting remains unclear. Reducing the number of communication has been studied in [25], only in the context of constant topology and without obtaining accelerated rates. [14] allows to obtain fast communication and gossip rates, yet requires a proximal step as well as synchrony between nodes to apply a momentum variable.

Decentralized algorithms with varying topology. We highlight that [17, 23, 15] are some of the first works to propose a framework for decentralized learning in the context of varying topology. However, they rely on an inner-loop propagating variables multiple times through a network, which imposes full synchrony and a communication overhead. In addition, as noted empirically in [24], inner-loops lead to a plateau-effect. Furthermore, we note that [18, 31] employ a formulation derived from [33, 7], casting decentralized learning as a monotonous inclusion, obtaining a linear rate thanks to a preconditioning step of a Forward-Backward like algorithm. However, being sequential by nature, this type of algorithm is not amenable to a continuized framework.

Error feedback/Gradient tracking. A major lock for asynchrony is the use of Gradient Tracking [16, 26, 23] or Error Feedback [35, 18]. Indeed, gradient operations are locally tracked by a running mean variable which must be updated at each gradient update: this operation is not compatible with an asynchronous

framework as it requires communication between nodes. Furthermore, to obtain accelerated rates, a multi-consensus inner-loop seems mandatory, which is again not desirable.

Decoupling procedures Decoupling subsequent steps of an optimization procedures traditionally allows subsequent speed-ups [14, 12, 2, 3]. This contrasts with methods which couple gradient and gossip updates, such that they happen in a predefined order, i.e. simultaneously [9] or sequentially [17, 15]. In decoupled optimization procedures, inner-loops are not desirable because they require an external procedure that can be potentially slow and require a block-barrier instruction during the execution of the algorithm (e.g., [14, 13]).

Notations

We introduce our necessary notations: for a positive semi-definite matrix A, $\|x\|_A \triangleq x^\mathsf{T} A x$, $f = \mathcal{O}(g)$ means there is a constant $C > 0$ such that $|f| \leq C|g|$, e_i is the canonical basis, $\mathbf{1}$ is the vector of 1, $\mathbf{I}$ the identity, A^+ is the pseudo-inverse of A and for a smooth convex functions F, $d_F(x, y) \triangleq F(x) - F(y) - \langle \nabla F(y), x - y \rangle$ is its Bregman divergence. We further write $\mathbf{e}_i \triangleq e_i \otimes \mathbf{I}$.

3 Fast Asynchronous Algorithm for Time-Varying Connectivity Networks

3.1 Gossip Framework

We consider the problem defined by Eq. 1 in a distributed environment constituted by n nodes whose dynamic is indexed by a continuous time index $t \in \mathbb{R}^+$. Each node has a local memory and can compute a local gradient ∇f_i, as well as elementary operations, in an instantaneous manner. As said above, having no delay is less realistic, yet adding them also leads to significantly more difficult proofs whose adaptation to our framework remains largely unclear. Next, we will assume that our computations and gossip result from independent inhomogeneous piecewise constant P.P.P. with no delay. For the sake of simplicity, we assume that all nodes can compute a gradient at the same rate:

Assumption 3.1 (Homogeneous gradient computations)**.** *The gradient computations are re-normalized to fire independently at a rate of 1 computation per second. For the i-th worker, we write $N_i(t)$ the corresponding P.P.P. of rate 1, as well as $\mathbf{N}(t) = (N_i(t))_{i \leq n}$.*

Next, we will also model the bandwidth of each machine. For an edge $(i, j) \in \mathcal{E}(t)$, we write $M_{ij}(t)$ the P.P.P. with rate $\lambda_{ij}(t) \geq 0$. When this P.P.P. fires, both nodes can potentially share their local memories. The rate $\lambda_{ij}(t)$ is adjustable locally by machine i, which can decide to speed or slow-down its local communication. While $\lambda_{ij}(t)$ and $\lambda_{ji}(t)$ may refer to different quantities, we highlight that this communication process is symmetric and we denote by

$\bar{\mathcal{E}}(t)$ the corresponding undirected graph. Given our notations, we note that if $(i,j) \notin \mathcal{E}(t)$, then the connexion between (i,j) can be thought as a P.P.P. with intensity 0. We now introduce the instantaneous expected gossip matrix of our graph:

$$\Lambda(t) \triangleq \sum_{(i,j)\in\mathcal{E}(t)} \lambda_{ij}(t)(e_i - e_j)(e_i - e_j)^\mathsf{T} .$$

We also write $\mathbf{\Lambda}(t) \triangleq \sum_{(i,j)\in\mathcal{E}(t)} \lambda_{ij}(t)(\mathbf{e}_i - \mathbf{e}_j)(\mathbf{e}_i - \mathbf{e}_j)^\mathsf{T}$ its tensorized counterpart that will be useful for our proofs and defining our Lyapunov potential. Following [34], we will further compare this quantity to the centralized gossip matrix:

$$\pi \triangleq \mathbf{I} - \frac{1}{n}\mathbf{1}\mathbf{1}^\mathsf{T} = \frac{1}{2n}\sum_{i,j}(\mathbf{e}_i - \mathbf{e}_j)(\mathbf{e}_i - \mathbf{e}_j)^\mathsf{T} .$$

In order to characterize the connectivity of $\Lambda(t)$, we introduce its instantaneous connectivity which is similar to [17], given by:

$$\frac{1}{\chi_1(t)} \triangleq \inf_{x\perp\mathbf{1},\|x\|=1} x^\mathsf{T}\Lambda(t)x .$$

We might also write $\chi_1[\Lambda(t)]$ in order to avoid confusions, depending on the context. Next, we introduce the maximal effective resistance of the network, as in [9, 8]:

$$\chi_2(t) \triangleq \frac{1}{2}\sup_{(i,j)\in\mathcal{E}(t)} (e_i - e_j)^\mathsf{T}\Lambda^+(t)(e_i - e_j) .$$

We remind the following Lemma, which will be useful to control $\chi_1(t), \chi_2(t)$ and to compare our bounds with the bounds that make use of the spectral gap of a graph:

Lemma 3.1 (Bound on the connectivity constants)**.** *The spectrum of $\Lambda(t)$ is non-negative. Furthermore, we have $\chi_1(t) = +\infty$ iff $\bar{\mathcal{E}}(t)$ is not a connected graph. Also, if the graph is connected, then:*

$$\frac{n-1}{\mathrm{Tr}\,\Lambda(t)} \leq \min(\chi_1(t), \chi_2(t)) .$$

Also, assume in addition that $c\lambda_{ij}(t) \geq \frac{\mathrm{Tr}\,\Lambda(t)}{2|\mathcal{E}(t)|}$ for some $c > 0$, then $c \geq 1$ and:

$$\chi_2(t) \leq \frac{(n-1)|\mathcal{E}(t)|}{c\mathrm{Tr}\,\Lambda(t)} .$$

Proof. We note that $\Lambda(t)$ is symmetric and has a non-negative spectrum, as:

$$x^\mathsf{T}\Lambda(t)x = \sum_{(i,j)\in\mathcal{E}(t)} \lambda_{ij}(t)\|x_i - x_j\|^2 .$$

From this, we also clearly see that $\chi_1(t) = +\infty$ iff the graph is disconnected. Next, we note that:

$$\sum_{(i,j)\in\mathcal{E}(t)} \lambda_{ij}(t)(e_i - e_j)^\mathsf{T}\Lambda^+(t)(e_i - e_j) = \mathrm{Tr}(\Lambda^+(t)\Lambda(t)) = n-1\,. \tag{2}$$

Thus,

$$n-1 \leq 2\chi_2(t)\sum_{(i,j)\in\mathcal{E}(t)} \lambda_{ij}(t) = \chi_2(t)\mathrm{Tr}(\Lambda(t))\,.$$

Next, it's clear that $\mathrm{Tr}(\Lambda(t)) = 2\sum_{(i,j)\in\mathcal{E}(t)} \lambda_{ij}(t) \geq \frac{\mathrm{Tr}(\Lambda(t))|\mathcal{E}(t)|}{c|\mathcal{E}(t)|}$ and thus $c \geq 1$. Following the same reasonning, we also got that:

$$\lambda_{ij}(t)(e_i - e_j)^\mathsf{T}\Lambda^+(t)(e_i - e_j) \leq n-1\,, \tag{3}$$

and

$$c\frac{\mathrm{Tr}\Lambda(t)}{2|\mathcal{E}(t)|}(e_i - e_j)^\mathsf{T}\Lambda^+(t)(e_i - e_j) \leq n-1\,. \tag{4}$$

Thus,

$$\chi_2(t) \leq \frac{(n-1)|\mathcal{E}(t)|}{c\mathrm{Tr}(\Lambda(t))}\,. \tag{5}$$

□

The last part of this Lemma allows to bound $\chi_2(t)$ when no degenerated behavior on the edge sampling happens: $c = 1$ corresponds to a uniform sampling of edges. The following assumption is necessary to avoid oscillatory effects due to the variations of $\Lambda(t)$:

Assumption 3.2 (Slowly varying graphs)**.** *Assume that $\Lambda(t)$ is piecewise constant on time intervals.*

In particular, it implies that each $\lambda_{ij}(t)$ is piece-wise constant. Next we bound uniformly the connectivity of our gossip environment in order to avoid the degenerated effect of an unbounded spectral gap and is similar to [17]:

Assumption 3.3 (Strongly connected topology of the expected gossip)**.** *Assume that there is $\chi_1^* > 0$ such that $\chi_1(t) \leq \chi_1^*$.*

We might write this quantity $\chi_1^*[\Lambda]$ to stress the depdency in $\Lambda(t)$. From supra, it's clear that $\chi_2(t) \leq \chi_1(t)$ so that under 3.3, $\chi_2(t)$ is upper bounded by $0 < \chi_2^* \leq \chi_1^*$, and this quantity will allow to get accelerated rates.

3.2 Dynamic to optimum

Next, we follow a standard approach [19, 17, 32, 12] for solving Eq. 1, which consists in introducing an extra-dual variable $\hat{x}$, for $0 < \nu < \mu$:

$$\begin{aligned}
(1) &= \inf_{x=\hat{x},\pi\hat{x}=0} \sum_{i=1}^{n} f_i(x_i) - \frac{\nu}{2}\|x\|^2 + \frac{\nu}{2}\|\hat{x}\|^2 \\
&= \inf_{x,\hat{x}} \sup_{y,z} \sum_{i=1}^{n} f_i(x_i) - \frac{\nu}{2}\|x\|^2 + \frac{\nu}{2}\|\hat{x}\|^2 + y^{\mathsf{T}}(\hat{x}-x) + z^{\mathsf{T}}(\pi\hat{x}) \\
&= \inf_{x} \sup_{y,z} \inf_{\hat{x}} \sum_{i=1}^{n} f_i(x_i) - \frac{\nu}{2}\|x\|^2 + \frac{\nu}{2}\|\hat{x}\|^2 + y^{\mathsf{T}}(\hat{x}-x) + z^{\mathsf{T}}(\pi\hat{x}) \\
&= \inf_{x} \sup_{y,z} \sum_{i=1}^{n} f_i(x_i) - \frac{\nu}{2}\|x\|^2 - x^{\mathsf{T}}y - \frac{1}{2\nu}\|\pi z + y\|^2 .
\end{aligned}$$

Introducing the convex function $F(x) = \sum_{i=1}^{n} f_i(x_i) - \frac{\nu}{2}\|x\|^2$, the saddle points (x^*, y^*, z^*) of this Lagrangian, are given by:

$$\left\{ \begin{array}{ll} \nabla F(x^*) - y^* & = 0 \\ \frac{y^* + \pi z^*}{\nu} + x^* & = 0 \\ \pi z^* + \pi y^* & = 0 . \end{array} \right. \tag{6}$$

Contrary to [17], we do not employ a Forward-Backward algorithm, which requires both an extra-inversion step and additional regularity on the considered proximal operator. Not only this condition does not hold in this precise case, but this is not desirable in a continuized framework where iterates are not ordered in a predefined sequence and requires a local descent at each instant. Another major difference is that no Error-feedback is required by our approach, which is a lock for asynchrony, makes the proof more challenging and leads to additional communications. Instead, we show it is enough to incorporate a standard fixed point algorithm, *without any specific preconditioning* (see [6]). We consider the following dynamic:

$$\left\{ \begin{array}{l}
dx_t = \eta(\tilde{x}_t - x_t)dt - \gamma(\nabla F(x_t) - \tilde{y}_t)\, d\mathbf{N}(t) \\
d\tilde{x}_t = \tilde{\eta}(x_t - \tilde{x}_t)dt - \tilde{\gamma}(\nabla F(x_t) - \tilde{y}_t)\, d\mathbf{N}(t) \\
d\tilde{y}_t = -\theta(y_t + z_t + \nu\tilde{x}_t)dt + (\delta + \tilde{\delta})(\nabla F(x_t) - \tilde{y}_t)d\mathbf{N}(t) \\
dy_t = \alpha(\tilde{y}_t - y_t)dt \\
dz_t = \alpha(\tilde{z}_t - z_t)dt - \beta \sum_{(i,j)\in\mathcal{E}(t)} (\mathbf{e}_i - \mathbf{e}_j)(\mathbf{e}_i - \mathbf{e}_j)^{\mathsf{T}}(y_t + z_t)dM_{ij}(t) \\
d\tilde{z}_t = \tilde{\alpha}(z_t - \tilde{z}_t)dt - \tilde{\beta} \sum_{(i,j)\in\mathcal{E}(t)} (\mathbf{e}_i - \mathbf{e}_j)(\mathbf{e}_i - \mathbf{e}_j)^{\mathsf{T}}(y_t + z_t)dM_{ij}(t) ,
\end{array} \right. \tag{7}$$

where $\nu, \tilde{\eta}, \eta, \gamma, \alpha, \tilde{\alpha}, \theta, \delta, \tilde{\delta}, \beta, \tilde{\beta}$ are undetermined parameters yet. As in [27], variables are paired in order to obtain a Nesterov acceleration. The variables (x, y) allow to decouple the gossip steps from the gradient steps by using independent P.P.P.s. Furthermore, the Lebesgue integrable path of $\tilde{y}_t$ does not

correspond to a standard momentum, as in a continuized framework [9]; however it turns out to be a crucial component of our method. Compared to [17], no extra multi-consensus step needs to be integrated. Our formulation of an asynchronous gossip step is similar to [9] which introduces a stochastic variable on edges; however, contrary to this work, our gossip and gradient computations are decoupled and our rates are accelerated. In fact, we can also consider SGD [4], by replacing $\nabla F(x)$ by an estimate $\nabla F(x,\xi)$, for $\xi \in \Xi$, some measurable space. We will need the following assumption on the bias and variance of the gradient:

Assumption 3.4 (Unbiased gradient with uniform additive noise)**.** *We assume that the estimate of the gradient has no bias:*

$$\mathbb{E}_\xi \nabla F(x,\xi) = \nabla F(x)\,,$$

and that its quadratic error is uniformly bounded by $\sigma > 0$:

$$\mathbb{E}_\xi \|\nabla F(x,\xi) - \nabla F(x)\|^2 \leq \sigma^2\,.$$

Next, for SGD use, we simply modify the three first lines of Eq. (7) that we replace by:

$$\begin{cases} dx_t = \eta(\tilde{x}_t - x_t)dt - \gamma \int_\Xi (\nabla F(x_t,\xi) - \tilde{y}_t)\, d\mathbf{N}(t,\xi) \\ d\tilde{x}_t = \tilde{\eta}(x_t - \tilde{x}_t)dt - \tilde{\gamma} \int_\Xi (\nabla F(x_t,\xi) - \tilde{y}_t)\, d\mathbf{N}(t,\xi) \\ d\tilde{y}_t = -\theta(y_t + z_t + \nu\tilde{x}_t)dt + (\delta+\tilde{\delta}) \int_\Xi (\nabla F(x_t,\xi) - \tilde{y}_t) d\mathbf{N}(t,\xi)\,, \end{cases}$$

Simulating those SDEs [1] can be efficiently done in standard numerical frameworks, as explained in Sec. 4.3.

3.3 Theoretical guarantees

We follow the approach introduced in [9] for studying the convergence of (7). To do so, we introduce the following Lyapunov potential $X \triangleq (x,\tilde{x},\tilde{y}), Y \triangleq (y,z,\tilde{z},m)$:

$$\begin{aligned}\Phi(t,X,Y) \triangleq A_t\|x-x^*\|^2 + \tilde{A}_t d_F(x,x^*) + B_t\|y-y^*\|^2 + \tilde{B}_t\|\tilde{y}-y^*\|^2 \\ + C_t\|z+y-z^*-y^*\|^2 + \tilde{C}_t\|\tilde{z}-z^*\|^2_{\mathbf{\Lambda}(t)^+}\,,\end{aligned}$$

where $A_t, \tilde{A}_t, B_t, \tilde{B}_t, C_t, \tilde{C}_t, D_t$ are non-negative functions to be defined. We will use this potential to control the trajectories of $X_t \triangleq (x_t,\tilde{x}_t,\tilde{y}_t), Y_t \triangleq (y_t,z_t,\tilde{z}_t)$, and we note that our dynamic can be conveniently rewritten as:

$$\begin{cases} dX_t = a_1(X_t,Y_t)dt + b_1(X_t)d\mathbf{N}(t) \\ dY_t = a_2(X_t,Y_t)dt + \sum_{(i,j)\in\mathcal{E}(t)} b_2^{ij}(Y_t)dM_{ij}(t)\,, \end{cases}$$

where $a_1, a_2, b_1 = (b_1^i)_i, (b_2^{ij})_{ij}$ are smooth functions.

Theorem 3.2 (Gradient Descent)**.** *Assume each f_i is μ-strongly convex and L-smooth. We assume 3.1-3.3, and we also assume that $\chi_1^*\chi_2^* \leq \frac{1}{2}$. Then there exists some parameters for the dynamic Eq. (7) and $c > 0$ (independent from χ_2^*), such that for any initialization $x_0, \tilde{x}_0, y_0, \tilde{y}_0, z_0 \in span(\pi), \tilde{z}_0 \in span(\pi)$, we get for $t \in \mathbb{R}^+$:*

$$\mathbb{E}[\|x_t - x^*\|^2] \leq C_0 e^{-ct\sqrt{\frac{\mu}{L}}},$$

where x^* is the solution of 1 and $C_0 > 0$ is a constant which depends only on the initialization.

Proof. Because Φ is smooth and $\mathcal{E}(t)$ is constant on intervals, we get via Ito's formula [20] applied to the semi-martingale (X_t, Y_t), gluing intervals where $\mathcal{E}(t)$ is constant (as well as the weights $\lambda_{ij}(t)$), that:

$$\begin{aligned}\Phi(t, X_t, Y_t) = \Phi(0, X_0, Y_0) &+ \int_0^T \langle \nabla\Phi(t, X_t, Y_t), \begin{pmatrix} 1 \\ a_1(X_t, Y_t) \\ a_2(X_t, Y_t) \end{pmatrix} \rangle dt \\ &+ \sum_{i=1}^n \int_0^T \big(\Phi(t, X_t + b_1^i(X_t), Y_t) - \Phi(t, X_t, Y_t)\big) dt \\ &+ \sum_{(i,j)\in\mathcal{E}(t)} \int_0^T \big(\Phi(t, X_t, Y_t + b_2^{ij}(Y_t)) - \Phi(t, X_t, Y_t)\big)\lambda_{ij}(t) dt + \Theta_T\,,\end{aligned}$$

where:

$$\begin{aligned}\Theta_T \triangleq &\sum_{i=1}^n \int_0^T \big(\Phi(t, X_{t^-}, Y_{t^-} + b_1^i(X_{t^-})) - \Phi(u, X_{t^-}, Y_{t^-})\big)(dN_i(t) - dt) \\ &+ \sum_{(i,j)\in\mathcal{E}(t)} \int_0^T \big(\Phi(t, X_{t^-} + b_2^{ij}(X_{t^-}), Y_{t^-}) - \Phi(t, X_{t^-}, Y_{t^-})\big)(dM_{ij}(t) - \lambda_{ij}(t)dt)\,.\end{aligned}$$

We will use the following technical Lemma, which is also difficult to prove and whose proof is deferred to the appendix:

Lemma 3.3. *There exists some parameters $\nu, \tilde{\eta}, \eta, \gamma, \tilde{\gamma}, \alpha, \tilde{\alpha}, \theta, \delta, \tilde{\delta}, \beta, \tilde{\beta}$ and $c > 0$ such that:*

$$\begin{aligned}\langle \nabla\Phi(t, X_t, Y_t), \begin{pmatrix} 1 \\ a_1(X_t, Y_t) \\ a_2(X_t, Y_t) \end{pmatrix} \rangle &+ \big(\Phi(t, X_t + b_1(X_t), Y_t) - \Phi(t, X_t, Y_t)\big) \\ &+ \sum_{(i,j)\in\mathcal{E}(t)} \lambda_{ij}(t)\big(\Phi(t, X_t, Y_t + b_2^{ij}(Y_t)) - \Phi(t, X_t, Y_t)\big) \leq 0 \ \ a.s.\ ,\end{aligned}$$

with $A'_t = c\sqrt{\frac{\mu}{L}} A_t$, with $A_0 = 1$.

Following the two lemma above, we get that:

$$0 \leq \mathbb{E}[\Phi(t, X_t, Y_t)] \leq \mathbb{E}[\Phi(0, X_0, Y_0)]\,.$$

We thus know that $A_t = e^{c\sqrt{\frac{\mu}{L}}}$, which implies that:

$$\mathbb{E}[A_t \|x_t - x^*\|^2] \leq \mathbb{E}[\Phi(0, X_0, Y_0)]\,,$$

and we can obtain the conclusion of the theorem. □

We can obtain the following corollary, with a minor modification of our current proof:

Corollary 3.3.1 (Stochastic Gradient Descent)**.** *Assume each f_i is μ-strongly convex and L-smooth. We assume 3.1-3.4, and we also assume that $\chi_1^*\chi_2^* \leq \frac{1}{2}$. Then, for the SGD-dynamic Eq.* (8)*, the same parameters as Thm. 3.2 allows to obtain for $t \in \mathbb{R}^+$:*

$$\mathbb{E}[\|x_t - x^*\|^2] \leq C_0 e^{-ct\sqrt{\frac{\mu}{L}}} + C_1 \frac{1}{\sqrt{\mu L}}\,,$$

where x^* is the solution of 1, $C_0 > 0$ is the same constant as in Thm. 3.2 and C_1 is an absolute constant.

Proof. We remind the SGD version of our Lemma:

Lemma 3.4. *There exists some parameters $\nu, \tilde{\eta}, \eta, \gamma, \tilde{\gamma}, \alpha, \tilde{\alpha}, \theta, \delta, \tilde{\delta}, \beta, \tilde{\beta}$ and $c > 0, C > 0$ such that:*

$$\langle \nabla\Phi(t, X_t, Y_t), \begin{pmatrix} 1 \\ a_1(X_t, Y_t) \\ a_2(X_t, Y_t) \end{pmatrix} \rangle + \big(\Phi(t, X_t + b_1(X_t), Y_t) - \Phi(t, X_t, Y_t)\big) \tag{8}$$

$$+ \sum_{(i,j)\in\mathcal{E}(t)} \lambda_{ij}(t)\big(\Phi(t, X_t, Y_t + b_2^{ij}(Y_t)) - \Phi(t, X_t, Y_t)\big) \leq C A_t \frac{1}{L} \text{ a.s. }, \tag{9}$$

with $A'_t = c\sqrt{\frac{\mu}{L}} A_t$, with $A_0 = 1$.

The proof follows the same path, except that we have an extra term which writes:

$$\int_0^T \frac{A_t}{L} = e^{c\sqrt{\frac{\mu}{L}}} \mathcal{O}(\frac{1}{\sqrt{\mu L}}) \tag{10}$$

which leads to the conclusion following an identical path. □

We note that as claimed in [9], it would be possible to optimize L in order to adjust the trace-off bias-variance of our descent.

4 Practical implementation

4.1 Expected computational complexity

For a given graph $\mathcal{E}(t)$, multiple choices of $\Lambda(t)$ are possible and would still lead to accelerated rates as long as the condition $2\chi_1^*[\Lambda]\chi_2^*[\Lambda] \leq 1$ is verified. Thus, we discuss how to chose our instantaneous expected gossip matrix in order to compare to concurrent work. From the previous theorem and under the assumptions from the previous subsection, we deduce that to get a precision ϵ, a total of $T = \mathcal{O}(\sqrt{\frac{L}{\mu}}\log\epsilon)$ local gradient computations is required for each machine, which will happen, in expectation, at time T. Furthermore, the expected number of edges activated is given by:

$$\mathbb{E}[\int_0^T \sum_{(i,j)\in\mathcal{E}(t)} \lambda_{ij}(t)\,dt] = \frac{1}{2}\int_0^T \mathbb{E}[\mathrm{Tr}\,\Lambda(t)]\,dt\,. \tag{11}$$

More details can be found in Appendix C on our methodology for comparing with other methods, in particular the way to recover the order of magnitudes we mention. In the following, each algorithm to which we compare ourselves is parameterized by a Laplacian matrix with various properties. Systematically, for an update frequency f and a family of Laplacians $\{\mathcal{L}_q\}_q$ (which can be potentially reduced to a single element) given by concurrent work, we will set:

$$\Lambda(t) = \underbrace{\sqrt{2\chi_1^*[\mathcal{L}]\chi_2^*[\mathcal{L}]}}_{\triangleq\lambda^*}\mathcal{L}_{\lfloor tf\rfloor}\,, \tag{12}$$

where λ^* can be understood as a lower bound on the instantaneous expected rate of communication. In this case, it is clear that $\Lambda(t)$ satisfies the conditions of Thm. 3.2 or Corollary 3.3.1. From a physical point of view, it allows to relate the spatial quantities of our graphs to a necessary minimal communication rate between nodes of the network, see Appendix B for a discussion on this topic.

Comparison with ADOM+. In ADOM+ [17], one picks $\chi_1^*[\mathcal{L}] \geq 1$ and $f = \chi_1^*[\mathcal{L}]$. Then, the number of gossip steps of our algorithm is at most:

$$\sqrt{\chi_1^*[\mathcal{L}]\chi_2^*[\mathcal{L}]}\sup_q \mathrm{Tr}(\mathcal{L}_q)\sqrt{\frac{L}{\mu}}\log\epsilon = \mathcal{O}(\sqrt{\chi_1^*[\mathcal{L}]\chi_2^*[\mathcal{L}]}n\sqrt{\frac{L}{\mu}}\log\epsilon)$$

In this case, the expected computational complexity of ADOM+ is given by:

$$\sum_{t=1}^T \chi_1^*[\mathcal{L}]|\mathcal{E}(t)| \geq \mathcal{O}(\sqrt{\chi_1^*[\mathcal{L}]\chi_2^*[\mathcal{L}]}n\sqrt{\frac{L}{\mu}}\log\epsilon)\,,$$

which is potentially substentially higher than ours.

Comparison with standard Continuized. If $\mathcal{L}$ is a Laplacian picked such that $\operatorname{Tr}\mathcal{L} = 2$, as in [9], then [9] claims that at least

$$\mathcal{O}(\sqrt{\frac{L}{\mu}}\log\epsilon\sqrt{\chi_1^*[\mathcal{L}]\chi_2^*[\mathcal{L}]}) \tag{13}$$

gradient and gossip iterations are needed. The number of gossip iterations is the same as ours, yet, thanks to Lemma 3.1, the number of gradient iterations can be substantially higher without any additional assumptions, as $n-1 \leq 2\sqrt{\chi_1^*[\mathcal{L}]\chi_2^*[\mathcal{L}]}$. Furthermore, the computations of [9] still use the dual gradients and are for a fixed topology.

Comparison with methods that depends on the spectral gap. For instance, MSDA relies on a Tchebychev acceleration of the number of gossip steps (which is possible because [34] uses a fixed gossip matrix) and which allows to get a number of edges activated of the order of magnitude of:

$$\gamma^*|\mathcal{E}|\sqrt{\frac{L}{\mu}}\log\epsilon\,,$$

where γ^* is the spectral gap. For our algorithm, the number of gossip writes, with $f = 1$:

$$\sqrt{\frac{L}{\mu}}\log\epsilon\sqrt{\chi_1^*[\mathcal{L}]\chi_2^*[\mathcal{L}]}\operatorname{Tr}\mathcal{L} \leq \mathcal{O}(\gamma^*|\mathcal{E}|\sqrt{\frac{L}{\mu}}\log\epsilon)\,,$$

where the details of this bound can be found in the Appendix and relies solely on an assumption on the minimal weights in the Laplacian. We highlight that [34, 17] claimed that their respective algorithms are optimal because they can solve a worst-case graph in term of computations and number of synchronized gossips; our claim is, *by nature* different, as we are interested by the number of edges fired rather than the number of synchronized gossip rounds. Tab. 2 predicts the behavior of our algorithm for various class of graphs which are encoded via the Laplacian of a stochastic matrix. It shows that systematically, our algorithm leads to the best speed[1]. We note that the class of graph depicted in the Tab. 2 were used as worst case examples of [34, 17]. The next section implements and validates our ideas.

4.2 Algorithm

We now describe the algorithm used to implement the dynamics of Eq. (7), and in particular our simulator of P.P.P.. Let us write $T_1^{(i)} < T_2^{(i)} < ... < T_k^{(i)} < ...$ the time of the k-th event on the i-th node, which is either an edge activation, either a gradient update. We remind that the spiking times of a specific event

[1]For the case 2-grid, logarithmic term should appear yet we decided to neglect them.

Table 2: Complexity for various graphs using a stochastic matrix. For a star graph, $\chi_1^* = \mathcal{O}(1)$ and $\gamma^* = \mathcal{O}(n)$; for a line (or cyclic) graph, $\chi_1^* = \mathcal{O}(n^2)$, $\gamma^* = \mathcal{O}(n^2)$, $\chi_2^* = \mathcal{O}(1)$; or the full (complete) graph, $\chi_1^* = \mathcal{O}(1)$ and $\gamma^* = \mathcal{O}(1)$; For the d-dimensional grid, $\chi_1^* = \mathcal{O}(n^{2/d})$ and $\gamma^* = \mathcal{O}(n^{2/d})$, $\chi_2^* = \mathcal{O}(1)$.

Method	# edges activated				# total gradient iterations			
Graph	Star	Line	Complete	d-grid	Star	Line	Complete	d-grid
[17] ADOM+	n	n^3	n^2	$n^{1+2/d}$	n	n	n	n
[34] MSDA	$n^{3/2}$	n^2	n^2	$n^{1+1/d}$	n	n	n	n
[9] Continuized	n	n^2	n	$n^{1+1/d}$	n	n^2	n	$n^{1+1/d}$
Centralized	n	-	-	-	n	-	-	-
Ours	n	n^2	n	$n^{1+1/d}$	n	n	n	n

corresponds to random variables with independent exponential increments and can thus be generated at the beginning of our simulation. They can also be generated on the fly and locally to stress the locality and asynchronicity of our algorithm. Let's write $X_t = (X_t^{(i)})_i$ and $Y_t = (Y_t^{(i)})_i$, then on the i-th node and at the k-th iteration, we integrate the linear Ordinary Differential Equation (ODE) on $[T_k^{(i)}; T_{k+1}^{(i)}]$:

$$\begin{cases} dX_t &= a_1(X_t, Y_t)dt \\ dY_t &= a_2(X_t, Y_t)dt\,, \end{cases}$$

in order to define the values right before the spike, for $\mathcal{A}$ the corresponding constant matrix, we thus have:

$$\begin{pmatrix} X^{(i)}_{T^{(i)-}_{k+1}} \\ Y^{(i)}_{T^{(i)-}_{k+1}} \end{pmatrix} = \exp\left((T^{(i)}_{k+1} - T^{(i)}_k)\mathcal{A}\right) \begin{pmatrix} X^{(i)}_{T^{(i)}_k} \\ Y^{(i)}_{T^{(i)}_k} \end{pmatrix}. \tag{14}$$

Next, if one has a gradient update, then:

$$X^{(i)}_{T^{(i)}_{k+1}} = X^{(i)}_{T^{(i)-}_{k+1}} + b_1\left(X^{(i)}_{T^{(i)-}_{k+1}}\right).$$

Otherwise, if the edge (i,j) or (j,i) is activated, a communication bridge is created between both nodes i and j. In this case, the local update on i writes:

$$Y^{(i)}_{T^{(i)}_{k+1}} = Y^{(i)}_{T^{(i)-}_{k+1}} + b_2\left(Y^{(i)}_{T^{(i)-}_{k+1}}, Y^{(j)}_{T^{(i)-}_{k+1}}\right).$$

Note that, even if this event takes place along an edge (i,j), we can write it separately for nodes i and j by making sure they both have the events $T^{(i)}_{k_i} = T^{(j)}_{k_j}$, for some $k_i, k_j \in \mathbb{N}$, corresponding to this communication. As advocated, all those operations are local and we summarize in the Alg. 1 the algorithmical block which corresponds to our implementation. See Appendix D for more details on our implementation.

Algorithm 1: This algorithm block describes our implementation on each local-machine. The ODE routine is described by Eq. 14 and Ping is an instantaneous routine.

Input: On each machine $i \in \{1, ..., n\}$, an oracle able to evaluate ∇f_i, Parameters $\mu, L, \chi_1^*, t_{\max}$.

Initialize on each machine $i \in \{1, ..., n\}$:
  Set $X^{(i)}, Y^{(i)}$ to 0 ;
  Set $\mathcal{A}$;
  $T^{(i)} \leftarrow 0$;
Synchronize the clocks of all machines ;
In parallel *on workers* $i \in \{1, ..., n\}$, **while** $t < t_{\max}$, **continuously do:**
  $t \leftarrow clock()$;
  Ping surrounding machines and adjust $\lambda_{ij}(t)$;
  if *there is an event at time* t **then**
    $(X^{(i)}, Y^{(i)}) \leftarrow ODE(\mathcal{A}, t - T^{(i)}, (X^{(i)}, Y^{(i)}))$;
    if *the event is to take a gradient step* **then**
      $X^{(i)} \leftarrow X^{(i)} + b_1(X^{(i)})$;
    else if *the event is to communicate with* j **then**
      $Y^{(i)} \leftarrow Y^{(i)} + b_2(Y^{(i)}, Y^{(j)})$; `// Happens at` j `simultaneously.`
    $T^{(i)} \leftarrow t$;
return $x_{t_{\max}}^{(i)}$, *the estimate of* x^* *on each worker* i.

4.3 Numerical results

In this section, we study the behaviour of our method and compare to several settings inspired by [17, 9]. In our experiments, we perform the empirical risk minimization for both the decentralized linear and logistic regression tasks given either by:

$$f_i(x) = \frac{1}{m} \sum_{j=1}^{m} \log(1 + \exp(-b_{ij} a_{ij}^\top x)) + \frac{\mu}{2} \|x\|^2 , \tag{15}$$

or

$$f_i(x) = \frac{1}{m} \sum_{j=1}^{m} \|a_{ij}^\top x - c_{ij}\|^2, \tag{16}$$

where $a_{ij} \in \mathbb{R}^d$, $b_{ij} \in \{-1, 1\}$ and $c_{ij} \in \mathbb{R}$ correspond to m local data points stored at node i. For both varying and fixed topology settings, we follow a protocol similar to [17]: we generate n independent synthetic datasets with the `make_classification` and `make_regression` functions of scikit-learn [30], each worker storing $m = 100$ data points. We recall that the metrics of interest are the total number of local gradient steps and total number of individual messages exchanged *(*i.e.*, number of edges that fired)* to reach an ϵ-precision. We systematically used the proposed hyper-parameters of each reference paper for our implementation without any specific fine-tuning.

Comparison in the time-varying setting. We compare our method to ADOM+ [17] on a sequence of 50 random geometric graphs of size $n = 20$ in Fig. 1. To construct the graphs, we sample uniformly n points in $[0, 1]^2 \subset \mathbb{R}^2$ and connect each of them to all at a distance less than some user-specified radius, which allows to control the constant χ_1^* (we consider values in $\{3, 33, 180, 233\}$). We ensure the connectedness of the graphs by arbitrarily ordering their connected components and linking one to the next via an edge between two randomly selected nodes in each, exactly as done in [18]. We then use the instantaneous gossip matrix introduced in Eq. (12) with $f = \chi_1^*$. We compare ourselves to both versions of ADOM+: with and without the Multi-Consensus (M.-C.). Thanks to its M.-C. procedure, ADOM+ can significantly reduce the number of necessary gradient steps. However, consistently with our analysis in Sec. 4.1, our method is systematically better in all settings in terms of communications.

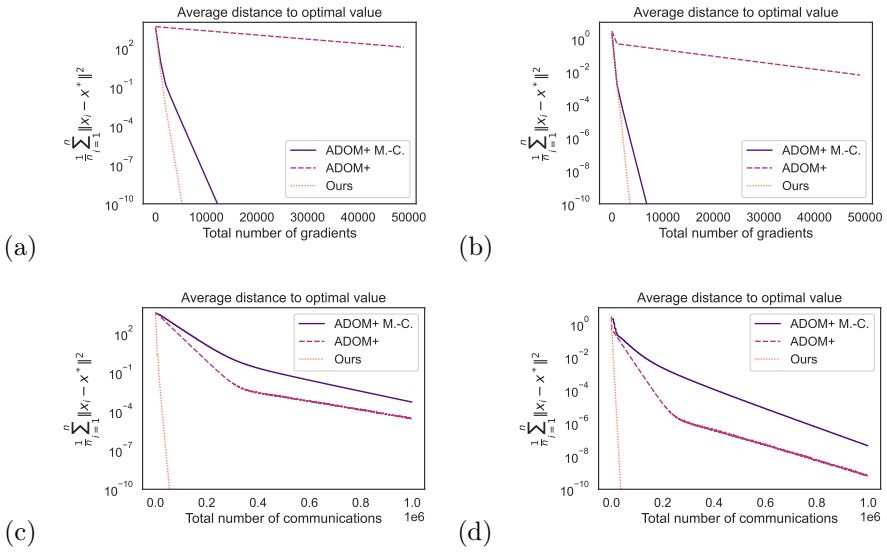

Figure 1: Comparison between ADOM+ [17] and DADAO, using the same data *((a,c) linear regression, (b,d) binary classification)* and the same sequence of random connected graphs with $\chi_1^* = 180$ linking $n = 20$ workers.

Comparison with accelerated methods in the fixed topology setting. Now, we fix the Laplacian matrix via Eq. (12) to compare simultaneously to the continuized framework [9] and ADOM+ [17]. We reports in Fig. 2 results corresponding to the complete graph with $n = 250$ nodes and the line graph of size $n = 150$. While sharing the same asymptotic rate, we note that the Continuized framework [9] and MSDA [34] have better absolute constants than DADAO, giving them an advantage both in terms of number of communica-

tion and gradients. However, in the continuized framework, the gradient and communication steps being coupled, the number of gradient computations can potentially be orders of magnitude worse than our algorithm, which is reflected by Fig. 2.b for the line graph. As for MSDA and ADOM+, Tab. 2 showed they do not have the best communication rates on certain classes of graphs, which is indeed confirmed in Fig. 2.c for MSDA and both communication plots for ADOM+.

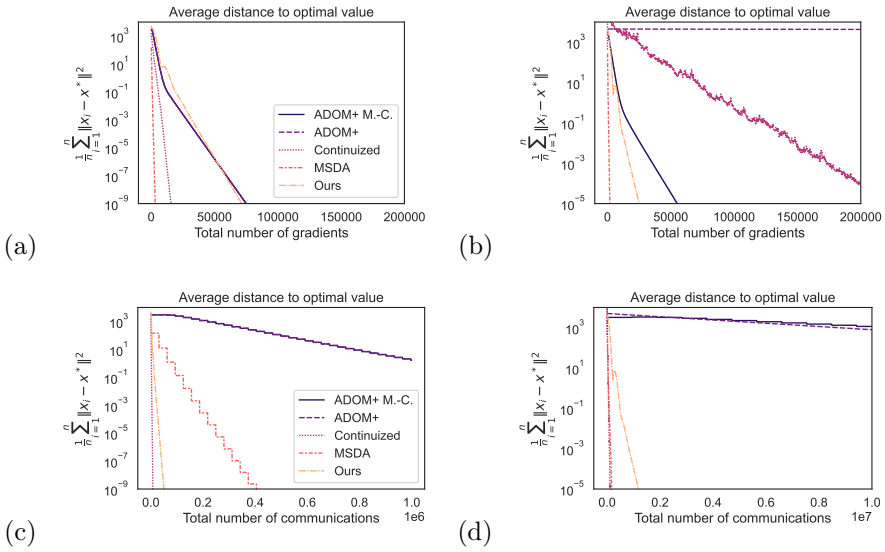

Figure 2: Comparison between ADOM+ [17], the continuized framework [9], MSDA [34] and DADAO, using the same data for the linear regression task, and the same graphs *((a,c) complete with* $n = 250$*, (b,d) line with* $n = 150$*).*

In conclusion, while several methods can share similar rates of convergence, ours is the only one to perform at least as well as its concurrent in every settings, for different graph's topology and 2 different tasks, as predicted by Tab. 1.

5 Conclusion

In this work, we have proposed a novel stochastic algorithm for the decentralized optimization of a sum of smooth and strongly convex functions. We have demonstrated, both theoretically and empirically, that this algorithm leads systematically to a substantial acceleration when compared to state-of-the-art works. Our algorithm is asynchronous, decoupled, primal and does not relies on an extra inner-loop, while being amenable to varying topology settings: each of those properties make it suitable for real applications.

In a future work, we would like to explore the robustness of such algorithm to more challenging variabilities occurring in real-life applications.

Acknowledgements

This work was supported by the Project ANR-21-CE23-0030 ADONIS and EMERG-ADONIS from Alliance SU. The authors would like to thank Mathieu Even, Hadrien Hendrikx and Dmitry Kovalev for helpful discussions.

References

[1] Ludwig Arnold. Stochastic differential equations. *New York*, 1974.

[2] Eugene Belilovsky, Michael Eickenberg, and Edouard Oyallon. Decoupled greedy learning of CNNs. 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 736–745. PMLR, 13–18 Jul 2020.

[3] Eugene Belilovsky, Louis Leconte, Lucas Caccia, Michael Eickenberg, and Edouard Oyallon. Decoupled greedy learning of cnns for synchronous and asynchronous distributed learning. *arXiv preprint arXiv:2106.06401*, 2021.

[4] Léon Bottou. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*, pages 177–186. Springer, 2010.

[5] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah. Randomized gossip algorithms. *IEEE Transactions on Information Theory*, 52(6):2508–2530, 2006.

[6] Laurent Condat, Daichi Kitahara, Andrés Contreras, and Akira Hirabayashi. Proximal splitting algorithms for convex optimization: A tour of recent advances, with new twists, 2019.

[7] Laurent Condat, Grigory Malinovsky, and Peter Richtárik. Distributed proximal splitting algorithms with rates and acceleration. *Frontiers in Signal Processing*, page 12, 2022.

[8] Wendy Ellens, Floske M Spieksma, Piet Van Mieghem, Almerima Jamakovic, and Robert E Kooij. Effective graph resistance. *Linear algebra and its applications*, 435(10):2491–2506, 2011.

[9] Mathieu Even, Raphaël Berthier, Francis Bach, Nicolas Flammarion, Hadrien Hendrikx, Pierre Gaillard, Laurent Massoulié, and Adrien Taylor. A continuized view on nesterov acceleration for stochastic gradient descent and randomized gossip. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021.

[10] Mathieu Even, Hadrien Hendrikx, and Laurent Massoulie. Decentralized optimization with heterogeneous delays: a continuous-time approach. *arXiv preprint arXiv:2106.03585*, 2021.

[11] Mathieu Even, Hadrien Hendrikx, and Laurent Massoulié. Asynchrony and acceleration in gossip algorithms, 2020.

[12] Hadrien Hendrikx. A principled framework for the design and analysis of token algorithms, 2022.

[13] Hadrien Hendrikx, Francis Bach, and Laurent Massoulié. An accelerated decentralized stochastic proximal algorithm for finite sums. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[14] Hadrien Hendrikx, Francis Bach, and Laurent Massoulié. An optimal algorithm for decentralized finite-sum optimization. *SIAM Journal on Optimization*, 31(4):2753–2783, 2021.

[15] Anastasia Koloskova, Nicolas Loizou, Sadra Boreiri, Martin Jaggi, and Sebastian Stich. A unified theory of decentralized sgd with changing topology and local updates. In *International Conference on Machine Learning*, pages 5381–5393. PMLR, 2020.

[16] Anastasiia Koloskova, Tao Lin, and Sebastian U Stich. An improved analysis of gradient tracking for decentralized machine learning. *Advances in Neural Information Processing Systems*, 34:11422–11435, 2021.

[17] Dmitry Kovalev, Elnur Gasanov, Alexander Gasnikov, and Peter Richtárik. Lower bounds and optimal algorithms for smooth and strongly convex decentralized optimization over time-varying networks. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021.

[18] Dmitry Kovalev, Alexander Gasnikov, and Peter Richtárik. Accelerated primal-dual gradient method for smooth and convex-concave saddle-point problems with bilinear coupling. *arXiv preprint arXiv:2112.15199*, 2021.

[19] Dmitry Kovalev, Egor Shulgin, Peter Richtárik, Alexander V Rogozin, and Alexander Gasnikov. Adom: accelerated decentralized optimization method for time-varying networks. In *International Conference on Machine Learning*, pages 5784–5793. PMLR, 2021.

[20] Günter Last and Mathew Penrose. *Lectures on the Poisson process*, volume 7. Cambridge University Press, 2017.

[21] Jonas Latz. Analysis of stochastic gradient descent in continuous time. *Statistics and Computing*, 31(4):1–25, 2021.

[22] Rémi Leblond, Fabian Pedregosa, and Simon Lacoste-Julien. Improved asynchronous parallel optimization analysis for stochastic incremental methods. *arXiv preprint arXiv:1801.03749*, 2018.

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

[24] Hongzhou Lin, Julien Mairal, and Zaid Harchaoui. A universal catalyst for first-order optimization. *Advances in neural information processing systems*, 28, 2015.

[25] Konstantin Mishchenko, Grigory Malinovsky, Sebastian Stich, and Peter Richtárik. Proxskip: Yes! local gradient steps provably lead to communication acceleration! finally! *arXiv preprint arXiv:2202.09357*, 2022.

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

[27] Yurii Nesterov. *Introductory lectures on convex optimization: A basic course*, volume 87. Springer Science & Business Media, 2003.

[28] Yurii Nesterov and Sebastian U Stich. Efficiency of the accelerated coordinate descent method on structured optimization problems. *SIAM Journal on Optimization*, 27(1):110–123, 2017.

[29] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[30] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[31] Adil Salim, Laurent Condat, Dmitry Kovalev, and Peter Richtárik. An optimal algorithm for strongly convex minimization under affine constraints. *arXiv preprint arXiv:2102.11079*, 2021.

[32] Adil Salim, Laurent Condat, Dmitry Kovalev, and Peter Richtárik. An optimal algorithm for strongly convex minimization under affine constraints.

In *International Conference on Artificial Intelligence and Statistics*, pages 4482–4498. PMLR, 2022.

[33] Adil Salim, Laurent Condat, Konstantin Mishchenko, and Peter Richtárik. Dualize, split, randomize: Fast nonsmooth optimization algorithms. *arXiv preprint arXiv:2004.02635*, 2020.

[34] Kevin Scaman, Francis Bach, Sébastien Bubeck, Yin Tat Lee, and Laurent Massoulié. Optimal algorithms for smooth and strongly convex distributed optimization in networks. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 3027–3036. PMLR, 06–11 Aug 2017.

[35] Sebastian U. Stich and Sai Praneeth Karimireddy. The error-feedback framework: Sgd with delayed gradients. *Journal of Machine Learning Research*, 21(237):1–36, 2020.

A Proof of the theorem

A.1 Properties and assumptions

The following properties will be used all along the proofs of the Lemma and Theorems and are totally related to the communication of our nodes.

Lemma A.1. *Under the assumptions of Theorem 3.2, if $z_0, \tilde{z}_0 \in span(\pi)$, then $z_t, \tilde{z}_t \in span(\pi)$ almost surely.*

Proof. It's clear that for any i, j, we get:

$$\pi(\mathbf{e}_i - \mathbf{e}_j)(\mathbf{e}_i - \mathbf{e}_j)^\mathsf{T} = (\mathbf{e}_i - \mathbf{e}_j)(\mathbf{e}_i - \mathbf{e}_j)^\mathsf{T}.$$

Thus, the variations of $(z_t, \tilde{z}_t)$ belong to $\mathrm{span}(\pi)$, and thus so is the trajectory. □

We derive the following Lemma, similar to a result from [5]:

Lemma A.2 (Spiking contraction)**.** *Under the assumptions of Theorem 3.2, we have:*

$$\sum_{(i,j)\in\mathcal{E}(t)} \lambda_{ij}(t)\left[\|(\mathbf{e}_i-\mathbf{e}_j)(\mathbf{e}_i-\mathbf{e}_j)^\mathsf{T}x-\pi x\|^2-\|\pi x\|^2\right] = -x^\mathsf{T}\mathbf{\Lambda}(t)x \leq -\frac{1}{\chi_1^*}\|\pi x\|^2.$$

Proof. If $i = j$, then $\lambda_{ii} = 0$. For a given (i, j), we get if $i \neq j$:

$$\begin{aligned}\|(\mathbf{e}_i - \mathbf{e}_j)(\mathbf{e}_i - \mathbf{e}_j)^\mathsf{T}x - \pi x\|^2 &= \|\pi x\|^2 + \|x_i - x_j\|^2 && (17)\\ &\quad - 2\langle \pi(x), (\mathbf{e}_i - \mathbf{e}_j)(\mathbf{e}_i - \mathbf{e}_j)^\mathsf{T}x\rangle \\ &= \|\pi(x)\|^2 - \langle x, (\mathbf{e}_i - \mathbf{e}_j)(\mathbf{e}_i - \mathbf{e}_j)^\mathsf{T}x\rangle. && (18)\end{aligned}$$

And this allows to conclude by sum. □

Lemma A.3 (Resistance)**.** *For i, j and any $x \in \mathbb{R}^d$, we have:*

$$\|(\mathbf{e}_i - \mathbf{e}_j)(\mathbf{e}_i - \mathbf{e}_j)^\mathsf{T}x\|^2_{\mathbf{\Lambda}(t)^+} \leq \chi_2^*\|(\mathbf{e}_i - \mathbf{e}_j)(\mathbf{e}_i - \mathbf{e}_j)^\mathsf{T}x\|$$

Proof. Indeed, we note that:

$$\begin{aligned}\|(\mathbf{e}_i - \mathbf{e}_j)(\mathbf{e}_i - \mathbf{e}_j)^\mathsf{T}x\|^2_{\mathbf{\Lambda}(t)^+} &= x^\mathsf{T}(\mathbf{e}_i - \mathbf{e}_j)(\mathbf{e}_i - \mathbf{e}_j)^\mathsf{T}\mathbf{\Lambda}(t)^+(\mathbf{e}_i - \mathbf{e}_j)(\mathbf{e}_i - \mathbf{e}_j)^\mathsf{T}x && (19)\\ &\leq 2\chi_2^* x^\mathsf{T}(\mathbf{e}_i - \mathbf{e}_j)(\mathbf{e}_i - \mathbf{e}_j)^\mathsf{T}x && (20)\\ &= \chi_2^*\|(\mathbf{e}_i - \mathbf{e}_j)(\mathbf{e}_i - \mathbf{e}_j)^\mathsf{T}x\|^2 && (21)\end{aligned}$$

□

Next, we set $\nu = \frac{\mu}{2}$ such that:

$$\frac{1}{2L}\|\nabla F(x) - \nabla F(y)\|^2 \leq d_F(x,y) \leq \frac{L}{2}\|x-y\|^2\,,$$

and

$$\frac{\nu}{2}\|x-y\|^2 \leq d_F(x,y) \leq \frac{1}{2\nu}\|\nabla F(x) - \nabla F(y)\|^2\,,$$

and we remind that:

$$\mathbb{E}_\xi d_{F(.,\xi)}(x,y) = d_{\mathbb{E}_\xi F(.,\xi)}(x,y)\,. \tag{22}$$

A.2 Proof of the Lemma 3.3 and Lemma 3.4

We first state a couple of inequalities that we will combine all together in order to obtain a bound on our Lyapunov function.

Proposition A.4. *First:*

$$\begin{aligned}
\phi_A \triangleq\ & A_t(d_F(x^+,x^*) - d_F(x,x^*)) + \tilde{A}_t(\|\tilde{x}^+ - x^*\|^2 - \|\tilde{x}-x^*\|^2) \\
& + \eta A_t\langle \tilde{x}-x, F(x) - \nabla F(x^*)\rangle + 2\tilde{\eta}\tilde{A}_t\langle x-\tilde{x}, \tilde{x}-x^*\rangle \qquad (23)\\
& \leq \|\nabla F(x) - \tilde{y}\|^2\left(A_t\frac{L\gamma^2}{2} - A_t\gamma + \tilde{A}_t\tilde{\gamma}^2\right) \\
& + A_t\gamma\langle \nabla F(x) - \tilde{y}, y^* - \tilde{y}\rangle + 2\tilde{\gamma}\tilde{A}_t\langle \tilde{y}-y^*, \tilde{x}-x^*\rangle \qquad (24)\\
& - 2\tilde{\gamma}\tilde{A}_t\left(d_F(\tilde{x},x^*) + d_F(x^*,x) - d_F(\tilde{x},x)\right) \\
& - \eta A_t(d_F(\tilde{x},x) + d_F(x,x^*) - d_F(\tilde{x},x^*)) - \tilde{A}_t\tilde{\eta}\|\tilde{x}-x^*\|^2 + \tilde{\eta}\|x-x^*\|^2
\end{aligned}$$

Proof. First, we have, using optimality conditions and smoothness:

$$\begin{aligned}
d_F(x^+,x^*) - d_F(x,x^*) &= d_F(x^+,x) - \langle x^+ - x, \nabla F(x^*) - \nabla F(x)\rangle && (25)\\
&\leq \frac{L}{2}\|x^+ - x\|^2 - \langle x^+ - x, \nabla F(x^*) - \nabla F(x)\rangle && (26)\\
&= \frac{L\gamma^2}{2}\|\tilde{y} - \nabla F(x)\|^2 - \gamma\|\nabla F(x) - \tilde{y}\|^2 \\
&\quad + \gamma\langle \nabla F(x) - \tilde{y}, y^* - \tilde{y}\rangle && (27)
\end{aligned}$$

Next, we note that, again using optimality conditions:

$$\begin{aligned}
\|\tilde{x}^+ - x^*\|^2 - \|\tilde{x}^+ - x^*\|^2 &= 2\langle \tilde{x}^+ - \tilde{x}, \tilde{x} - x^*\rangle + \|\tilde{x}^+ - \tilde{x}\|^2 && (28)\\
&= -2\tilde{\gamma}\langle \nabla F(x) - \tilde{y}, \tilde{x} - x^*\rangle + \tilde{\gamma}^2\|\nabla F(x) - \tilde{y}\|^2 && (29)\\
&= -2\tilde{\gamma}\langle \nabla F(x) - \nabla F(x^*), \tilde{x} - x^*\rangle \\
&\quad + 2\tilde{\gamma}\langle \tilde{y} - y^*, \tilde{x} - x^*\rangle + \tilde{\gamma}^2\|\nabla F(x) - \tilde{y}\|^2 && (30)\\
&= -2\tilde{\gamma}(d_F(\tilde{x},x^*) + d_F(x^*,x) - d_F(\tilde{x},x)) \\
&\quad + 2\tilde{\gamma}\langle \tilde{y} - y^*, \tilde{x} - x^*\rangle + \tilde{\gamma}^2\|\nabla F(x) - \tilde{y}\|^2 && (31)
\end{aligned}$$

Momentum in x associated with the term $d_F(x, x^*)$ gives:

$$\eta\langle\tilde{x} - x, \nabla F(x) - \nabla F(x^*)\rangle = -\eta(d_F(\tilde{x}, x) + d_F(x, x^*) - d_F(\tilde{x}, x^*)) \tag{32}$$

and momentum in $\tilde{x}$ associated with $\|\tilde{x} - x^*\|^2$ leads to:

$$2\tilde{\eta}\langle x - \tilde{x}, \tilde{x} - x^*\rangle = -2\tilde{\eta}\|\tilde{x} - x^*\|^2 + 2\tilde{\eta}\langle x - x^*, \tilde{x} - x^*\rangle \leq -\tilde{\eta}\|\tilde{x} - x^*\|^2 + \tilde{\eta}\|x - x^*\|^2 \tag{33}$$

□

Corollary A.4.1. *Under Assumption 3.4, we have:*

$$\begin{aligned}\tilde{\phi}_A \triangleq\ & \mathbb{E}_\xi[A_t(d_F(x^+, x^*) - d_F(x, x^*)) + \tilde{A}_t(\|\tilde{x}^+ - x^*\|^2 - \|\tilde{x} - x^*\|^2) \\ & + \eta A_t\langle\tilde{x} - x, F(x) - \nabla F(x^*)\rangle + 2\tilde{\eta}\tilde{A}_t\langle x - \tilde{x}, \tilde{x} - x^*\rangle] \qquad (34)\\ \leq\ & \phi_A + \sigma^2(A_t\frac{L\gamma^2}{2} - A_t\gamma + \tilde{A}_t\tilde{\gamma}) \qquad (35)\end{aligned}$$

Proof. Using exactly the same computations and the Eq. (37), we next note that:

$$\begin{aligned}\mathbb{E}_\xi[\|\nabla F(x, \xi) - y\|^2] &= \mathbb{E}_\xi[\|\nabla F(x, \xi)\|^2 - 2\langle\nabla F(x, \xi), y\rangle + \|y\|^2] \qquad (36)\\ &\leq \|\nabla F(x) - y\|^2 + \sigma^2 \qquad (37)\end{aligned}$$

□

Proposition A.5. *Next, we show that if $\alpha B_t = \frac{\delta}{2}\tilde{B}_t$:*

$$\begin{aligned}\phi_B \triangleq\ & B_t(\|y^+ - y^*\|^2 - \|y - y^*\|^2) + \tilde{B}_t(\|\tilde{y}^+ - y^*\|^2 - \|\tilde{y} - y^*\|^2) \\ & + 2\alpha B_t\langle y - y^*, \tilde{y} - y\rangle - 2\theta\tilde{B}_t\langle y + z + \nu\tilde{x}, \tilde{y} - y^*\rangle \\ & + 2\alpha C_t\langle\tilde{y} - y, z + y - y^* - z^*\rangle \qquad (38)\\ =\ & -\frac{\delta}{2}\tilde{B}_t\|\tilde{y} - y^*\|^2 - \frac{\delta}{2}\tilde{B}_t\|y - y^*\|^2 - 2\tilde{\delta}\tilde{B}_t\langle\nabla F(x) - \tilde{y}, y^* - \tilde{y}\rangle \\ & + \delta\tilde{B}_t\|\nabla F(x) - \nabla F(x^*)\|^2 + \left((\delta + \tilde{\delta})^2 - \delta\right)\tilde{B}_t\|\nabla F(x) - y\|^2 \\ & - 2\theta\tilde{B}_t\langle y + z - y^* - z^*, \tilde{y} - y^*\rangle - 2\theta\nu\tilde{B}_t\langle\tilde{x} - x^*, \tilde{y} - y^*\rangle \\ & + 2\alpha C_t\langle\tilde{y} - y, z + y - y^* - z^*\rangle \qquad (39)\end{aligned}$$

Proof.

$$\begin{aligned}\|\tilde{y}^+ - y^*\|^2 - \|\tilde{y} - y^*\|^2 &= 2\langle\tilde{y} - y^*, \tilde{y}^+ - \tilde{y}\rangle + \|\tilde{y}^+ - \tilde{y}\|^2 \qquad (40)\\ &= 2\delta\langle\nabla F(x) - \tilde{y}, \tilde{y} - y^*\rangle + 2\tilde{\delta}\langle\nabla F(x) - \tilde{y}, \tilde{y} - y^*\rangle \\ &\quad (\delta + \tilde{\delta})^2\|\nabla F(x) - \tilde{y}\|^2 \qquad (41)\\ &= -2\tilde{\delta}\langle\nabla F(x) - \tilde{y}, y^* - \tilde{y}\rangle \\ &\quad + \delta\|\nabla F(x) - \nabla F(x^*)\|^2 - \delta\|\tilde{y} - y^*\|^2 \\ &\quad \left((\delta + \tilde{\delta})^2 - \delta\right)\|\nabla F(x) - \tilde{y}\|^2 \qquad (42)\end{aligned}$$

The momentum in $\tilde{y}$ associated with the term $\|\tilde{y} - y^*\|^2$ gives:

$$\begin{aligned} -2\theta\tilde{B}_t\langle y + z + \nu\tilde{x}, \tilde{y} - y^*\rangle = &- 2\theta\tilde{B}_t\langle y + z - y^* - z^*, \tilde{y} - y^*\rangle \\ &- 2\theta\nu\tilde{B}_t\langle \tilde{x} - x^*, \tilde{y} - y^*\rangle \end{aligned} \tag{43}$$

The momentum in y associated with the term $\|y - y^*\|^2$ gives:

$$2\alpha B_t\langle \tilde{y} - y, y - y^*\rangle = -\alpha B_t\|y - y^*\|^2 - \alpha B_t\|\tilde{y} - y\|^2 + \alpha B_t\|\tilde{y} - y^*\|^2 \tag{44}$$

and the one associated with $\|y + z - y^* - z^*\|^2$:

$$2\alpha C_t\langle \tilde{y} - y, z + y - y^* - z^*\rangle \tag{45}$$

□

Corollary A.5.1. *Under Assumption 3.4, we have:*

$$\begin{aligned} \tilde{\phi}_B \triangleq\ & \mathbb{E}_\xi[\tilde{B}_t(\|y^+ - y^*\|^2 - \|y - y^*\|^2) + \tilde{B}_t(\|\tilde{y}^+ - y^*\|^2 - \|\tilde{y} - y^*\|^2) \\ &+ 2\alpha B_t\langle y - y^*, \tilde{y} - y\rangle - 2\theta\tilde{B}_t\langle y + z + \nu\tilde{x}, \tilde{y} - y^*\rangle] \qquad (46) \\ \leq\ & \phi_B + \sigma^2((\delta^2 + (\delta + \tilde{\delta})^2)\tilde{B}_t) \qquad (47) \end{aligned}$$

Proof. Exactly as above. □

Proposition A.6. *Finally, assuming $\theta\tilde{B}_t = \tilde{\beta}\tilde{C}_t = \alpha C_t$, letting $1 \geq \tilde{\tau} > 0$, $z_{ij}^+ = \beta(\mathbf{e}_i - \mathbf{e}_j)(\mathbf{e}_i - \mathbf{e}_j)^\mathsf{T}(y + z)$ and $\tilde{z}_{ij}^+ = \tilde{\beta}(\mathbf{e}_i - \mathbf{e}_j)(\mathbf{e}_i - \mathbf{e}_j)^\mathsf{T}(y + z)$, then:*

$$\begin{aligned} &\phi_C + \phi_D - 2\theta\tilde{B}_t\langle y + z - y^* - z^*, \tilde{y} - y^*\rangle \triangleq \\ &\quad \sum_{ij}\lambda_{ij}(t)C_t\Big(\|y + z_{ij}^+ - y^* - z^*\|^2 - \|y + z - y^* - z^*\|^2\Big) \\ &+ \sum_{ij}\lambda_{ij}(t)\tilde{C}_t\Big(\|\tilde{z}_{ij}^+ - z^*\|^2 - \|\tilde{z} - z^*\|^2\Big) + 2\tilde{\alpha}\tilde{C}_t\langle z - \tilde{z}, \tilde{z} - z^*\rangle_{\mathbf{\Lambda}(t)^+} \qquad (48) \\ &+ 2\alpha C_t\langle \tilde{z} + \tilde{y} - z^* - y^*, z + y - y^* - z^*\rangle - 2\theta\tilde{B}_t\langle y + z - y^* - z^*, \tilde{y} - y^*\rangle \\ &\leq -2\tilde{\beta}\tilde{C}_t\langle \tilde{z} - z^*, \pi(y + z)\rangle + \tilde{\beta}^2\chi_2^*\tilde{C}_t\sum_{(i,j)\in\mathcal{E}(t)}\lambda_{ij}(t)\|(\mathbf{e}_i - \mathbf{e}_j)(\mathbf{e}_i - \mathbf{e}_j)^\mathsf{T}(y + z)\|^2 \\ &- \frac{\beta}{\chi_1^*}C_t\|\pi(y + z)\|^2 + \beta(\beta - 1)C_t\sum_{(i,j)\in\mathcal{E}(t)}\lambda_{ij}(t)\|(\mathbf{e}_i - \mathbf{e}_j)(\mathbf{e}_i - \mathbf{e}_j)^\mathsf{T}(y + z)\|^2 \\ &- \alpha C_t\|y + z - y^* - z^*\|^2 + \tilde{\alpha}\chi_1^*\tilde{C}_t\|z - z^*\|^2 - \tilde{\alpha}\tilde{C}_t\|\tilde{z} - z^*\|_{\mathbf{\Lambda}(t)^+}^2 \\ &- \tilde{\tau}\frac{1}{2}\tilde{\beta}\frac{\nu}{L}\tilde{C}_t\|z - z^*\|^2 + \tilde{\tau}\frac{\nu}{L}\frac{2\alpha\theta}{\delta}B_t\|y - y^*\|^2 \qquad (49) \end{aligned}$$

Proof. Having in mind that $\pi(y^* + z^*) = 0$ and $\mathbf{\Lambda}(t)^+\mathbf{\Lambda}(t) = \pi$, we get, using

Lemma A.1 and Lemma A.3 on the inequality (53):

$$\Delta_{\tilde{z}} \triangleq \sum_{(i,j)\in\mathcal{E}(t)} \lambda_{ij}(t)\big(\|\tilde{z}_{ij}^{+} - z^*\|^2_{\mathbf{\Lambda}(t)^+} - \|\tilde{z} - z^*\|^2_{\mathbf{\Lambda}(t)^+}\big) \tag{50}$$

$$= \sum_{(i,j)\in\mathcal{E}(t)} \lambda_{ij}(t) 2\langle \tilde{z} - z^*, \tilde{z}_{ij}^{+} - \tilde{z}\rangle_{\mathbf{\Lambda}(t)^+} + \|\tilde{z}_{ij}^{+} - \tilde{z}\|^2_{\mathbf{\Lambda}(t)^+} \tag{51}$$

$$= -2\tilde{\beta} \sum_{(i,j)\in\mathcal{E}(t)} \lambda_{ij}(t)\langle \tilde{z} - z^*, (\mathbf{e}_i - \mathbf{e}_j)(\mathbf{e}_i - \mathbf{e}_j)^{\mathsf{T}}(y + z - y^* - z^*)\rangle_{\mathbf{\Lambda}(t)^+}$$
$$+ \sum_{(i,j)\in\mathcal{E}(t)} \lambda_{ij}(t)\tilde{\beta}^2 \|(\mathbf{e}_i - \mathbf{e}_j)(\mathbf{e}_i - \mathbf{e}_j)^{\mathsf{T}}(y + z)\|^2_{\mathbf{\Lambda}(t)^+} \tag{52}$$

$$\leq -2\tilde{\beta}\langle \tilde{z} - z^*, \mathbf{\Lambda}(t)^+\mathbf{\Lambda}(t)(y + z)\rangle$$
$$+ \chi_2^*\tilde{\beta}^2 \sum_{(i,j)\in\mathcal{E}(t)} \lambda_{ij}(t)\|(\mathbf{e}_i - \mathbf{e}_j)(\mathbf{e}_i - \mathbf{e}_j)^{\mathsf{T}}(y + z)\|^2 \tag{53}$$

$$= -2\tilde{\beta}\langle \tilde{z} - z^*, \pi(y + z)\rangle + \chi_2^*\tilde{\beta}^2 \sum_{(i,j)\in\mathcal{E}(t)} \lambda_{ij}(t)\|(\mathbf{e}_i - \mathbf{e}_j)(\mathbf{e}_i - \mathbf{e}_j)^{\mathsf{T}}(y + z)\|^2 \tag{54}$$

We also have, as $y^+ = y$ and using Lemma A.2:

$$\Delta_z \triangleq \sum_{(i,j)\in\mathcal{E}(t)} \lambda_{ij}(t)(\|y^+ + z_{ij}^{+} - y^* - z^*\|^2 - \|y + z - y^* - z^*\|^2) \tag{55}$$

$$= 2 \sum_{(i,j)\in\mathcal{E}(t)} \lambda_{ij}(t)\langle y + z_{ij}^{+} - y - z, y + z - y^* - z^*\rangle$$
$$+ \sum_{(i,j)\in\mathcal{E}(t)} \lambda_{ij}(t)\|y + z_{ij}^{+} - y - z\|^2 \tag{56}$$

$$= -2 \sum_{(i,j)\in\mathcal{E}(t)} \beta\lambda_{ij}(t)\langle (\mathbf{e}_i - \mathbf{e}_j)(\mathbf{e}_i - \mathbf{e}_j)^{\mathsf{T}}(y + z), y + z - y^* - z^*\rangle$$
$$+ \sum_{(i,j)\in\mathcal{E}(t)} \beta^2\lambda_{ij}(t)\|(\mathbf{e}_i - \mathbf{e}_j)(\mathbf{e}_i - \mathbf{e}_j)^{\mathsf{T}}(y + z)\|^2 \tag{57}$$

$$= \sum_{(i,j)\in\mathcal{E}(t)} \lambda_{ij}(t)\Big(- \beta\|(\mathbf{e}_i - \mathbf{e}_j)(\mathbf{e}_i - \mathbf{e}_j)^{\mathsf{T}}(y + z)\|^2 - \beta\|\pi(y + z)\|^2$$
$$+ \beta\|(\mathbf{e}_i - \mathbf{e}_j)(\mathbf{e}_i - \mathbf{e}_j)^{\mathsf{T}}(y + z) - \pi(y + z)\|^2$$
$$+ \beta^2\|(\mathbf{e}_i - \mathbf{e}_j)(\mathbf{e}_i - \mathbf{e}_j)^{\mathsf{T}}(y + z)\|^2\Big) \tag{58}$$

$$\leq -\frac{\beta}{\chi_1^*}\|\pi(y + z)\|^2 + \beta(\beta - 1) \sum_{(i,j)\in\mathcal{E}(t)} \lambda_{ij}(t)\|(\mathbf{e}_i - \mathbf{e}_j)(\mathbf{e}_i - \mathbf{e}_j)^{\mathsf{T}}(y + z)\|^2 \tag{59}$$

The momentum in $\tilde{z}$ associated with $\|\tilde{z} - z^*\|^2_{\mathbf{\Lambda}(t)^+}$ gives:

$$2\tilde{\alpha}\tilde{C}_t\langle z - \tilde{z}, \tilde{z} - z^*\rangle_{\mathbf{\Lambda}(t)^+} \leq \tilde{\alpha}\chi_1^*\tilde{C}_t\|z - z^*\|^2 - \tilde{\alpha}\tilde{C}_t\|\tilde{z} - z^*\|^2_{\mathbf{\Lambda}(t)^+} \tag{60}$$

And the one in z associated with $\|y+z-y^*-z^*\|^2$ gives:

$$2\alpha C_t\langle \tilde{z}-z, z+y-y^*-z^*\rangle \tag{61}$$

Then, assuming that $\theta\tilde{B}_t = \tilde{\beta}\tilde{C}_t = \alpha C_t$, we have:

$$\begin{aligned}
&2\alpha C_t\langle \tilde{y}-y, z+y-y^*-z^*\rangle - 2\tilde{\beta}\tilde{C}_t\langle \tilde{z}-z^*, y+z-y^*-z^*\rangle \\
&- 2\theta\tilde{B}_t\langle y+z-y^*-z^*, \tilde{y}-y^*\rangle + 2\alpha C_t\langle \tilde{z}-z, z+y-y^*-z^*\rangle &(62)\\
&= -2\alpha C_t\|y+z-y^*-z^*\|^2 &(63)
\end{aligned}$$

At this stage, we split the negative term (63) in two halves, upper-bounding one of the halves by remembering that $\frac{\nu}{L} \leq 1$ and introducing $1 \geq \tilde{\tau} > 0$:

$$\begin{aligned}
-\alpha C_t\|y+z-y^*-z^*\|^2 \leq& -\tilde{\tau}\frac{\nu}{L}\alpha C_t\|y+z-y^*-z^*\|^2 &(64)\\
=& -\tilde{\tau}\tilde{\beta}\frac{\nu}{L}\tilde{C}_t\|y+z-y^*-z^*\|^2 &(65)\\
\leq& -\tilde{\tau}\frac{1}{2}\tilde{\beta}\frac{\nu}{L}\tilde{C}_t\|z-z^*\|^2 + \tilde{\tau}\tilde{\beta}\frac{\nu}{L}\tilde{C}_t\|y-y^*\|^2 &(66)\\
=& -\tilde{\tau}\frac{1}{2}\tilde{\beta}\frac{\nu}{L}\tilde{C}_t\|z-z^*\|^2 + \tilde{\tau}\frac{\nu}{L}\frac{2\alpha\theta}{\delta}B_t\|y-y^*\|^2 &(67)
\end{aligned}$$

□

Keeping in mind that $\theta\tilde{B}_t = \tilde{\beta}\tilde{C}_t = \alpha C_t$ and $\frac{\delta}{2}\tilde{B}_t = \alpha B_t$, we put everything together. Defining $\Psi = \phi_A + \phi_B + \phi_C + \phi_D$, we have:

$$\Psi \leq \|\nabla F(x) - \tilde{y}\|^2 \left(A_t \frac{L\gamma^2}{2} - A_t\gamma + \tilde{A}_t\tilde{\gamma}^2 + \left((\delta+\tilde{\delta})^2 - \delta\right)\tilde{B}_t \right) \tag{68}$$

$$+ \|\tilde{z} - z^*\|^2_{\mathbf{\Lambda}(t)^+} \left(-\tilde{\alpha}\tilde{C}_t + \tilde{C}'_t \right) \tag{69}$$

$$+ \|\tilde{y} - y^*\|^2 (\tilde{B}'_t - \frac{\delta}{2}\tilde{B}_t) \tag{70}$$

$$+ \|x - x^*\|^2 (\tilde{A}_t\tilde{\eta} - \tilde{A}_t\frac{\nu\tilde{\gamma}}{2}) \tag{71}$$

$$+ \|\tilde{x} - x^*\|^2 (\tilde{A}'_t - \tilde{A}_t\tilde{\eta}) \tag{72}$$

$$+ \|\nabla F(x) - \nabla F(x^*)\|^2 (\delta\tilde{B}_t - \frac{\tilde{\gamma}}{2L}\tilde{A}_t) \tag{73}$$

$$+ \|\pi(y+z) - \pi(y^*+z^*)\|^2 (-\frac{\beta}{\chi_1^*}C_t) \tag{74}$$

$$+ \sum_{(i,j)\in\mathcal{E}(t)} \lambda_{ij}(t) \|(\mathbf{e}_i - \mathbf{e}_j)(\mathbf{e}_i - \mathbf{e}_j)^\mathsf{T}(y+z)\|^2 \left(\chi_2^*\tilde{\beta}^2\tilde{C}_t + \beta(\beta-1)C_t \right) \tag{75}$$

$$+ \|z - z^*\|^2 (\chi_1^*\tilde{\alpha} - \tilde{\tau}\frac{1}{2}\tilde{\beta}\frac{\nu}{L})\tilde{C}_t \tag{76}$$

$$+ \|y - y^*\| (B'_t - (1 - \tilde{\tau}\frac{\nu}{L}\frac{2\theta}{\delta})\alpha B_t) \tag{77}$$

$$+ \|y + z - y^* - z^*\|^2 (C'_t - \alpha C_t) \tag{78}$$

$$+ d_F(x, x^*)(A'_t - \eta A_t) \tag{79}$$

$$+ d_F(\tilde{x}, x)(-A_t\eta + 2\tilde{\gamma}\tilde{A}_t) \tag{80}$$

$$+ d_F(\tilde{x}, x^*)(A_t\eta - 2\tilde{\gamma}\tilde{A}_t) \tag{81}$$

$$+ \langle \nabla F(x) - \tilde{y}, y^* - \tilde{y} \rangle (-2\tilde{\delta}\tilde{B}_t + \gamma A_t) \tag{82}$$

$$+ \langle \tilde{y} - y^*, \tilde{x} - x^* \rangle \left(2\tilde{\gamma}\tilde{A}_t - 2\theta\nu\tilde{B}_t \right) \tag{83}$$

Resolution GD

Proof of Lemma 3.3. Our goal is to put to zero all of the terms appearing next to scalar products, and make the factors of positive quantities (norms or divergences) be less or equal to zero. Given our relations, we guess that each exponential has the same rate. Thus, with $\tau > 0$, we fix $\frac{\delta}{2} = \tilde{\eta} = \eta = \tilde{\alpha} = \tau\sqrt{\frac{\nu}{L}}$, which leads to $\tilde{\gamma} = \frac{2\tau}{\sqrt{\nu L}}$ using Eq. (71). Also, from Eq. (81):

$$4\tilde{A}_t = \nu A_t.$$

Next, from Eq. (73) and Eq. (83), it's necessary that:

$$2L\delta = \theta\nu\,,$$

thus $\theta = 4\tau\sqrt{\frac{L}{\nu}}$. From Eq. (83), we get:

$$\tilde{A}_t = 2L\nu\tilde{B}_t.\,,$$

Combining this previous equation with Eq. (82), as $4\tilde{A}_t = \nu A_t$, we have $\tilde{\delta} = 4L\gamma$. Next, Eq. (68) gives, with the equations above:

$$\begin{aligned} A_t(\frac{L\gamma^2}{2} - \gamma) + \tilde{A}_t\tilde{\gamma}^2 + \left((\delta+\tilde{\delta})^2 - \delta\right)\tilde{B}_t &= A_t\frac{L\gamma^2}{2} - A_t\gamma + \frac{\nu}{4}\tilde{\gamma}^2 A_t \\ &\quad + \left(\delta^2 + \tilde{\delta}^2 + \delta\right)\frac{A_t}{8L} \\ &= A_t\left(\frac{L\gamma^2}{2} - \gamma + \frac{\nu}{4}\frac{4\tau^2}{\nu L}\right) \\ &\quad + A_t(2\tau\sqrt{\frac{\nu}{L}} + 4\tau^2\frac{\nu}{L} + 16L^2\gamma^2)\frac{1}{8L} \\ &\leq A_t(\gamma^2\frac{5}{2}L - \gamma + \frac{5}{4}\frac{\tau^2}{L} + \frac{\sqrt{2}}{8}\frac{\tau}{L}) \end{aligned}$$

We thus pick $\gamma = \frac{1}{4L}$ and $\tau = \frac{1}{8}$, so that $\tilde{\delta} = 1$. Via Eq. (77), we fix $\tilde{\tau} = \frac{1}{8} < 1$. With Eq. (76), we then get:

$$\tilde{\beta} = 2\chi_1^*\sqrt{\frac{L}{\nu}}$$

We also put $\alpha = 2\tau\sqrt{\frac{\nu}{L}}$ and only one last equation, Eq. (75), needs to be satisfied, for which we pick $\beta = \frac{1}{2}$:

$$\chi_2^*\tilde{\beta}^2\tilde{C}_t + \beta(\beta-1)C_t = (\chi_2^*\tilde{\beta}\alpha - \frac{1}{4})C_t$$

This implies that $\chi_2^*\chi_1^* \leq \frac{1}{2}$. Finally, it's clear that all the equations are satisfied if we consider $A_t, \tilde{A}_t, B_t, \tilde{B}_t, C_t, \tilde{C}_t$ as exponentials proportional to $e^{\tau\sqrt{\frac{\nu}{L}}}$. □

Resolution SGD

Proof of Lemma 3.4. All the previous computations clearly hold, except that the term in front of σ^2 is given by:

$$\begin{aligned} (\delta^2 + (\delta+\tilde{\delta})^2)\tilde{B}_t + (A_t\frac{L\gamma^2}{2} - A_t\gamma + \tilde{A}_t\tilde{\gamma}) &= (\delta^2 + (\delta+\tilde{\delta})^2)\frac{A_t}{8L} \\ &\quad + (A_t\frac{L\gamma^2}{2} - A_t\gamma + \nu\frac{A_t}{4}\tilde{\gamma}) \quad (84) \\ &= \mathcal{O}(\frac{A_t}{L}) \quad (85) \end{aligned}$$

□

B Physical interpretation

To gain more insight on the condition $2\chi_1^*[\Lambda]\chi_2^*[\Lambda] \leq 1$, we can write $\Lambda(t)$ as the product of two more interpretable quantities:

$$\Lambda(t) = \underbrace{\sum_{(ij)\in\mathcal{E}(t)} \lambda_{ij}(t)}_{\triangleq\lambda(t)} \underbrace{\frac{2\Lambda(t)}{\text{Tr }\Lambda(t)}}_{\triangleq\tilde{\Lambda}(t)} \tag{86}$$

In this setting, $\lambda(t)$ is the instantaneous expected rate of communication over the whole graph at time t, while $\tilde{\Lambda}(t)$ can be interpreted as the Laplacian of $\mathcal{E}(t)$ weighted with the probabilities of each edge firing between time t and $t+dt$.

Being normalized, $\tilde{\Lambda}(t)$ only contains the information about the graph's connectivity at time t while $\lambda(t)$ is the global rate of communication. We have:

$$\chi_1[\Lambda(t)] = \frac{\chi_1[\tilde{\Lambda}(t)]}{\lambda(t)} \quad ; \quad \chi_2[\Lambda(t)] = \frac{\chi_2[\tilde{\Lambda}(t)]}{\lambda(t)}. \tag{87}$$

If we make the following assumptions,

Assumption B.1. *There is a $\lambda^* > 0$ such that, at all time t, $\lambda(t) \geq \lambda^*$.*

Assumption B.2. *There are $\tilde{\chi}_1^* > 0$, $\tilde{\chi}_2^* > 0$ such that, for all t, $\chi_1[\tilde{\Lambda}(t)] \leq \tilde{\chi}_1^*$ and $\chi_2[\tilde{\Lambda}(t)] \leq \tilde{\chi}_2^*$.*

meaning we assume bounds on the worst rate of communication and on the worst graph connectivity, we immediately have $\chi_1[\Lambda(t)] \leq \frac{\tilde{\chi}_1^*}{\lambda^*}$ and $\chi_2[\Lambda(t)] \leq \frac{\tilde{\chi}_2^*}{\lambda^*}$, leading to $\chi_1^* \leq \frac{\tilde{\chi}_1^*}{\lambda^*}$ and $\chi_2^* \leq \frac{\tilde{\chi}_2^*}{\lambda^*}$. Then, if the following condition on the worst rate of communication is met

$$\sqrt{2\tilde{\chi}_1^*\tilde{\chi}_2^*} \leq \lambda^*, \tag{88}$$

meaning that the instantaneous global rate of communication is always larger than some spectral quantity quantifying the graph's connectivity, it directly implies $2\chi_1^*[\Lambda]\chi_2^*[\Lambda] \leq 1$ and the convergence of our method.

C Comparison with other works

We now explain the results of Sec. 4.1.

C.1 Comparison with ADOM+

Using the notations of [17], we know that gossip matrices satisfy, for $q \in \mathbb{N}$:

$$\|W(q)x - x\|^2 \leq (1 - \frac{1}{\chi})\|x\|^2\,,$$

for some $\chi \geq 1$. It implies that:

$$\text{sp}(W(q)) \subset [1 - \sqrt{1 - \frac{1}{\chi}}, 2]\,,$$

and for χ large enough, $1-\sqrt{1-\frac{1}{\chi}} \approx \frac{1}{2\chi}$. Consequently, up to a renormalization factor, we have $\chi_1^*[W] \approx 2\chi$ and:

$$\mathrm{Tr}(W(q)) \leq 2n\,.$$

C.2 Acceleration of the continuized framework

Under the notation of [9], we note that, an additional simplification holds: $\theta'_{\mathrm{ARG}} = \theta_{\mathrm{ARG}}$. We remind that $\mathcal{L} = AA^T$ and that $Ae_{vw} = \sqrt{P_{vw}}(e_v - e_w)$. Next, we note that by definition:

$$\frac{R_{vw}}{P_{vw}} \triangleq \frac{e_{vw}^T A^+ A e_{vw}}{P_{vw}} \tag{89}$$

$$= \frac{e_{vw}^T A^+ (e_v - e_w)}{\sqrt{P_{vw}}} \tag{90}$$

$$= \frac{(A^{+T} e_{vw})^T (e_v - e_w)}{\sqrt{P_{vw}}} \tag{91}$$

$$= \frac{((AA^T)^{+T} A e_{vw})^T (e_v - e_w)}{\sqrt{P_{vw}}} \tag{92}$$

$$= (e_v - e_w)^T \mathcal{L}^+ (e_v - e_w)\,. \tag{93}$$

And we get the conclusion.

C.3 Comparison with methods that use the spectral gap

We note that: $\gamma^*|\mathcal{E}| = \sqrt{\chi_1}\sqrt{\|\Lambda(t)\|}|\mathcal{E}|$, and using Lemma 3.1 with the assumption that $\lambda_{ij}(t) \geq \frac{1}{2c}\frac{\|\Lambda(t)\|}{|\mathcal{E}|}$, (for some $c > 0$ which should be about $\mathcal{O}(1)$ if no degenerated effects happen) we obtain that :

$$\sqrt{\chi_2}\mathrm{Tr}(\Lambda(t)) \leq \frac{1}{c}\sqrt{(n-1)|\mathcal{E}(t)|\mathrm{Tr}(\Lambda(t))} \tag{94}$$

$$\leq \frac{1}{c}\sqrt{(n-1)|\mathcal{E}(t)\|\Lambda(t)\|} \tag{95}$$

$$\leq \frac{1}{c}\sqrt{\|\Lambda(t)\|}|\mathcal{E}(t)|\,. \tag{96}$$

D Practical Implementation

In this section, we describe in more details the implementation of our algorithm. As we did not physically executed our method on a compute network but rather carried it out on a single machine, all the asynchronous computations and communications had to be simulated. Thus, we will first discuss the method we followed to simulate our asynchronous framework, before detailing the practical steps of our algorithm through a pseudo-code.

D.1 Simulating the Poisson Point Processes

To emulate the asynchronous setting, before running our algorithm, we generate 2 independent sequences of jump times at the graph's scale: one for the computations and one for the communications. As we considered independent P.P.Ps, the time increments follow a Poisson distribution. At the graph's scale, each node spiking at a rate of 1, the Poisson parameter for the gradient steps process is n. Following the experimental setting of the Continuized framework [9], we considered that all edges in $\mathcal{E}(t)$ had the same probability of spiking between t and $t + dt$. Thus, given the sequence of graphs $\mathcal{E}(t)$ and $\mathcal{L}(t)$ their corresponding Laplacians, we computed the parameter λ^* of the communication process as such:

$$\lambda^* = \sqrt{2 \sup_t \chi_1 \left[\frac{\mathcal{L}(t)}{|\mathcal{E}(t)|}\right] \sup_t \chi_2 \left[\frac{\mathcal{L}(t)}{|\mathcal{E}(t)|}\right]}. \tag{97}$$

Having generated the 2 sequences of spiking times at the graph's scale, we run our algorithm playing the events in order of appearance, attributing the *location* of the events by sampling uniformly one node if the event is a gradient step, and sampling uniformly an edge in $\mathcal{E}(t)$ if it is a communication.

D.2 Pseudo Code

We keep the notations introduced in Eq. (7), and recall the following constant values specified in Appendix A.2:

$$\begin{array}{lllll} \eta = \frac{1}{8}\sqrt{\frac{\nu}{L}} & \gamma = \frac{1}{4L} & \delta = \frac{1}{4}\sqrt{\frac{\nu}{L}} & \alpha = \frac{1}{4}\sqrt{\frac{\nu}{L}} & \beta = \frac{1}{2} \qquad \theta = \frac{1}{2}\sqrt{\frac{L}{\nu}} \\ \tilde{\eta} = \frac{1}{8}\sqrt{\frac{\nu}{L}} & \tilde{\gamma} = \frac{1}{4\sqrt{\nu L}} & \tilde{\delta} = 1 & \tilde{\alpha} = \frac{1}{8}\sqrt{\frac{\nu}{L}} & \tilde{\beta} = 2\chi_1^*[\Lambda]\sqrt{\frac{L}{\nu}} \quad \nu = \frac{\mu}{2} \end{array}$$

For the sake of completeness, we also specify the matrix $\mathcal{A}$ describing the linear ODE (14):

$$\mathcal{A} = \begin{pmatrix} -\eta & \eta & 0 & 0 & 0 & 0 \\ \tilde{\eta} & -\tilde{\eta} & 0 & 0 & 0 & 0 \\ 0 & 0 & -\alpha & \alpha & 0 & 0 \\ 0 & -\theta\nu & -\theta & 0 & -\theta & 0 \\ 0 & 0 & 0 & 0 & -\alpha & \alpha \\ 0 & 0 & 0 & 0 & \tilde{\alpha} & -\tilde{\alpha} \end{pmatrix}$$

Described in Appendix D.1, we call `PPPspikes` the aforementioned process returning the ordered sequence of events and time of spikes of the two P.P.Ps. Then, we can write the pseudo-code of our implementation of the DADAO optimizer in Algorithm 2.

Algorithm 2: Pseudo-code of our implementation of DADAO on a single machine.

Input: On each machine $i \in \{1, ..., n\}$, an oracle able to evaluate ∇f_i,
Parameters $\mu, L, \chi_1^*, t_{\max}, n, \lambda^*$.
The sequence of time varying graphs $\mathcal{E}(t)$.

Initialize on each machine $i \in \{1, ..., n\}$:
  Set $X^{(i)} = (x_i, \tilde{x}_i, \tilde{y}_i)$ and $Y^{(i)} = (y_i, z_i, \tilde{z}_i)$ to 0 ;
  Set constants $\nu, \tilde{\eta}, \eta, \gamma, \alpha, \tilde{\alpha}, \theta, \delta, \tilde{\delta}, \beta, \tilde{\beta}$ using μ, L, χ_1^*;
  Set $\mathcal{A}$;
  $T^{(i)} \leftarrow 0$;
$\texttt{ListEvents}, \texttt{ListTimes} \leftarrow \texttt{PPPspikes}(n, \lambda^*, t_{\max})$;
$n_{\text{events}} \leftarrow |\texttt{ListEvents}|$;
for $k \in [\![1, n_{\text{events}}]\!]$ **do**
  if $\texttt{ListEvents}[k]$ *is to take a gradient step* **then**
    $i \sim \mathcal{U}([\![1, n]\!])$;
    $\begin{pmatrix} X^{(i)} \\ Y^{(i)} \end{pmatrix} \leftarrow \exp\left((\texttt{ListTimes}[k] - T^{(i)})\mathcal{A}\right) \begin{pmatrix} X^{(i)} \\ Y^{(i)} \end{pmatrix}$;
    $x_i \leftarrow x_i - \gamma\left(\nabla f_i(x_i) - \nu x_i - \tilde{y}_i\right)$;
    $\tilde{x}_i \leftarrow \tilde{x}_i - \tilde{\gamma}\left(\nabla f_i(x_i) - \nu x_i - \tilde{y}_i\right)$;
    $\tilde{y}_i \leftarrow \tilde{y}_i + (\delta + \tilde{\delta})\left(\nabla f_i(x_i) - \nu x_i - \tilde{y}_i\right)$;
    $T^{(i)} \leftarrow \texttt{ListTimes}[k]$;
  else if $\texttt{ListEvents}[k]$ *is to take a communication step* **then**
    $(i, j) \sim \mathcal{U}(\mathcal{E}(\texttt{ListTimes}[k]))$;
    $\begin{pmatrix} X^{(i)} \\ Y^{(i)} \end{pmatrix} \leftarrow \exp\left((\texttt{ListTimes}[k] - T^{(i)})\mathcal{A}\right) \begin{pmatrix} X^{(i)} \\ Y^{(i)} \end{pmatrix}$;
    $\begin{pmatrix} X^{(j)} \\ Y^{(j)} \end{pmatrix} \leftarrow \exp\left((\texttt{ListTimes}[k] - T^{(j)})\mathcal{A}\right) \begin{pmatrix} X^{(j)} \\ Y^{(j)} \end{pmatrix}$;
    $m_{ij} \leftarrow (y_i + z_i - y_j - z_j)$; `// Message exchanged.`
    $z_i \leftarrow z_i - \beta m_{ij}$;
    $\tilde{z}_i \leftarrow \tilde{z}_i - \tilde{\beta} m_{ij}$;
    $z_j \leftarrow z_j + \beta m_{ij}$;
    $\tilde{z}_j \leftarrow \tilde{z}_j + \tilde{\beta} m_{ij}$;
    $T^{(i)} \leftarrow \texttt{ListTimes}[k]$;
    $T^{(j)} \leftarrow \texttt{ListTimes}[k]$;
return x_i, *the estimate of* x^* *on each worker* i.

E Further experiments

In this section, we present additional numerical results comparing our method DADAO to ADOM+ [17] in the time varying setting, and report our results using SGD.

E.1 Time-Varying setting

In this section, we study the effect of the parameter χ_1^* on the convergence speed of ADOM+ [17] and DADAO by varying it between $\chi_1^* \in \{3, 33, 180, 233\}$ for random geometric graphs of size $n = 20$ on the decentralized linear regression task with time-varying topology. To visualize the difference in connectivity these changes in χ_1^* represent, we plot 4 examples of graphs of the said types with varying values of χ_1^* in Fig. 3. In Fig. 4, we show the different convergence speeds it entails.

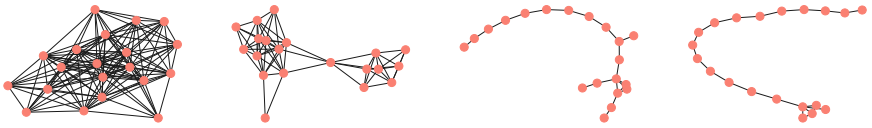

Figure 3: Examples of random geometric graphs of size $n = 20$ with χ_1^* taking values in, from left to right, $\chi_1^* \in \{3, 33, 180, 233\}$.

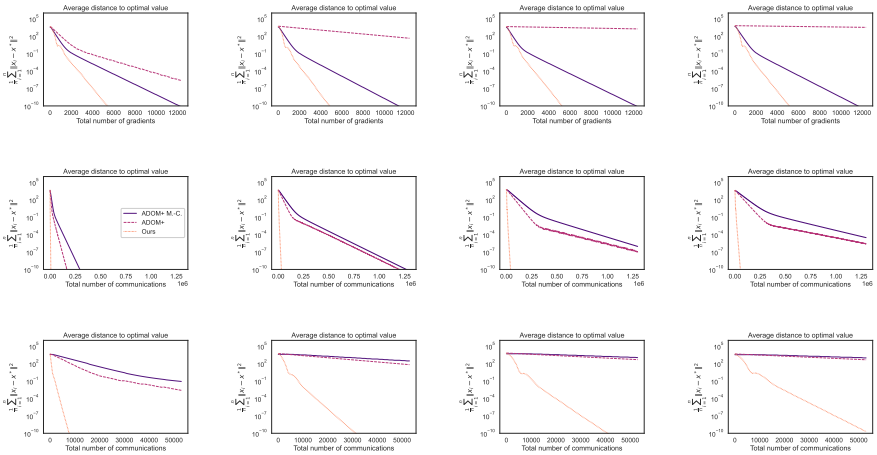

Figure 4: Comparison between ADOM+ [17] and DADAO, using the same data for linear regression on $n = 20$ workers and the same sequence of random connected graphs with varying topology and χ_1^* taking values in, from the left to the right column, $\chi_1^* \in \{3, 33, 180, 233\}$.

As expected, we observe in Fig. 4 that varying χ_1^* has no effect on the number of gradient computations of both ADOM+ M.-C and DADAO, but the smaller the χ_1^*, the better the slope for ADOM+ in terms of gradient steps. We also confirm for all 3 methods that the smaller χ_1^*, the less communication is needed to reach an ϵ-precision.

E.2 Stochastic Gradient Descent with DADAO

In the SGD setting, we sample uniformly at random a mini-batch of size B data points on each worker and compute the losses and stochastic gradients $\nabla f_i(x_i, \xi)$ with respect to these samples. To study the effect of the quadratic error σ^2 of our gradients on the resulting biases of our parameters, we fix both the data (for linear regression) and the communication network (graph star of size $n = 20$) and try different values of B. To monitor our results, we plot the mean distance to x^* of the running average over time of our local parameters. Taking the notations introduced in Sec. 4.2, this can be written as:

$$\frac{1}{n}\sum_{i=1}^{n}\left\|\frac{1}{k_i}\sum_{j=1}^{k_i}x_j^{(i)} - x^*\right\|^2,$$

where k_i designates a local event counter. We report our results in Fig. 5.

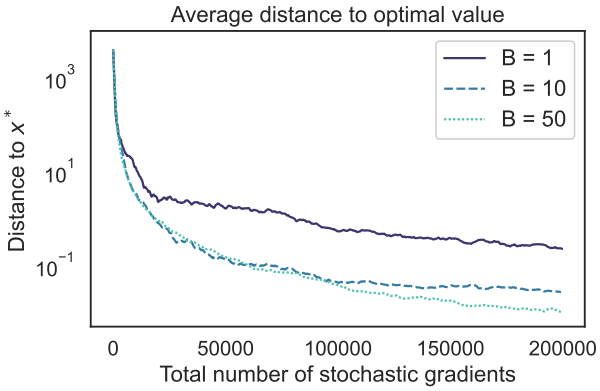

Figure 5: Effect of the batch size B on the convergence of our method DADAO.

We confirm that the less variance on our stochastic gradients, the less our estimates $\frac{1}{k_i}\sum_{j=1}^{k_i}x_j^{(i)}$ are biased.