Asynchronous Speedup in Decentralized Optimization

Mathieu Even, Hadrien Hendrikx, Laurent Massoulié

***Abstract*— In decentralized optimization, nodes of a communication network each possess a local objective function, and communicate using gossip-based methods in order to minimize the average of these per-node functions. While synchronous algorithms are heavily impacted by a few slow nodes or edges in the graph (the *straggler problem*), their asynchronous counterparts are notoriously harder to parametrize. Indeed, their convergence properties for networks with heterogeneous communication and computation delays have defied analysis so far.**

In this paper, we use a *continuized* framework to analyze asynchronous algorithms in networks with delays. Our approach yields a precise characterization of convergence time and of its dependency on heterogeneous delays in the network. Our continuized framework benefits from the best of both continuous and discrete worlds: the algorithms it applies to are based on event-driven updates. They are thus essentially discrete and hence readily implementable. Yet their analysis is essentially in continuous time, relying in part on the theory of delayed ODEs.

Our algorithms moreover achieve an *asynchronous speedup*: their rate of convergence is controlled by the eigengap of the network graph weighted by local delays, instead of the network-wide worst-case delay as in previous analyses. Our methods thus enjoy improved robustness to stragglers.

***Index Terms*— Decentralized, gossip, optimization, asynchronous**

I. Introduction

We consider solving stochastic optimization problems that are distributed amongst n agents (indexed by $V = [n]$) who can compute stochastic gradients in parallel. This includes classical federated setups, such as distributed and federated learning. Depending on the application, agents have access to either same shared data distribution or a different agent-specific distributions. In recent years, such stochastic optimization problems have continued to grow rapidly in size, both in terms of the dimension d of the optimization variable—i.e., the number of model parameters in machine learning—and in terms of the quantity of data—i.e., the number of data samples m being used over all agents. With d and m regularly reaching the hundreds or thousands of billions [12], [59], it is increasingly necessary to use parallel optimization algorithms to handle the large scale.

With *communication cost* being one of the major bottlenecks of parallel optimization algorithms, there are several directions aimed to improve communication efficiency. Amongst the others (such as local update steps [55], [63] and communication compression [2], [31]), **decentralization** and **asynchrony** are the two popular techniques for reducing the communication time. Decentralization [30], [38] eliminates the dependency on the central server—frequently a major bottleneck in distributed learning—while naturally amplifying privacy guarantees [13]. Asynchrony [5], [49], [60] shortens the time per computation rounds and allows more updates to be made during the same period of time. It aims to overcome several possible sources of delays: nodes may have *heterogeneous hardware* with different computational throughputs [26], [27], *network latency* can slow the communication of gradients, and nodes may even just *drop out* [50]. Moreover, slower "*straggler*" compute nodes can arise in many natural parallel settings, including training ML models using multiple GPUs [11] or in the cloud; sensitivity to these stragglers poses a serious problem for synchronous algorithms, that depend on the slowest agent. In decentralized synchronous optimization where communication times between pairs of nodes may be heterogeneous, the algorithm can even be further slowed down by *straggling communication links*.

A. Decentralized and asynchronous setting

More formally, we study the following optimization problem:

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

where each individual function $f_i : \mathbb{R}^d \to \mathbb{R}$ for $i \in [n]$ is held by an agent i, and we consider *asynchronous* and *decentralized* optimization methods that do not rely on a central coordinator. In the case of empirical risk minimization, f_i represents the empirical risk for the local dataset of node i, and f the empirical risk over all datasets. Another important example, that plays the role of a toy problem for both decentralized and/or stochastic optimization is that of **network averaging** [9], corresponding to $f_i(x) = \|x - c_i\|^2$ where c_i is a vector attached to node i. In this case, the solution of Problem (1) reads $\bar{c} = \frac{1}{n}\sum_{i=1}^{n} c_i$.

We assume that agents are located at the nodes of a connected, undirected graph $G = (V, E)$ with node set $V = [n]$.

Manuscript submitted August 31th.

The work of Mathieu Even and Laurent Massoulié is supported by the French government under management of Agence Nationale de la Recherche as part of the "Investissements d'avenir" program, reference ANR19-P3IA-0001 (PRAIRIE 3IA Institute).

Mathieu Even is with Inria Paris and ENS Paris, France.

Hadrien Hendrikx is with Ecole Polytechnique Fédérale de Lausanne, Switzerland.

Laurent Massoulié is with Inria Paris, ENS Paris and the MSR-Inria Joint Center.

An agent $i \in V$ can compute first-order quantities (gradients) related to its local objective function f_i, and can communicate with any adjacent agent in the graph. Our model of asynchrony derives from the popular randomized gossip model of [9]. In this model, nodes update their local values at random activation times using pairwise communication updates. This asynchronous model makes the idealized assumption of instantaneous communications, and hence does not faithfully represent practical implementations. To alleviate this drawback, several works [4], [36], [39], [53], [57], [58], [61], [64], [65], [69], [69] introduce communication and computation delays in either pairwise updates, or in asymmetric gossip communications.

However, all these works provide convergence guarantees that either require global synchronization between the nodes, or are implicitly determined by an upper bound on the worst-case delay in the whole graph. Indeed they assume that for some given $k_{\max} > 0$, for all edges $(ij) \in E$, each communication between agents i and j overlaps with at most $k_{\max}$ other communications in the whole graph. Thus assuming distributed asynchronous operation where individual nodes schedule their interactions based only on local information, the $k_{\max}$ constraint can only be enforced by requiring individual nodes to limit their update frequency to $1/(n\tau_{\max})$, while the resulting algorithms have temporal convergence guarantees that need to be proportional to $\tau_{\max}$. They are thus not robust to *stragglers*, i.e. slow nodes or edges in the graph that induce large $\tau_{\max}$. Moreover, all these works assume that agents i or graph edges (ij) are activated for agent interaction sequentially in an *i.i.d.* manner, which can only be unforced with a central coordinator, and can lead to deadlocks.

B. Graph-dependent asynchronous speedup

To understand the scope for improvement over methods that rely on such worst-case delays or over synchronous algorithms, recall that for synchronous algorithms with updates performed every $\tau_{\max}$ seconds, for L-smooth and σ-strongly convex functions f_i, the time required to reach precision $\varepsilon > 0$ for $\frac{1}{n}\sum f_i$ is lower-bounded by [51]:

$$\Omega\left(\tau_{\max}\mathrm{Diam}(G)\sqrt{\kappa}\ln(\varepsilon^{-1})\right), \tag{2}$$

where $\kappa = L/\sigma$ is the condition number of the functions f_i and $\mathrm{Diam}(G)$ is the diameter of graph G.

In this article we seek better dependency on individual delays in the network. Specifically we consider the following

*Assumption 1 (**Heterogeneous delays**):* There exist τ_{ij} for $(ij) \in E$ and τ_i^{comp} for $i \in V$ such that communications between two neighboring agents i and j in the graph take time at most τ_{ij}, and a computation at node i takes time at most τ_i^{comp}.

Under such heterogeneous delay assumptions, **how robust to stragglers can decentralized algorithms be?** One can adapt the proof of [51] to Assumption 1 to establish the generalized form of lower bound (2):

$$\Omega\left(D(\tau)\sqrt{\kappa}\ln(\varepsilon^{-1})\right), \tag{3}$$

where $D(\tau) = \sup_{(i,j)\in V^2}\mathrm{dist}(i,j)$ for:

$$\mathrm{dist}(i,j) = \inf_{\substack{(i=i_0,\dots,i_p=j),\\ \forall 1\le k\le p,\,(i_k,i_{k+1})\in E}} \tau_i^{\mathrm{comp}} + \tau_j^{\mathrm{comp}} + \sum_{k=0}^{p-1}\tau_{i_k i_{k+1}}.$$

Here $\mathrm{dist}(i,j)$ is the time distance between nodes i and j, and $D(\tau)$ is the *diameter* of graph G for this distance. $D(\tau)$ is the generalization of $\tau_{\max}\mathrm{Diam}(G)$ to the heterogeneous-delay setting. This lower bound suggests that robustness to stragglers is possible: indeed if a fraction of the nodes or edges is too slow (large delay τ_{ij}), this may not even impact this lower bound, since the shortest path between two nodes may always take another route.

We aim at building *decentralized* algorithms with performance guarantees that enjoy such robustness to individual delay bounds. However, since we focus on fully decentralized algorithms, our performance guarantees will not be expressed in terms of some diameter $D(\tau)$ as in (3) but instead in terms of some spectral characteristics of the graph at hand[1]. Specifically, let us introduce the Graph Laplacian.

*Definition 1 (**Graph Laplacian**):* Let $\nu = (\nu_{ij})_{(ij)\in E}$ be a set of non-negative real numbers. The *Laplacian* of the graph G weighted by the ν_{ij}'s is the matrix $\Delta_G(\nu)$ with (i,j) entry equal to $-\nu_{ij}$ if $(ij) \in E$, $\sum_{k\sim i}\nu_{ik}$ if $j = i$, and 0 otherwise. In the sequel ν_{ij} always refers to the weights of the Laplacian, and $\lambda_2(\Delta_G(\nu))$ denotes this Laplacian's second smallest eigenvalue.

We thus seek performance guarantees similar to (3) with in place of $D(\tau)$ the term $\lambda_2(\Delta_G(\nu))^{-1}$ for some parameters ν_{ij} that depend on delay characteristics local to edge (ij). As a consequence, we highlight the fact that due to this local dependency on the delays (that no existing work considers), there will always exist graphs and topologies that keep our rate of convergence constant, while making existing approaches (that all depend on worst-case delays) fail to converge in a reasonable amount of time.

C. Related works

Asynchronous optimization. Asynchronous optimization has a long history. In the 1970s, [5] considered shared-memory asynchronous fixed-point iterations, and an early convergence result for Asynchronous SGD was established by [60]. Recent analyses typically rely on bounded delays [1], [37], [49], [56], while some algorithms try to adapt to the delays [20], [32], [42], [44], [45], [54], [70], in order to depend only on an average delay. For more examples of stochastic asynchronous algorithms, we refer readers to the surveys by [3], [7].

More closely related to our work but in the centralized setting, [30], [44] concurrently proved that *SGD is always faster than minibatch SGD*. Concretely, in terms of *wall-clock* time (*physical* time), if each of $i \in [n]$ machines takes a time τ_i to compute a stochastic gradient and to communicate with the central server, Asynchronous SGD is $\frac{1}{n}\sum_{i=1}^n \frac{\tau_{\max}}{\tau_i} \ge 1$ faster than its Minibatch SGD (its synchronous counterpart). In this case, the *asynchronous speedup* consists in the speedup

[1]Note that similar spectral characteristics (albeit based on a single worst-case delay parameter $\tau_{\max}$) appear in [4], [36], [51], [53], [61], [64].

induced by asynchrony: the asynchronous algorithm is robust to heterogeneous delays and to stragglers, and the more heterogeneous the delays are, the better the asynchronous algorithm is compared to the synchronous one. Our paper focuses on the asynchronous speedup that can be reached with decentralized communications, a very related yet much more challenging problem.

Decentralized optimization. Gossip algorithms [9], [15] were initially introduced to compute the global average of local vectors with local pairwise communications only (no central coordinator), and were generalized to decentralized optimization. Two types of gossip algorithms appear in the literature: synchronous ones, where all nodes communicate with each other simultaneously [15], [31], [51], [52], and randomized ones [9], [47]. A third category considers directed (non-symmetric) communication graphs [4], [67] which are much easier to implement asynchronously. Random-walk (token) based approaches can also be considered [17], [23]. In the synchronous framework, the communication speed is limited by the slowest node (*straggler* problem), whereas the classical randomized gossip framework of [9] assumes communications to happen instantaneously, and thus does not address the question of how to deal with delays.

Decentralized optimization and asynchrony. Combining both decentralization and asynchrony is a challenging problem, and it is only recently that this question has risen a surge of interest [4], [8], [40], [41], [46], [69]. These works are however restricted to a given communication protocol and static topologies under an *i.i.d.* sampling of the nodes or edges that become active [4], [8], [37], [46], no communication delays [8], [37], [46], or their analyses rely on an upper-bound on the maximal computation and communication delays [4], [8], [39]–[41], [46], [66], [69]. This latter point is exactly the goal of our paper: how can we relax the worst case delay dependency and capture quantitatively delay heterogeneity in the graph and relate it to a *wall-clock asynchronous speedup*? Similarly to the asynchronous speedup of Asynchronous SGD described just above, we will quantify the asynchronous speedup as a graph dependent quantity that takes into account pairwise communication delays and their heterogeneity.

In this paper, we deal with asynchrony and delays from a different viewpoint than the references cited above, where delays are introduced in the analysis as discrete-time delays that are uniformly bounded by some given quantity, while our analysis is inspired by time-delayed ODE systems [48]. Consequently, the assumptions related to delays and asynchrony (such as Assumption 1) do not need to be translated into discrete-time ones, as in the above references. Finally, we believe our continuous time framework to be particularly adequate for the study and design of asynchronous algorithms, in the decentralized setting as in this paper, but also in centralized settings where it may remove the need to introduce a discrete ordering of events and thus avoid difficulties that lead to unrealistic assumptions, such as the *after/before-read* approaches [35].

Finally, [62] study how sparsifying the communication graph can lead to faster decentralized algorithm. Their approach is different from ours in Section VI: they do not consider asynchronous algorithms with physical constraints (delays and capacity), but synchronous algorithms where sequentially matchings are built in the graph. Yet, we observe similar phenomenon as theirs in Section VI.

Network time protocol (NTP). Our work implicitly assumes that nodes of the graph are aware of a global absolute time (continuous-time), which is a feature of the *continuized* framework, presented in a subsequent subsection. This is very different from knowing a global (discrete) iteration counter that tracks the number of updates. The latter is impossible in a decentralized framework, while the former can be achieved fairly easily, up to some synchronization errors. In its standard version, the Network Time Protocol (*NTP*) ensures that machines connected to internet have access to the global time with a precision of the millisecond. In our case, in order to achieve better precision, more refined versions of NTPs can be used, such as [21], that yields a precision of the order of the nanosecond (10^{-9} seconds). Hence, up to negligible errors, agents indeed share a global continuous-time clock.

D. Contributions

(i) We first consider the *network averaging problem*, for which we introduce *Delayed Randomized Gossip* in Section II. Building on recent works on continuized gradient descent for Nesterov acceleration [18], we analyze Delayed Randomized Gossip in the continuized framework, that allows a continuous-time analysis of an algorithm even though the latter is based on discrete, hence practically implementable operations. Our analysis leads to explicit stability conditions that have the appealing property of being *local*, i.e. they require each agent to tune its algorithm parameters to delay bounds in its graph neighborhood.

They ensure a linear rate of convergence determined by $\lambda_2\big(\Delta_G(\nu)\big)$, for weights of order $\nu = 1/(\sum_{(kl)\sim(ij)} \tau_{kl})^2$. This dependency of weights in the Laplacian on local delay bounds is what we call the *asynchronous speedup*, since it implies a scaling that is no longer proportional to $\tau_{\max}$.

(ii) Using an augmented graph approach, we propose algorithms that generalize Delayed Randomized Gossip to solve the decentralized optimization problem in Section IV. Under strong convexity and smoothness assumptions on the local functions f_i, we obtain local stability conditions yielding an asynchronous speedup for this more general setup.

(iii) We further generalize our setup with the introduction of *local capacity constraints* in Section V, in order to take into account the fact that nodes or edges cannot handle an unlimited number of operations in parallel. To that end, we introduce *truncated Poisson point processes* in the continuized framework for the analysis.

(iv) The theoretical guarantees for our algorithms in Sections II, IV and V are all based on general guarantees for so-called delayed coordinate gradient descent in the continuized framework, that we present and establish in Section III. These results may be of independent interest beyond our current focus on decentralized optimization.

[2]We write $(ij) \sim (kl)$ and say that two edges (ij), (kl) are neighbors if they share at least one node.

(v) Finally, we identify from our stability conditions and convergence guarantees a phenomenon reminiscent of *Braess*'s paradox (Section VI): deleting some carefully chosen edges can lead to faster convergence. This in turn suggests rules for sparsifying communication networks in distributed optimization.

Our paper is thus organized in a linear way, where difficulties are added one by one for pedagogical sake. We start by adding delays in communications (II) before providing the necessary tools for the analysis (III). We then build on this to progressively add local functions (IV) and local capacity constraints (V). Our results are overall proof concepts that asynchronous and decentralized algorithms can benefit from an asynchronous speedup, that we quantify.

II. Delayed Randomized Gossip for Network Averaging

Focusing in this section on the Network Averaging Problem, we introduce Delayed Randomized Gossip and state its convergence guarantees. We first begin with reminders on randomized gossip [9].

A. Randomized gossip

Let $G = (V, E)$ be a connected graph on the set of nodes $V = [n]$, representing a communication network of agents. Each agent $i \in V$ is assigned a real vector $x_0(i) \in \mathbb{R}^d$. The goal of the averaging (or gossip) problem is to design an iterative procedure allowing each agent in the network to estimate the average $\bar{x} = \frac{1}{n}\sum_{i=1}^n x_0(i)$ using only local communications, *i.e.*, communications between adjacent agents in the network.

In randomized gossip [9], time t is indexed continuously by $\mathbb{R}^+$. A Poisson point process [29] (abbreviated as *P.p.p.* in the sequel) $\mathcal{P} = \{T_k\}_{k\geq 1}$ of intensity $I > 0$ on $\mathbb{R}^+$ is generated: $T_0 = 0$ and $(T_{k+1} - T_k)_{k\geq 0}$ are *i.i.d.* exponential random variables of mean $1/I$. For positive intensities $(p_{ij})_{(ij)\in E}$ such that $\sum_{(ij)\in E} p_{ij} = I$, for every $k \geq 0$, at T_k an edge $(i_k j_k)$ is *activated* with probability $p_{i_k j_k}/I$, upon which adjacent nodes i_k and j_k communicate and perform a pairwise update. The *P.p.p.* assumption implies that edges are activated independently of one another and from the past: the activation times of edge (ij) form a *P.p.p.* of intensity p_{ij}.

To solve the gossip problem, [9] proposed the following strategy: each agent $i \in V$ keeps a local estimate $x_t(i)$ of the average and, upon activation of edge $(i_k j_k)$ at time $T_k \in \mathbb{R}^+$, the activated nodes i_k, j_k average their current estimates:

$$x_{T_k}(i_k),\, x_{T_k}(j_k) \longleftarrow \frac{x_{T_k-}(i_k) + x_{T_k-}(j_k)}{2}\,. \tag{4}$$

Writing $f(x) = \sum_{(ij)\in E} \frac{p_{ij}}{I} f_{ij}(x)$, for $f_{ij}(x) = \frac{1}{2}\|x(i) - x(j)\|^2$ and $x = (x(i))_{i\in V}$, [18] observe that local averages (4) correspond to stochastic gradient steps on f:

$$x_{T_k} \longleftarrow x_{T_k-} - \frac{K_{i_k j_k}}{p_{i_k j_k}} \nabla f_{i_k j_k}(x_{T_k-})\,, \tag{5}$$

for step sizes $K_{i_k j_k} = \frac{p_{i_k j_k}}{2}$.

These updates can also be derived from coordinate gradient descent steps. Let $A \in \mathbb{R}^{V\times E}$ be such that for all $(ij) \in E$, $Ae_{ij} = \mu_{ij}(e_i - e_j)$ for arbitrary $\mu_{ij} \in \mathbb{R}$, where $(e_{ij})_{(ij)\in E}$ and $(e_i)_{i\in V}$ are the canonical bases of $\mathbb{R}^E$ and $\mathbb{R}^V$. Then, let $g(\lambda) = \frac{1}{2}\|A\lambda\|^2$ for $\lambda \in \mathbb{R}^{E\times d}$, so that the coordinate gradient $\nabla_{ij} g(\lambda)$ writes $\nabla_{ij} g(\lambda) = \mu_{ij}((A\lambda)_i - (A\lambda)_j)$. Thus, provided that for some $\lambda_{T_k-} \in \mathbb{R}^{E\times d}$, $x_{T_k-} - \bar{x} = A\lambda_{T_k-}$, the local averaging defined in Equation (4) is equivalent to $x_{T_k} - \bar{x} = A\lambda_{T_k}$, where:

$$\lambda_{T_k} = \lambda_{T_k-} - \frac{K_{i_k j_k}}{p_{i_k j_k}\mu_{i_k j_k}^2} \nabla_{i_k j_k} g(\lambda_{T_k-})\,, \tag{6}$$

for $K_{i_k j_k} = \frac{p_{i_k j_k}}{2}$. Hence, the gossip algorithm of [9] can be viewed as a simple block-coordinate gradient descent on variables $\lambda \in \mathbb{R}^{E\times d}$ indexed by the edges of the graph instead of the nodes.

Yet, this continuous-time model with *P.p.p.* activations implicitly assumes instantaneous communications, or some form of waiting. Indeed, the gradient is computed on the current value of the parameter, which is x_{T_k-}. In the presence of (heterogeneous) communication delays (Assumption 1), a more realistic update uses the parameter x_{S_k} at a previous time $S_k < T_k$, to account for the time it takes to compute and communicate the gradient. In this case, the updates write as

$$x_{T_k} \longleftarrow x_{T_k-} - \frac{K_{i_k j_k}}{p_{i_k j_k}} \nabla f_{i_k j_k}(x_{S_k})\,. \tag{7}$$

Equivalently, from the point of view of node i_k:

$$x_{T_k}(i_k) \leftarrow x_{T_k-}(i_k) - \frac{K_{i_k j_k}}{p_{i_k j_k}}\big(x_{S_k}(i_k) - x_{S_k}(j_k)\big)\,.$$

B. The continuized framework

Our approach uses the **continuized framework** [18], which amounts to consider continuous-time evolution of key quantities, with discrete jumps at the instants of Poisson point processes. This gives the best of both continuous (for the analysis and assumptions) and discrete (for the implementation) worlds. From now on and for the rest of the paper, we assume that Assumption 1 holds.

Edges $(ij) \in E$ locally generate independent *P.p.p.* $\mathcal{P}_{ij}$ of intensity $p_{ij} > 0$ (random activation times, with *i.i.d.* intervals, exponentially distributed with mean $1/p_{ij}$). As mentioned previously, $\mathcal{P} = \bigcup_{(ij)\in E} \mathcal{P}_{ij}$ is a *P.p.p.* of intensity $I = \sum_{(ij)\in E} p_{ij}$, and noting $\mathcal{P} = \{T_1 < T_2 < \dots\}$, at each clock ticking T_k, $k \geq 1$, an edge $(i_k j_k)$ is chosen with probability $p_{i_k j_k}/I$. This time T_k corresponds to a communication update between nodes i_k and j_k started at time $T_k - \tau_{i_k j_k}$ [3]. Assumption 1 ensures that the communication started at time $T_k - \tau_{ij}$ takes some time $\tau^{(k)} \leq \tau_{i_k j_k}$ and is thus completed before time T_k so that the update at time T_k is indeed implementable. Consequently, the sequence $(x_t)_t$ generated by Algorithm 1

[3] Standard properties of P.p.p. guarantee that the sequence of points of $\mathcal{P}_{ij}$ translated by τ_{ij} is a P.p.p. with the same distribution.

Algorithm 1 Delayed randomized gossip, edge (ij)

Step size $K_{ij} > 0$ and intensity $p_{ij} > 0$

Initialization $T_1(ij) \sim \mathrm{Exp}(p_{ij})$

for $\ell = 1, 2, \dots$ **do**

  $T_{\ell+1}(ij) = T_\ell(ij) + \mathrm{Exp}(p_{ij})$.

end for

for $\ell = 1, 2, \dots$ **do**

  At time $T_\ell(ij) - \tau_{ij}$ for, i sends $\hat{x}_i = x_{T_\ell(ij)-\tau_{ij}}(i)$ to j and j sends $\hat{x}_j = x_{T_\ell(ij)-\tau_{ij}}(j)$ to i.

  At time $T_\ell(ij)$,

$$\begin{aligned} x_{T_\ell(ij)}(i) &\leftarrow x_{T_\ell(ij)-}(i) - \frac{K_{ij}}{p_{ij}}\big(\hat{x}_i - \hat{x}_j\big)\,, \\ x_{T_\ell(ij)}(j) &\leftarrow x_{T_\ell(ij)-}(j) - \frac{K_{ij}}{p_{ij}}\big(\hat{x}_j - \hat{x}_i\big)\,, \end{aligned} \tag{9}$$

end for

writes as:

$$\begin{cases} x_{T_k}(i) = x_{T_k-}(i) \quad \text{if} \quad i \notin \{i_k, j_k\}\,, \\ x_{T_k}(i_k) \leftarrow x_{T_k-}(i_k) - \dfrac{K_{i_kj_k}}{p_{i_kj_k}}\big(x_{T_k-\tau_{i_kj_k}}(i_k) - x_{T_k-\tau_{i_kj_k}}(j_k)\big), \\ x_{T_k}(j_k) \leftarrow x_{T_k-}(j_k) - \dfrac{K_{i_kj_k}}{p_{i_kj_k}}\big(x_{T_k-\tau_{i_kj_k}}(j_k) - x_{T_k-\tau_{i_kj_k}}(i_k)\big). \end{cases}$$

Algorithm 1 is the pseudo-code for *Delayed Randomized Gossip*, from the viewpoint of two adjacent nodes i and j. The times $T_\ell(ij)$ for $\ell \geq 1$ denote the activation times of edge (ij). They follow a P.p.p. of intensity p_{ij}, and are sequentially determined by adjacent nodes i and j. $\mathrm{Exp}(p)$ is an exponential random variable of parameter p. In Algorithm 1, Delayed Randomized Gossip is presented from the local viewpoint of edges $(ij) \in E$ ($T_\ell(ij)$ is the ℓ^{th} activation of edge (ij)), while the equations just above are a global description of the algorithm (T_k is the k^{th} edges activation in the graph).

Formally, this decentralized and asynchronous algorithm corresponds to a jump process solution of a *delayed stochastic differential equation*. Defining $N(\mathrm{d}t, (ij))$ as the *Poisson* measure on $\mathbb{R}^+ \times E$ of intensity $I\mathrm{d}t \otimes \mathcal{U}_p$ where $\mathcal{U}_p$ is the probability distribution on E proportional to $(p_{ij})_{(ij)\in E}$ ($\mathcal{U}_p((ij)) = p_{ij}/I$), we have:

$$\mathrm{d}x_t = -\int_{\mathbb{R}^+\times E} \frac{K_{ij}}{p_{ij}} \nabla f_{ij}(x_{t-\tau_{ij}}) \mathrm{d}N(t, (ij))\,. \tag{8}$$

Next section presents convergence guarantees for iterates generated by delayed randomized gossip.

C. Convergence guarantees

We begin by recalling the key quantities introduced. *(i)* The constraints inherent to the problem are the communication delays, upper-bounded by constants τ_{ij}. *(ii)* Parameters of the algorithm are: step sizes $K_{ij} > 0$ and intensities p_{ij} of the local P.p.p. that trigger communications between adjacent nodes i and j. For arbitrary intensities p_{ij} and delay bounds τ_{ij}, we shall provide local conditions on the step sizes K_{ij} that guarantee stability and convergence guarantees. This is to be contrasted with the situation –discussed in Section V– where in addition there are capacity constraints, for which additional conditions on the intensities p_{ij} are needed to prove convergence.

*Theorem 1 (**Delayed Randomized Gossip**):* Assume that for all $(ij) \in E$, we have[4]:

$$K_{ij} \leq \frac{p_{ij}}{1 + \sum_{(kl)\sim(ij)} p_{kl}\big(\tau_{ij} + e\tau_{kl}\big)}\,. \tag{10}$$

Let $\nu_{ij} \equiv K_{ij}$, $(ij) \in E$, and $\tau_{\max} = \max_{(ij)\in E} \tau_{ij}$. Let $\gamma > 0$ be such that:

$$\gamma \leq \min\left(\frac{\lambda_2\big(\Delta_G(\nu)\big)}{2}\,,\, \frac{1}{\tau_{\max}}\right).$$

For any $T \geq 0$, for $(x_t)_{t\geq 0}$ generated with delayed randomized gossip (Algorithm 1) or equivalently by the delayed SDE in Equation (8), we have:

$$\frac{\int_0^T e^{\gamma t}\mathbb{E}\left[\|x_t - \bar{x}\|^2\right]\mathrm{d}t}{\int_0^T e^{\gamma t}\|x_0 - \bar{x}\|^2 \mathrm{d}t} \leq e^{-\frac{\gamma T}{2}} \frac{1 + \frac{\tau_{\max}}{T}}{1 - \gamma\tau_{\max}}\,. \tag{11}$$

Using Jensen inequality then yields the following corollary: a weighted average of the iterates is decreasing linearly with time. This is a countinuous-time counterpart of the weighted average considered in most decentralized optimization algorithms (strongly convex case in [30], e.g.). Note that this integral is in fact discrete and can be expressed as a sum, since $(x_s)_s$ is a jump process. Finally, to converse this result in discrete time (number of pairwise communications), we just need to notice that the number of communications that happened before time $T \geq 0$ is equal in mean to $T\sum_{(ij)\in E} p_{ij}$, and concentrates around this value for T large.

Corollary 1: Under the same assumptions as Theorem 1, for $(x_t)_{t\geq 0}$ generated with delayed randomized gossip, define $(\tilde{x}_t)_{t\geq 0}$ as the exponentially weighted averaging along the trajectory of (x_t):

$$\tilde{x}_t = \gamma \frac{\int_0^t e^{\gamma s} x_s \mathrm{d}s}{e^{\gamma t} - 1}\,.$$

Then, for all $T \geq 0$,

$$\mathbb{E}\left[\|\tilde{x}_T - \bar{x}\|^2\right] \leq e^{-\frac{\gamma T}{2}} \|x_0 - \bar{x}\|^2 \frac{1 + \frac{\tau_{\max}}{T}}{1 - \gamma\tau_{\max}}\,.$$

An essential aspect of Theorem 1 lies in the explicit sufficient conditions for convergence it establishes for our proposed schemes, and on how they only rely on (upper bounds on) individual delays. We now discuss the ***asynchronous speedup*** obtained by fine-tuning algorithm parameters according to delays.

For many graphs of interest such as grids, hypergrids, trees…in the large network limit $n \to \infty$ one has $\lambda_2(\Delta_G(\nu)) \to 0$[5] and so $\min(\lambda_2(\Delta_G), 1/\tau_{\max}) = \lambda_2(\Delta_G)$ should hold. More precisely, let Λ_n be the smallest non-null eigenvalue of the Laplacian with weights equal to $1/\mathrm{degree}_{ij}$

[4]Note that $(ij) \sim (ij)$; constant e is $\exp(1)$.

[5]Networks for which this fails are known as *expanders*.

(the degree of an edge). Then, if for instance $p_{ij} = 1/\tau_{ij}$, as long as we have:

$$\Lambda_n \leq \frac{\tau_{\min}}{d_{\max}\tau_{\max}},$$

where $d_{\max}$ is the max degree in the graph, then we have $\gamma = \mathcal{O}\left(\frac{\lambda_2\left(\Delta_G(\nu)\right)}{2}\right)$. For the line and cyclic graphs, we have $\Lambda_n = \mathcal{O}(1/n^2)$, for the $D-$dimensional grid we have $\Lambda_n = \mathcal{O}(1/n^{2/D})$, so that the above condition will hold in most cases. However, for small graphs or expander graphs that do not verify the condition $\Lambda_n = \mathcal{O}(\frac{\tau_{\min}}{\tau_{\max}})$, the linear rate of convergence turns into $\frac{1}{\tau_{\max}}$. Noting that synchronous gossip has a linear rate of convergence of $\frac{\Lambda_n}{\tau_{\max}}$, we still notice an overall improvement of magnitude Λ_n for such graphs, and our decentralized approach behaves as if it were centralized. The *asynchronous speedup* consists in having a rate of convergence as the eigengap of the Laplacian of the graph weighted by local communication constraints: this is thus the case here, with the term $\lambda_2(\Delta_G(K))$, where each K_{ij} is impacted only by local quantities.

As mentioned in the introduction, this quantity should be understood as the analogue in decentralized optimization of the squared diameter of the graph (using time distances) in (3) in centrally coordinated algorithms and as expected, gossip algorithms are affected by spectral properties of the graph. In Theorem 1, these properties reflect delay heterogeneity across the graph: here, $\lambda_2(\Delta_G(K))^{-1}$ the mixing time of a random walk on the graph where jumping from node i to j takes a time $\tilde{\tau}_{ij} = K_{ij}^{-1}$. In contrast, previous analyses (of synchronous or asynchronous algorithms) involve the mixing time of a random walk with times between jumps set to a quantity that is linearly dependent on $\tau_{\max}$. We coin this discrepancy the *asynchronous speedup*.

Equation (10) suggests a scaling of $p_{ij} \approx 1/\tau_{ij}$, giving local weights K_{ij} of order $1/(\text{degree}_{ij}\tau_{ij})$ where degree_{ij} is the degree of edge (ij) in the edge-edge graph. On the other hand, synchronous algorithms are slowed down by the slowest node: the equivalent term would be of order $\lambda_2(\Delta_G(1/(\text{degree}_{ij}\tau_{\max}))$. Indeed, for a *gossip matrix* $W \in \mathbb{R}^{V\times V}$ (W is a symmetric and stochastic matrix), the equivalent factor in synchronous gossip [15] is $\lambda_2(\Delta_G(W_{ij}\tau_{\max}))$, and W_{ij} is usually set as $1/\text{degree}_{ij}$.

D. A delayed ODE for mean values in gossip

Before proving Theorem 1, we provide some intuition for its conditions and the resulting convergence rate. We do this by studying the means of the iterates, that verify a delayed linear ordinary differential equation, easier to study than the process itself, for which we provide stability conditions.

Denoting $y_t = \mathbb{E}[x_t] \in \mathbb{R}^{n\times d}$, for $t \geq 0$, where $(x_t)_{t\geq 0}$ is generated using delayed randomized gossip updates (7), we have:

$$\frac{\mathrm{d}y_t}{\mathrm{d}t} = -\sum_{(ij)\in E} K_{ij}\nabla f_{ij}(y_{t-\tau_{ij}})\,. \tag{12}$$

Indeed, for any $t \geq 0$ and $\mathrm{d}t > 0$,

$$\begin{aligned}\mathbb{E}[x_{t+\mathrm{d}t}|x_t] - x_t &= -x_t + (1 - I\mathrm{d}t)x_t + o(\mathrm{d}t)\\ &\quad + \mathrm{d}t\sum_{(ij)\in E} p_{ij}\big(x_t - \frac{K_{ij}}{p_{ij}}\nabla f_{ij}(x_{t-\tau_{ij}})\big)\\ &= -\mathrm{d}t\sum_{(ij)\in E} K_{ij}\nabla f_{ij}(x_{t-\tau_{ij}}) + o(\mathrm{d}t)\,.\end{aligned}$$

Taking the mean, dividing by $\mathrm{d}t$ and making $\mathrm{d}t \to 0$ leads to the delayed ODE verified by $y_t = \mathbb{E}[x_t]$. Such delay-differential ODEs are classical [48] yet their stability properties are notoriously hard to characterize. This is typically attacked by means of *Lyapunov-Krasovskii* functionals or *Lyapunov-Razumikhin* functions [22]. Alternatively, sufficient conditions for convergence and stability guarantees on (y_t) can be obtained, under specific conditions, by enforcing stability of the original system after *linearizing* it with respect to delays [43]. Linearizing in the sense of [43] means making the approximation $y_{t-\tau_{ij}} = y_t - \tau_{ij}\frac{\mathrm{d}y_t}{\mathrm{d}t}$. Under this approximation, we have:

$$\frac{\mathrm{d}y_t}{\mathrm{d}t} = -\sum_{(ij)\in E} K_{ij}\big(\nabla f_{ij}(y_t) - \tau_{ij}\nabla f_{ij}(\frac{\mathrm{d}y_t}{\mathrm{d}t})\big)\,.$$

For any weights ν_{ij} and vector z, $\sum_{ij}\nu_{ij}\nabla f_{ij}(z) = \Delta_G(\nu)z$. Thus the delay-linearized ODE reads

$$(\mathrm{Id} - \Delta_G(\{K_{ij}\tau_{ij}\}))\frac{\mathrm{d}y_t}{\mathrm{d}t} = -\Delta_G(\{K_{ij}\})y_t\,. \tag{13}$$

This delay-linearized ODE (13) provides intuition on the behavior of $\mathbb{E}[x_t]$. Indeed, (13) is stable provided that $\rho(\Delta_G(\{\tau_{ij}K_{ij}\})) < 1$, in which case it has a linear rate of convergence of order $\lambda_2(\Delta_G(\{K_{ij}\})$.

Even though this stability condition and the rate of convergence are only heuristics, since (13) is obtained through an approximation of the delayed ODE verified by $\mathbb{E}[x_t]$ (12), this stability condition for the delay-linearized system implies stability of the original delayed system under assumptions on the matrices and delays involved [43], that hold in our case, leading to the following

Proposition 1: Assume that the spectral radius of the weighted Laplacian $\Delta_G(\{\tau_{ij}K_{ij}\})$ verifies $\rho(\Delta_G(\{\tau_{ij}K_{ij}\})) < 1$. Then the delayed *ODE* (12) is stable.

Consequently, the stability conditions (necessary conditions on step sizes K_{ij} in Equation (10)) obtained in Theorem 1 are very natural. Indeed, a simple way to enforce $\rho(\Delta_G(\{\tau_{ij}K_{ij}\}) < 1$ based on local conditions consists in imposing $\sum_j \tau_{ij}K_{ij} < 1$ for all i. This is a weaker condition than the one stated in Theorem 1, but it only gives stability of the means. Furthermore, the rate of convergence of delayed randomized gossip in Theorem 1, that takes the form of the eigengap of a weighted graph Laplacian, is also that of any solution of the delay-linearized ODE (13).

Proof: For $A \in \mathbb{R}^{V\times E}$ as defined in Section II-A for non-null weights μ_{ij}, define the following delayed ODE:

$$\frac{\mathrm{d}\lambda_t}{\mathrm{d}t} = -\sum_{(ij)\in E}\frac{K_{ij}}{\mu_{ij}^2}e_{ij}^\top A^\top A\lambda_{t-\tau_{ij}}\,. \tag{14}$$

For (y_t) solution of (12), if there exists λ_0 such that $A\lambda_0 = y_0$, then $y_t = A\lambda_t$ for all t, where λ_t is solution of (14) initialized at the value λ_0. Then, since $AA^\top$ is the Laplacian of graph G with weights $\mu_{ij}^2 > 0$, A is of rank $n-1$. For all λ, $A\lambda$ is in the orthogonal of $\mathbb{R}\mathbf{1}$ ($\mathbf{1} \in \mathbb{R}^V$ is the vector with all entries equal to 1), so that $\mathrm{Im}(A)$ is exactly the orthogonal of $\mathbb{R}\mathbf{1}$. Finally, since for (y_t) a solution of (12), $y_t - (\mathbf{1}^\top y_0)\mathbf{1}$ is also solution of (12) and takes values in the orthogonal of $\mathbb{R}\mathbf{1}$, it is sufficient to prove stability of (14).

To that end, we use Theorem 1 of [43]. For $z \in \mathbb{R}^E$, let $D(z) \in \mathbb{R}^{E\times E}$ be the diagonal matrix with diagonal equal to z. Let $M = D(\frac{K}{\mu^2})A^\top A$. Then, the delayed ODE (14) writes as:

$$\frac{\mathrm{d}\lambda_t(ij)}{\mathrm{d}t} = -\sum_{(kl)\in E} M_{(ij),(kl)}\lambda_{t-\tau_{ij}}(kl) \quad , (ij) \in E\,,$$

and ODE that takes the same form as Equation (7) in [43], for $D^{\leftarrow}_{(ij)} = \tau_{ij}$, $D^{\rightarrow}_{(ij)} = 0$ and $D_{(ij)} = \tau_{ij}$, $R = E$ and with our matrix M. In order to ensure that M is symmetric and positive semi-definite, we take $\mu_{ij}^2 = K_{ij}$, to have $M = A^\top A$. The assumptions of Theorem 1 of [43] are verified, so that the delayed ODE (14) is table if $\rho(D(\tau)M) < 1$. We then write $\rho(D(\tau)M) = \rho(D(\sqrt{\tau})A^\top AD(\sqrt{\tau}) = \rho(AD(\sqrt{\tau})(AD(\sqrt{\tau}))^\top)$, and notice that $AD(\sqrt{\tau})(AD(\sqrt{\tau}))^\top$ is the Laplacian of graph G with weights $\mu_{ij}^2\tau_{ij} = K_{ij}\tau_{ij}$, concluding the proof. ■

E. Proof of Theorem 1

In the proof, we use the assumed bounds τ_{ij} on actual delays in our algorithm to ensure that communications between i and j started at a time $t - \tau_{ij}$ induce communication updates at time t. Our algorithms thus behave exactly as if individual communication delays coincide with these upper bounds τ_{ij}, which allows us to analyze algorithms with constant, albeit heterogeneous delays.

In contrast an analysis in discrete time would use a global iteration counter, and discrete-time delays would not be constant,making the analysis either much more involved or unable to capture the asynchronous speedup described above.

Proof: Theorem 1 is obtained by applying a general result on delayed coordinate descent in the continuized framework that we detail in Section III.

Specifically, we consider the function $g(\lambda) = \frac{1}{2}\|A\lambda\|^2$ for $\lambda \in \mathbb{R}^{E\times d}$ and $A \in \mathbb{R}^{V\times E}$ as defined in Section II-A. As in Section II-D, there exists $\lambda \in \mathbb{R}^{E\times d}$ such that $x_0 - \bar{x} = A\lambda$. Let $(\lambda_t)_{t\geq 0}$ be defined with $\lambda_0 = \lambda$, and the delayed coordinate gradient steps at the clock tickings of the *P.p.p.*'s:

$$\lambda_{T_k} \leftarrow \lambda_{T_k-} - \frac{K_{i_kj_k}}{p_{i_kj_k}}\nabla_{i_kj_k}g(\lambda_{T_k-\tau_{i_kj_k}})\,.$$

For all $t \geq 0$, we then have $x_t = \bar{x} + A\lambda_t$, where we recall that the process (x_t) follows the delayed randomized gossip updates (9) of Algorithm 1. Then, for all $t \geq 0$, we have $g(\lambda_t) = \frac{1}{2}\|A\lambda_t\|^2 = \frac{1}{2}\|x_t - \bar{x}\|^2$.

The result of Theorem 1 follows from a control of $\mathbb{E}\left[g(\lambda_t)\right]$ that is a direct consequence of Theorem 2 in next section with the specific choices $m = |E|$ and coordinate blocks corresponding to edges. The assumptions of Theorem 2 are verified with $L_{ij} = 2\mu_{ij}^2$, $M_{(ij),(kl)} = \sqrt{L_{ij}L_{kl}}$, and strong convexity parameter $\lambda_2(\Delta_G(\nu_{ij} = \mu_{ij}^2))$ for the specific choice $\mu_{ij}^2 = K_{ij}$, as is shown in Lemmas 2, 3, 4 in the Appendix, giving us exactly Theorem 1. ■

III. Delayed coordinate gradient descent in the continuized framework

Let J be a σ-strongly convex function on $\mathbb{R}^D$. For $k = 1, ..., m$, let E_k be a subspace of $\mathbb{R}^d$, and assume that:

$$\mathbb{R}^d = \bigoplus_{k=1}^m E_k\,, \tag{15}$$

where $\oplus$ denotes a direct sum of linear spaces. For $x \in \mathbb{R}^D$, let x_k denote its orthogonal projection on E_k and let $\nabla_k J := (\nabla J)_k$, and assume that the subspaces $E_1, ..., E_m$ are orthogonal. For $k, \ell \in [m]$, we say that k and ℓ are **adjacent** and we write $k \sim \ell$ if and only if $\nabla_k\nabla_\ell J = \nabla^2_{kl}J$ is not identically constant equal to 0. This induces a symmetric graph structure on the coordinates $k \in [m]$. In the context of gossip network averaging, $m = |E|$ and each subspace E_k corresponds to an edge $e_k = (i_kj_k)$ of the graph; in that context, we have $k \sim \ell$ if and only if edges e_k and e_ℓ share a node.

In the network averaging problem previously described, the function J used is $g(\lambda) = \frac{1}{2}\|A\lambda\|$ for $\lambda \in \mathbb{R}^{E\times d}$ the edge variables. Subspaces are E_{ij} of dimension d for $(ij) \in E$ (and $m = |E|$) corresponding to variables of λ associated to edge (ij).

A. Algorithm and assumptions

1) Continuized delayed coordinate gradient descent algorithm: For $k \in [m]$, let $\mathcal{P}_k$ be a *P.p.p.* of intensity p_k denoting the times at which an update can be performed on subspace E_k. For $t \in \mathcal{P}_k$ let $\varepsilon_k(t) \in \{0, 1\}$ be the indicator of whether the update is performed or not. Let also η_k be some positive step size for $k \in [m]$. Consider then the following continuous-time process $X(t)$, where $X_k(t)$ (the projection of $X(t)$ on E_k) evolves according to:

$$\mathrm{d}X_k(t) = -\varepsilon_k(t)\eta_k\nabla_k J((X(t-\tau_k))\mathcal{P}_k(\mathrm{d}t)\,, \tag{16}$$

where $\mathcal{P}_k(\mathrm{d}t)$ corresponds to a Dirac at the points of the *P.p.p.* $\mathcal{P}_k$. In words, $(X(t))_{t\geq 0}$ is a jump process that takes coordinate gradient descent steps along subspaces $(E_k)_{k\in[m]}$ at the times of independent Poisson point processes $(\mathcal{P}_k)_{k\in[m]}$. We introduced variables $(\varepsilon_k(t))_{k\in[m],t\in\mathcal{P}_k}$ with values in $\{0, 1\}$ to represent capacity constraints: $\varepsilon_k(t) = 0$ if the update at time $t \in \mathcal{P}_k$ cannot be performed due to some constraint saturation; these variables $\varepsilon_k(t)$ will be essential in our treatment of communication and computation capacity constraints in Section V.

2) Regularity assumptions: J is σ-strongly convex, and L_k-smooth on E_k for $k \in [m]$. Furthermore, there exist non-negative real numbers $M_{k,\ell}$ and $M_{\ell,k}$ for $k \sim \ell$ such that for all $k = 1, ..., m$ and $x, y \in \mathbb{R}^D$, we have:

$$\|\nabla_k J(x) - \nabla_k J(y)\| \leq \sum_{\ell\sim k} M_{k,\ell}\|x_\ell - y_\ell\|\,. \tag{17}$$

When J is L_k smooth on E_k as we assume, the above condition is verified by the choice $M_{i,j} = L_j$, $i \sim j$. If $\nabla_k J$ is M_k-Lipschitz, Condition (17) is verified by the choice $M_{k,\ell} = M_k$. Assumption (17) however allows for more freedom, and is particularly well suited for our analysis. In particular for decentralized optimization, it will be convenient to take $M_{k\ell} = \sqrt{L_k L_\ell}$.

3) Assumptions on variables $\varepsilon_k(t)$, $t \in \mathcal{P}_k$: For $t \in \mathcal{P}_k$, random variable $\varepsilon_k(t)$ is $\sigma\big(\mathcal{P}_\ell \cap [t-\tau_k, t), \ell \in [m]\big)$-measurable, and there exists a constant $\varepsilon_k > 0$ such that:

$$\mathbb{E}\left[\varepsilon_k(t)\right] \geq \varepsilon_k \,,$$

Furthermore, we assume that $\varepsilon_k(t)$ is negatively correlated with each quantity $N_\ell(t-\tau_k, t) = |\mathcal{P}_\ell \cap [t-\tau_k, t]|$, *i.e.* that for all $k, \ell \in [m]$,

$$\mathbb{E}\left[\varepsilon_k(t) N_\ell(t-\tau_k, t)\right] \leq \mathbb{E}\left[\varepsilon_k(t)\right] \mathbb{E}\left[N_\ell(t-\tau_k, t)\right] \,. \quad (18)$$

In our subsequent treatment of communication and capacity contraints, we shall see that the above assumptions are verified for $\epsilon_k(t)$ the indicator that t is a point a *truncated P.p.p.* $\tilde{\mathcal{P}}_k$ defined as follows:

*Definition 2 (**Truncated** **P.p.p.**):* Let $(\mathcal{P}_k)_{1\leq k\leq m}$ be *P.p.p.* of respective intensities $(p_k)_{1\leq k\leq m}$, $(\tau_k)_{1\leq k\leq m}$ non-negative delays. Let N_k be the Poisson point measures associated to $\mathcal{P}_k$, $k \in [m]$. For $(\mathcal{C}_r)_{1\leq r\leq M}$ subsets of $[m]$, we define the truncated Poisson point measures $(\tilde{N})_{1\leq k\leq m}$ of intensities $(p_k)_{1\leq k\leq m}$ and parameters $(\tau_k)_k, (q_{k,r})_{k\in[m], r\in[M]}$ as:

$$\mathrm{d}\tilde{N}_k(t) = 1_{\left\{\bigcap_{1\leq k\leq M}\left\{\sum_{\ell\in\mathcal{C}_r} N_\ell([t-\tau_k, t)) \leq q_{k,r}\right\}\right\}} \mathrm{d}N_k(t) \,, \quad (19)$$

and we let $\tilde{\mathcal{P}}_k$ be the point process associated to this point measure.

B. Convergence guarantees and analysis

The main result of this Section is the following

*Theorem 2 (**Delayed Coordinate Gradient Descent**):* Under the stated assumptions on regularity of G and on variables $\epsilon_k(t)$, assume further that the step sizes η_k are given by $\eta_k = \frac{K_k}{p_k L_k}$ where for all $k \in [m]$,

$$K_k \leq \frac{p_k}{1 + \sum_{\ell\sim k} p_\ell \left(\frac{\tau_k M_{k,\ell} + e\tau_\ell M_{\ell,k}}{\sqrt{L_k L_\ell}}\right)} \,, \quad (20)$$

and let $\gamma \in \mathbb{R}_+$ be such that:

$$\gamma < \min\left(\sigma \min_k \frac{\varepsilon_k K_k}{L_k}, \frac{1}{\tau_{\max}}\right) \,, \quad (21)$$

where $\tau_{\max} := \max_{k\in[m]} \tau_k$. Then for any $T > 0$ the solution $X(t)$ to Equation (16) verifies

$$\frac{\int_0^T e^{\gamma t} \mathbb{E}\left[J(X(t)) - J(x^\star)\right] \mathrm{d}t}{\int_0^T e^{\gamma t}\big(J(X(0)) - J(x^\star)\big)\mathrm{d}t} \leq e^{-\frac{\gamma T}{2}} \frac{1 + \frac{\tau_{\max}}{T}}{1 - \gamma\tau_{\max}} \,. \quad (22)$$

Proof: We proceed in three steps. The first step consists in upper bounding, for $t \geq 0$, the quantity $\frac{\mathrm{d}\mathbb{E}[J(X(t))]}{\mathrm{d}t}$. We then introduce in Step 2 a Lyapunov function inspired by the Lyapunov-Krasovskii functional [22]), and by using the result proved in the first step, we show that it verifies a delayed ordinary differential inequality. The last step then consists in deriving the desired result from this delayed differential inequality.

Step 1: To bound $\frac{\mathrm{d}\mathbb{E}[J(X(t))]}{\mathrm{d}t}$, we study infinitesimal increments between t and $t + \mathrm{d}t$ for $\mathrm{d}t \to 0$. This approach is justified by results on *stochastic ordinary differential equation with Poisson jumps*, see [14]. For $t \geq 0$, let $\mathcal{F}_t$ be the filtration induced by $\mathcal{P}_k \cap [0, t)$, $k \in [m]$ *i.e.*, the filtration up to time t. By convention, for non-positive t, we write $X(t) = X(0)$. The following inequalities are written up to $o(\mathrm{d}t)$ terms, that we omit to lighten notations. Finally, we write

$$g_{k,t} = \nabla_k J(X(t)) \,, \quad k \in [m], t \geq 0 \,.$$

We have, using local smoothness properties of G and the fact that for a *P.p.p.* $\mathcal{P}$ of intensity p, $\mathbb{P}(\mathcal{P} \cap [t, t+\mathrm{d}t] = \emptyset) = 1 - p\mathrm{d}t + o(\mathrm{d}t)$ and $\mathbb{P}(\#\mathcal{P} \cap [t, t+\mathrm{d}t] = 1) = p\mathrm{d}t + o(\mathrm{d}t)$:

$$\begin{aligned}
&\frac{\mathbb{E}\left[J(X(t+\mathrm{d}t)) - J(X(t))|\mathcal{F}_t\right]}{\mathrm{d}t} \\
&= \sum_{k=1}^m p_k \left(G\left(X(t) - \frac{\varepsilon_k(t) K_k}{p_k L_k} g_{k,t-\tau_k}\right) - J(X(t))\right) \\
&\leq \sum_{k=1}^m p_k \left(-\frac{K_k}{p_k L_k} \langle \varepsilon_k(t) g_{k,t-\tau_k}, g_{k,t}\rangle \right. \\
&\quad \left. + \frac{L_k}{2} \left\| \varepsilon_k(t) \frac{K_k}{p_k L_k} \nabla_k g_{k,t-\tau_k} \right\|^2 \right) \,.
\end{aligned}$$

First, we rewrite $-\frac{\varepsilon_k(t) K_k}{p_k L_k} \langle g_{k,t-\tau_k}, g_{k,t}\rangle$ as

$$-\frac{\varepsilon_k(t) K_k}{p_k L_k} \|g_{k,t-\tau_k}\|^2 - \frac{\varepsilon_k(t) K_k}{p_k L_k} \langle g_{k,t-\tau_k}, g_{k,t} - g_{k,t-\tau_k}\rangle,$$

and bound the second term there by

$$\begin{aligned}
&-\frac{\varepsilon_k(t) K_k}{p_k L_k} \langle g_{k,t-\tau_k}, g_{k,t} - g_{k,t-\tau_k}\rangle \\
&\leq \frac{\varepsilon_k(t) K_k}{p_k L_k} \|g_{k,t-\tau_k}\| \, \|g_{k,t} - g_{k,t-\tau_k}\| \\
&\leq \frac{K_k}{p_k L_k} \|\varepsilon_k(t) g_{k,t-\tau_k}\| \sum_{\ell\sim k} M_{k,\ell} \|X_\ell(t) - X_\ell(t-\tau_k)\| \,,
\end{aligned}$$

where we used the Cauchy-Schwarz inequality and then local Lipschitz property (17) of $\nabla_k G$. Writing

$$\begin{aligned}
&\|X_\ell(t) - X_\ell(t-\tau_k)\| \\
&= \left\| \int_{(t-\tau_k)^+}^t \frac{\varepsilon_\ell(s) K_\ell}{p_\ell L_\ell} g_{\ell,s-\tau_\ell} N_\ell(\mathrm{d}s) \right\| \,,
\end{aligned}$$

where N_ℓ is the Poisson point measure associated to $\mathcal{P}_\ell$, we have (where we use a triangle inequality for integrals):

$$\begin{aligned}
&\frac{K_k M_{k,\ell}}{p_k L_k} \mathbb{E}\left[\|\varepsilon_k(t) g_{k,t-\tau_k}\| \|X_\ell(t) - X_\ell(t-\tau_k)\|\right] \\
&\leq \mathbb{E}\left[\int_{(t-\tau_k)^+}^t M_{k,\ell} \frac{\varepsilon_k(t) K_k \varepsilon_\ell(s) K_\ell}{L_k p_k p_\ell L_\ell} \|g_{k,t-\tau_k}\| \|g_{\ell,s-\tau_\ell}\| N_\ell(\mathrm{d}s)\right] \\
&\leq \mathbb{E}\left[\int_{(t-\tau_k)^+}^t \frac{1}{2} \left(\frac{K_k^2 M_{k,\ell}}{p_k^2 L_k \sqrt{L_k L_\ell}} \|\varepsilon_k(t) g_{k,t-\tau_k}\|^2 \right.\right. \\
&\quad \left.\left. + \frac{K_\ell^2 M_{k,\ell}}{p_\ell^2 L_\ell \sqrt{L_k L_\ell}} \|\varepsilon_\ell(s) g_{\ell,s-\tau_\ell}\|^2\right) N_\ell(\mathrm{d}s)\right] \,.
\end{aligned}$$

For the first term, since both $\varepsilon_k(t)$ and $N_\ell(\mathrm{d}s)$ for s in the integral are independent from $X(t-\tau_k)$ (and thus from

$g_{k,t-\tau_k}$), and where we write $N_\ell(u,v)$ the number of clock tickings of $\mathcal{P}_\ell$ in the interval $[u,v)$, we obtain:

$$\mathbb{E}\left[\int_{(t-\tau_k)^+}^{t} \frac{1}{2}\frac{K_k^2 M_{k,\ell}}{p_k^2 L_k\sqrt{L_k L_\ell}}\|\varepsilon_k(t)g_{k,t-\tau_k}\|^2\right] = \frac{\mathbb{E}\left[N_\ell(t-\tau_k,t)\varepsilon_k(t)\right]}{2}\frac{K_k^2 M_{k,\ell}}{p_k^2 L_k\sqrt{L_k L_\ell}}\mathbb{E}\left[\|g_{k,t-\tau_k}\|^2\right].$$

Furthermore, using our negative correlation assumption, $\mathbb{E}\left[N_\ell(t-\tau_k,t)\varepsilon_k(t)\right] \leq \mathbb{E}\left[N_\ell(t-\tau_k,t)\right]\mathbb{E}\left[\varepsilon_k(t)\right] = p_\ell\tau_k\mathbb{E}\left[\varepsilon_k(t)\right]$, and since $\varepsilon_k(t)$ and $g_{k,t-\tau_k}$ are independent, $\mathbb{E}\left[\varepsilon_k(t)\right]\mathbb{E}\left[\|g_{k,t-\tau_k}\|^2\right] = \mathbb{E}\left[\varepsilon_k(t)\|g_{k,t-\tau_k}\|^2\right]$.

For the second term, since the process $(\varepsilon_\ell(s)g_{\ell,s-\tau_\ell})_s$ is predictable (in the sense that it is independent from $N_u(\mathrm{d}s)$ for all u), we have

$$\begin{aligned}&\mathbb{E}\left[\int_{(t-\tau_k)^+}^{t}\frac{K_\ell^2 M_{k,\ell}}{2p_\ell^2 L_\ell\sqrt{L_k L_\ell}}\|\varepsilon_\ell(s)g_{\ell,s-\tau_\ell}\|^2 N_\ell(\mathrm{d}s)\right]\\ &= \int_{(t-\tau_k)^+}^{t}\frac{K_\ell^2 M_{k,\ell}}{2p_\ell^2 L_\ell\sqrt{L_k L_\ell}}\mathbb{E}\left[\|\varepsilon_\ell(s)g_{\ell,s-\tau_\ell}\|^2\right]\mathbb{E}\left[N_\ell(\mathrm{d}s)\right]\\ &= \int_{(t-\tau_k)^+}^{t}\frac{K_\ell^2 M_{k,\ell}}{2p_\ell^2 L_\ell\sqrt{L_k L_\ell}}\mathbb{E}\left[\|\varepsilon_\ell(s)g_{\ell,s-\tau_\ell}\|^2\right]p_\ell\mathrm{d}s\,.\end{aligned}$$

Hence,

$$\begin{aligned}&\frac{K_k M_{k,\ell}}{p_k L_k}\mathbb{E}\left[\|\varepsilon_k(t)g_{k,t-\tau_k}\|\|X_\ell(t)-X_\ell(t-\tau_k)\|\right]\\ &\leq \frac{p_\ell\tau_k K_k^2 M_{k,\ell}}{2p_k^2 L_k\sqrt{L_k L_\ell}}\mathbb{E}\left[\|\varepsilon_k(t)g_{k,t-\tau_k}\|^2\right]\\ &\quad+\int_{(t-\tau_k)^+}^{t}\frac{K_\ell^2 M_{k,\ell}}{2p_\ell^2 L_\ell\sqrt{L_k L_\ell}}\mathbb{E}\left[\|\varepsilon_\ell(s)g_{\ell,s-\tau_\ell}\|^2\right]p_\ell\mathrm{d}s\,.\end{aligned}$$

Combining all our elements and taking $\mathrm{d}t\to 0$, we hence have:

$$\begin{aligned}\frac{\mathrm{d}\mathbb{E}\left[J(X(t))\right]}{\mathrm{d}t} &\leq -\sum_{k=1}^{m}\frac{K_k}{L_k}\left(1-\frac{K_k}{2p_k}\right)\mathbb{E}\left[\|\varepsilon_k(t)g_{k,t-\tau_k}\|^2\right]\\ &+\sum_{k=1}^{m}\sum_{\ell\sim k}\frac{p_\ell\tau_k K_k^2 M_{k,\ell}}{2p_k L_k\sqrt{L_k L_\ell}}\mathbb{E}\left[\|\varepsilon_k(t)g_{k,t-\tau_k}\|^2\right]\\ &+\sum_{k=1}^{m}\sum_{\ell\sim k}\int_{(t-\tau_k)^+}^{t}\frac{p_k K_\ell^2 M_{k,\ell}}{2p_\ell L_\ell\sqrt{L_k L_\ell}}\mathbb{E}\left[\|\varepsilon_\ell(s)g_{\ell,s-\tau_\ell}\|^2\right]\mathrm{d}s\,.\end{aligned}\tag{23}$$

Step 2: Now, introduce the following Lyapunov function:

$$\mathcal{L}_T^\gamma = \int_0^T e^{\gamma t}\mathbb{E}\left[J(X(t))-J(x^\star)\right]\mathrm{d}t\,,$$

that we wish to upper-bound by some constant, where γ is as in (21). We have:

$$\frac{\mathrm{d}\mathcal{L}_T^\gamma}{\mathrm{d}T} = J(X(0))-J(x^\star)+\gamma\mathcal{L}_T^\gamma+\int_0^T e^{\gamma t}\frac{\mathrm{d}\mathbb{E}\left[J(X(t))\right]}{\mathrm{d}t}\mathrm{d}t\,.$$

Integrating the bound (23) on $\frac{\mathrm{d}\mathbb{E}[J(X(t))]}{\mathrm{d}t}$, we obtain, using $\int_0^T\int_{(t-\tau)^+}^t h(u)\mathrm{d}u\mathrm{d}t \leq \tau\int_0^T h(t)\mathrm{d}t$ for non-negative h:

$$\begin{aligned}\frac{\mathrm{d}\mathcal{L}_T^\gamma}{\mathrm{d}T} &\leq J(X(0))-J(x^\star)+\gamma\mathcal{L}_T^\gamma\\ &-\sum_{k=1}^{m}\frac{K_k}{L_k}\left(1-\frac{K_k}{2p_k}\right)\int_0^T e^{\gamma t}\mathbb{E}\left[\|\varepsilon_k(t)g_{k,t-\tau_k}\|^2\right]\mathrm{d}t\\ &+\sum_{k=1}^{m}A_k\int_0^T e^{\gamma t}\mathbb{E}\left[\|\varepsilon_k(t)g_{k,t-\tau_k}\|^2\right]\mathrm{d}t\,,\end{aligned}$$

where

$$A_k = \frac{K_k^2}{2p_k L_k}\sum_{\ell\sim k}\frac{p_\ell\tau_k M_{k,\ell}}{\sqrt{L_k L_\ell}}+e^{\gamma\tau_\ell}\frac{p_\ell\tau_\ell M_{\ell,k}}{\sqrt{L_k L_\ell}}\,.$$

Remark now that we have

$$\frac{K_k^2}{2p_k L_k}+A_k\leq\frac{K_k}{2L_k},\quad k\in[m].\tag{24}$$

Indeed, (24) is equivalent to

$$K_k\leq\frac{p_k}{1+\sum_{\ell\sim k}\left(\frac{p_\ell\tau_k M_{k,\ell}}{\sqrt{L_k L_\ell}}+e^{\gamma\tau_\ell}\frac{p_\ell\tau_\ell M_{\ell,k}}{\sqrt{L_k L_\ell}}\right)}\,,$$

which follows from the assumed bounds (20) on K_k and the fact that $\gamma\leq 1/\tau_{\max}$, assumed in (21). we then have, using (24) and the fact that, by strong convexity, $J(X(t))-J(x^\star)\leq\frac{1}{2\sigma}\|\nabla J(X(t))\|^2=\frac{1}{2\sigma}\sum_{k=1}^m\|\nabla g_{k,t}\|^2$:

$$\begin{aligned}\frac{\mathrm{d}\mathcal{L}_T^\gamma}{\mathrm{d}T} &\leq J(X(0))-J(x^\star)+\gamma\mathcal{L}_T^\gamma\\ &-\sum_{k=1}^{m}\frac{K_k}{2L_k}\int_0^{T-\tau_k}e^{\gamma(t+\tau_k)}\mathbb{E}\left[\|\varepsilon_k(t+\tau_k)g_{k,t}\|^2\right]\mathrm{d}t\\ &\leq J(X(0))-J(x^\star)+\gamma\mathcal{L}_T^\gamma\\ &-\min_{k\in[m]}\left(\frac{K_k\varepsilon_k e^{\gamma\tau_k}}{2L_k}\right)\int_0^{T-\tau_{\max}}e^{\gamma t}\mathbb{E}\left[\sum_{k=1}^m\|g_{k,t}\|^2\right]\mathrm{d}t\\ &\leq J(X(0))-J(x^\star)+\gamma\left(\mathcal{L}_T^\gamma-\mathcal{L}_{T-\tau_{\max}}^\gamma\right),\end{aligned}$$

where we used the assumption (21) that $\gamma\leq\sigma\min_{k\in[m]}\left(\frac{K_k\varepsilon_k}{L_k}\right)$.

Step 3: The proof is then concluded by using the following lemma, to control solutions of this delayed ordinary differential inequality.

Lemma 1: Let $h:\mathbb{R}\to\mathbb{R}^+$ a differentiable function such that:

$$\begin{aligned}&\forall t\leq 0\,,\ h(t)=0\,,\\ &\forall t\geq 0\,,\ h'(t)\leq a+b(h(t)-h(t-\tau))\,,\end{aligned}$$

for some positive constants a,b,τ verifying $\tau b<1$. Then:

$$\forall t\in\mathbb{R}\,,\quad h(t)\leq\frac{a(t+\tau)}{1-\tau b}\,.$$

Proof: Let $\delta(t)=h(t)-h(t-\tau)$. For any $t\geq 0$, we have:

$$\begin{aligned}\delta(t) &= \int_{t-\tau}^t h'(s)\mathrm{d}s\\ &\leq\int_{t-\tau}^t(a+b\delta(s))\mathrm{d}s\,.\end{aligned}$$

Let $c = \frac{\tau a}{1-\tau b}$ (solution of $x = \tau(a+bx)$) and $t_0 = \inf\{t > 0 | \delta(t) \geq c\} \in \mathbb{R} \cup \{\infty\}$. Assume that t_0 is finite. Then, $\delta(t) < c$ for $t < t_0$ and by continuity $\delta(t_0) = c$, so that:

$$c = \delta(t_0) \leq \int_{t_0-\tau}^{t_0} (a + b\delta(s))\mathrm{d}s$$
$$< \int_{t_0-\tau}^{t_0} (a+bc)\mathrm{d}s = \tau(a+bc) = c\,,$$

as for all $s < t_0$, $\delta(s) < c$. This is absurd, and thus t_0 is not finite: $\forall t > 0, \delta(t) < c$, giving us for all $t \geq 0$ $h'(t) \leq a + bc$ and thus $h(t) \leq c(t+\tau)/\tau$. ∎

To conclude the proof of Theorem (2), we apply Lemma 1 to $h(T) = \mathcal{L}_T^\gamma$ with $a = J(X(0)) - J(x^\star)$, $b = \gamma$ and $\tau = \tau_{\max}$ to obtain that for all $T > 0$,

$$\mathcal{L}_T^\gamma \leq \big(J(X(0)) - J(x^\star)\big)\frac{T + \tau_{\max}}{1 - \tau_{\max}\gamma}\,.$$

The result of Theorem 2 follows by dividing this inequality by $\int_0^T e^{\gamma t}\mathrm{d}t = \frac{e^{\gamma T}-1}{\gamma}$:

$$\frac{\int_0^T e^{\gamma t}\mathbb{E}\left[J(X(t)) - J(x^\star)\right]\mathrm{d}t}{\int_0^T e^{\gamma t}\big(J(X(0)) - J(x^\star)\big)\mathrm{d}t} \leq \frac{\gamma}{e^{\gamma T}-1}\frac{T+\tau_{\max}}{1-\tau_{\max}\gamma}$$
$$= \frac{\gamma T}{e^{\gamma T}-1}\frac{1+\tau_{\max}/T}{1-\tau_{\max}\gamma}$$
$$\leq e^{-\gamma T/2}\frac{1+\tau_{\max}/T}{1-\tau_{\max}\gamma}\,,$$

where we used that for $x \geq 0$, $\frac{e^x-1}{x} \geq e^{x/2}$. ∎

IV. Extension to decentralized optimization

Using Theorem 2, we are now armed to generalize the delayed randomized gossip algorithm and analysis to more general settings. In this section we extend our results to decentralized optimization, going beyond the quadratic objective functions considered network averaging.

A. Delayed Decentralized Optimization

Consider the decentralized optimization problem (1). We make the following assumptions on the individual objective functions f_i therein :

each f_i, $i \in V$, is σ-strongly convex and L-smooth, (25)

see [10] for definitions. Let $f(z) := \sum_{i\in[n]} f_i(z)$ for $z \in \mathbb{R}^d$ and $F(x) = \sum_{i\in[n]} f_i(x_i)$ for $x = (x_1, \cdots, x_n) \in \mathbb{R}^{n\times d}$ where $x_i \in \mathbb{R}^d$ corresponds to node $i \in [n]$.

*Definition 3 (**Fenchel Conjugate**):* For any function $g : \mathbb{R}^p \to \mathbb{R}$, its *Fenchel conjugate* is denoted by g^* and defined on $\mathbb{R}^p$ by $g^*(y) = \sup_{x\in\mathbb{R}^p}\langle x, y\rangle - g(x) \in \mathbb{R} \cup \{+\infty\}$.

Our algorithm for delayed decentralized optimization is built on delayed randomized gossip for network averaging, augmented with local computations. Each node $i \in V$ keeps two local variables: the communication variable $x_i(t)$, used to run delayed randomized gossip, and a computation variable $y_i(t)$, used to make local computation updates in the following way.

Local computations. Each node i generates a *Poisson point process* $\mathcal{P}_i^{\mathrm{comp}} = \{T_1^{\mathrm{comp}}(i) < T_2^{\mathrm{comp}}(i), \ldots\}$ of intensity p_i^{comp}. At the clock tickings $T_k^{\mathrm{comp}}(i)$, a local computation update is made corresponding to a computation started at a time $T_k^{\mathrm{comp}}(i) - \tau_i^{\mathrm{comp}}$, where τ_i^{comp} is the upper bound on the time to perform an elementary computation at node i, introduced in Assumption 1. Thus by assumption the computation started at time $T_k^{\mathrm{comp}}(i) - \tau_i^{\mathrm{comp}}$ is completed by time $T_k^{\mathrm{comp}}(i)$ so that the update can be performed at that time. The precise form of this update is given by Equation (29).

Communications. In parallel of these local computations, a *Delayed Randomized Gossip* is run on the graph. Dedicated *P.p.p.* $(\mathcal{P}_{ij})_{(ij)\in E}$ with respective intensities $(p_{ij})_{(ij)\in E}$ are associated to communication updates of all network edges, and used to perform updates as prescribed by Equation (9) in *Delayed Randomized Gossip*.

The resulting Delayed Decentralized Optimization algorithm, or *DDO* for short, is described in Algorithm 3 from a local viewpoint and is a combination of Algorithm 1 for communication updates along edges $(ij) \in E$ with Algorithm 2 for local computation updates at nodes $i \in V$.

From a global viewpoint, the algorithm is generated as follows. Let $\{T_k\}_{k\geq 0}$ be a *P.p.p.* process of intensity $\sum_{(ij)\in E} p_{ij} + \sum_{i\in V} p_i^{\mathrm{comp}}$. For all $k \geq 0$, at time k a communication or a computation update is performed. With probability proportional to p_{ij} the communication update (9) is performed along edge (ij) ($T_k = T_\ell(ij)$ in that case), with probability proportional to p_i^{comp} the computation update (29) is performed at node i ($T_k = T_\ell^{\mathrm{comp}}(i)$ in that case).

B. Convergence guarantees

The process $(x(t), y(t)) \in \mathbb{R}^{2n\times d}$ defined by algorithm *DDO*, Algorithm 3, satisfies the following convergence guarantees that generalize Theorem 1 to decentralized optimization beyond the case of quadratic functions.

*Theorem 3 (**Delayed Decentralized Optimization**):* Under the regularity assumptions (25), assume further that for all $1 \leq i \in V$ and $(ij) \in E$, we have:

$$K_{ij} \leq \frac{p_{ij}}{1 + \sum_{(kl)\sim(ij)} p_{kl}\big(\tau_{ij} + e\tau_{kl}\big)}$$
$$K_i^{\mathrm{comp}} \leq \frac{p_i^{\mathrm{comp}}}{1 + \sum_{j\sim i} p_{ij}\big(\tau_i^{\mathrm{comp}} + e\tau_{ij}\big)}\,. \tag{26}$$

Let $\tau_{\max} := \max\left(\max_{(ij)\in E}\tau_{ij}, \max_{i\in V}\tau_i^{\mathrm{comp}}\right)$. Then for $\gamma > 0$ such that

$$\gamma \leq \min\left(\frac{\sigma}{4L}\lambda_2\big(\Delta_G(K)\big)\,,\ \frac{1}{\tau_{\max}}\right)\,, \tag{27}$$

the process $(x(t), y(t))$ generated by *DDO* satisfy

$$\frac{\int_0^T e^{\gamma t}\mathbb{E}\left[\left\|\frac{\sigma}{2}x(t) - \bar{x}^\star\right\|^2\right]\mathrm{d}t}{\int_0^T e^{\gamma t}\left\|\frac{\sigma}{2}x(0) - \bar{x}^\star\right\|^2\mathrm{d}t} \leq e^{-\frac{\gamma T}{2}}\frac{L}{\sigma}\frac{1+\frac{\tau_{\max}}{T}}{1-\gamma\tau_{\max}}\,, \tag{28}$$

where $\bar{x}^\star = (x^\star, \ldots, x^\star)^\top \in \mathbb{R}^{n\times d}$ for $x^\star$ minimizer of $f = \sum_i f_i$.

DDO is based on a dual formulation and uses an augmented graph representation introduced in [25] to decouple computations from communications, as detailed in the proof. The dual gradient computations in Algorithm 2 can be expensive in general; they could be avoided by using a primal-dual approach for the computation updates [33].

The convergence guarantees we obtain resemble classical ones: Interpreting γ as the reciprocal of the time scale for convergence, we recognize in its upper bound (27) an "optimization factor" $\kappa_{\text{comp}}^{-1} := \sigma/L$, and a "communication factor" $\kappa_{\text{comm}}^{-1} = \lambda_2\big(\Delta_G(K)\big)$. Our method is non-accelerated, so the computation factor κ_{comp}, the condition number of the optimization problem, is expected. The communication factor captures the delay heterogeneity in the graph as in *Delayed Randomized Gossip*, leading to the *asynchronous speedup* discussed in Section II after Theorem 1.

Previous approaches have considered accelerating decentralized optimization by obtaining $\sqrt{\kappa_{\text{comp}}}$ instead of κ_{comp} and/or $\sqrt{\kappa'_{\text{comm}}}$ instead of κ'_{comm} for κ'_{comm} a communication factor in the rate of convergence [24], [34], [51]. Our result yields a speedup of a different nature: we obtain a communication factor κ_{comm} that can be arbitrarily larger than previously considered κ'_{comm} for networks with huge delay heterogeneity.

Algorithm 2 Local computations, node i

1: Step size $K_i^{\text{comp}} > 0$
2: Initialization $x_0(i) = y_0(i) = 0$
3: Initialization $T_1^{\text{comp}}(i) \sim \text{Exp}(p_i^{\text{comp}})$
4: **for** $\ell = 1, 2, \dots$ **do**
5: $\quad T_{\ell+1}^{\text{comp}}(i) = T_\ell^{\text{comp}}(i) + \text{Exp}(p_i^{\text{comp}})$.
6: **end for**
7: **for** $\ell = 1, 2, \dots$ **do**
8: $\quad$ At time $T_\ell^{\text{comp}}(i) - \tau_i^{\text{comp}}$, node i computes $g_i = \nabla\phi_i^*(y_i(T_\ell^{\text{comp}}(i) - \tau_i^{\text{comp}}))$ (takes a time less than τ_i^{comp}) and keeps $\hat{x}_i = x_i(T_\ell^{\text{comp}}(i) - \tau_i^{\text{comp}})$ in memory, where $\phi_i = f_i - \frac{\sigma}{4}\|.\|^2$.
9: $\quad$ At time $T_\ell^{\text{comp}}(i)$,

$$y_i(T_\ell^{\text{comp}}(i)) \overset{t}{\leftarrow} y_i(T_\ell^{\text{comp}}(i)-) - \frac{\sigma K_i^{\text{comp}}}{p_i^{\text{comp}}}\big(g_i - \hat{x}_i\big)\,,$$
$$x_i(T_\ell^{\text{comp}}(i)) \overset{t}{\leftarrow} x_i(T_\ell^{\text{comp}}(i)-) - \frac{K_i^{\text{comp}}}{2p_i^{\text{comp}}}\big(\hat{x}_i - g_i\big)\,. \tag{29}$$

10: **end for**

Algorithm 3 *DDO*

1: Node initializations $x_0(i) = y_0(i) = 0$, $i = 1, \dots, n$
2: **for** $i \in V$ and $(ij) \in E$, asynchronously, in parallel **do**
3: $\quad$ Communication updates along edge (ij) according to Algorithm 1
4: $\quad$ Local computation updates at node i according to Algorithm 2
5: **end for**
6: **Output:** $\frac{\sigma}{2}x_i(t)$ at time t and node i.

C. Proof of Theorem 3

Proof: Following the augmented graph approach [25], for each "physical" node $i \in V$, we associate a "virtual" node i^{comp}, corresponding to the computational unit of node i. We then consider the augmented graph $G^+ = (V^+, E^+)$, where $V^+ = V \cup V^{\text{comp}}$ (for $V^{\text{comp}} = \{i^{\text{comp}}, i \in V\}$) and $E^+ = E \cup E^{\text{comp}}$ (for $E^{\text{comp}} = \{(ii^{\text{comp}}), i \in V\}$).

For $i \in V$, function f_i is then split (using σ-strong convexity) into a sum of two $\sigma/2$-strongly convex functions: $f_i = \phi_i + \phi_{i^{\text{comp}}}$ where $\phi_{i^{\text{comp}}}(x_i) = f_i(x_i) - \frac{\sigma}{4}\|x_i\|^2$ and $\phi_i(x_i) = \phi_{\text{comm}}(x_i) = \frac{\sigma}{4}\|x_i\|^2$.

The optimization objective (1)

$$\min_{x_1=\dots=x_n} \frac{1}{n}\sum_{i=1}^n f_i(x_i)\,, \quad x = (x_1, \dots, x_n) \in \mathbb{R}^{V\times d}$$

can then be rewritten as

$$\min_{x\in\mathbb{R}^{V^+}} \left\{ F(x) = \sum_{i\in V}\phi_i(x_i) + \sum_{i^{\text{comp}}\in V^{\text{comp}}} \phi_{i^{\text{comp}}}(x_{i^{\text{comp}}}) \right\},$$

under the constraint $x_i = x_j$ for $(ij) \in E^+$. This constraint can then be rewritten as $A^\top x = 0$ for $A \in \mathbb{R}^{E^+\times V^+}$ such that for all $(ij) \in E^+$, $Ae_{ij} = \mu_{ij}(e_i - e_j)$, as was done for network averaging, considering the augmented graph instead of the original graph. Using Lagrangian duality, denoting $F_A^*(\lambda) := F^*(A\lambda)$ for $\lambda \in \mathbb{R}^{E^+\times d}$ where F^* is the Fenchel conjugate of F, we have:

$$\min_{x\in\mathbb{R}^{V^+\times d}, x_i=x_j, (ij)\in E^+} F(x) = \max_{\lambda\in\mathbb{R}^{E\times d}} -F_A^*(\lambda).$$

Thus $F_A^*(\lambda)$ is to be minimized over the dual variable $\lambda \in \mathbb{R}^{E^+\times d}$. The rest of the proof is divided in two steps: in the first, we derive the updates of the *DDO* algorithm from coordinate gradient descent steps on dual variables, and in the second step we apply Theorem 2 to prove rates of convergence for these coordinate gradient descent steps on function F_A^*.

The partial derivative of F_A^* with respect to coordinate $(ij) \in E^+$ of $\lambda \in \mathbb{R}^{E^+\times d}$ reads:

$$\nabla_{ij}F_A^*(\lambda) = \mu_{ij}(\nabla\phi_i^*((A\lambda)_i) - \nabla\phi_j^*((A\lambda)_j))\,.$$

Consider then the following step of coordinate gradient descent for F_A^* on coordinate $(i_k j_k) \in E^+$ of λ, performed when edge $(i_k j_k)$ is activated at iteration k (corresponding to time t_k):

$$\lambda_{t_k} = \lambda_{t_k-} - \frac{1}{(2\sigma^{-1})\mu_{i_kj_k}^2}\nabla_{i_kj_k}F_A^*(\lambda_{t_k-\tau_{i_kj_k}})\,, \tag{30}$$

corresponding to an instantiation of delayed coordinate gradient descent in the continuized framework, on function F_A^*, for *P.p.p.* of intensities (p_{ij}) for $(ij) \in E$ and p_i^{comp} for $(ii^{\text{comp}}) \in E^{\text{comp}}$. Denoting $v_t = A\lambda_t \in \mathbb{R}^{V^+\times d}$ for $t \geq 0$, we obtain the following formula for updating coordinates i_k, j_k of v when i_kj_k activated, *irrespectively of the choice of* μ_{ij} *in matrix* A:

$$v_{t_k,i_k} = v_{t_k-,i_k} - \frac{\nabla\phi_{i_k}^*(v_{t_k-\tau_{i_kj_k},i_k}) - \nabla\phi_{j_k}^*(v_{t_k-\tau_{i_kj_k},j_k})}{2\sigma^{-1}}\,,$$
$$v_{t_k,j_k} = v_{t_k-,j_k} + \frac{\nabla\phi_{i_k}^*(v_{t_k-\tau_{i_kj_k},i_k}) - \nabla\phi_{j_k}^*(v_{t_k-\tau_{i_kj_k},j_k})}{2\sigma^{-1}}\,. \tag{31}$$

Such updates can be performed locally at nodes i and j after communication between the two nodes (if (ij) is a 'physical edge'), or locally (if (ij) is 'virtual edge'). We refer in the sequel to this scheme as the Coordinate Descent Method. While $\lambda \in \mathbb{R}^{E\times d}$ is a dual variable defined on the edges, $v \in \mathbb{R}^{n\times d}$ is also a dual variable, but defined on the nodes. The *primal surrogate* of v is defined as $x = \nabla F^*(v)$ *i.e.* $x_i = \nabla f_i^*(v_i)$ at node i. It can hence be computed with local updates on v. The decentralized updates of Algorithm 3 (computational updates in Algorithm 2, communication updates in Algorithm 1) are then direct consequences of Equation (31).

The last step of the proof consists in applying Theorem 2 in order to obtain Theorem 3. The function F_A^* we introduced satisfies the assumptions of Theorem 2 with coordinate blocks corresponding to edges E^+: The regularity assumptions are satisfied with smoothness parameter $L_{ij} = 8\mu_{ij}^2\sigma^{-1}$ and local Lipschitz coefficients $M_{(ij),(kl)} = \sqrt{L_{ij}L_{kl}}$ for any $(ij),(kl) \in E^+$, as shown in Lemmas 2 and 3 in the Appendix. F_A^* is moreover σ-strongly convex[6] with σ derived using Lemmas 4 and 5, and the weights associated to matrix A are chosen so that $\mu_{ij}^2 = \frac{\varepsilon_{ij}K_{ij}\sigma}{2\mu_{ij}^2}$.

Finally, the output of the algorithm at node i is the primal surrogate of variable $x_i(t)$ (associated to ϕ_i), which is equal to $\nabla\phi_i(x_i(t)) = \frac{\sigma}{2}x_i(t)$.

■

V. Handling communication and computation capacity limits

A. Communication and computation capacity constraints

A given node or edge in the network may be able to handle only a limited number of communications or computations simultaneously. In Delayed Randomized Gossip and *DDO* algorithms, such constraints could be violated when some P.p.p. generates many points in a short interval. We extend our algorithms and resulting convergence guarantees to take into account these additional constraints.

In the continuized framework, this constraint can be enforced by truncating the *P.p.p.* that handles activations (Definition 2). We formalize communication and capacity constraints in Assumption 2, and show that asynchronous speedup is still achieved in this setting in Theorem 4.

In the previous sections, step size parameters $K_{ij}, K_i^{\text{comp}}$ of the algorithms could be tuned to counterweight the effect of delays for arbitrary intensities p_{ij}. With the introduction of capacity constraints we will see that the local optimizers at every node must also bound the intensities $p_{ij}, p_i^{\text{comp}}$ based on local quantities. The resulting rate of convergence is the same as in Theorems 1 and 3, up to a constant factor of $1/2$.

We formalize communication and computation capacity constraints as follows.

Assumption 2 (Capacity constraints): For some $q_{ij}, q_i^{\text{comm}}, q_i^{\text{comp}} \in \mathbb{N}^* \cup \{\infty\}$, $i \in V$ and $(ij) \in E$,

1) *Computation Capacity:* Node i can compute only q_i^{comp} gradients in an interval of time of length τ_i^{comp};
2) *Communication Capacity, edge-wise limitations:* Only q_{ij} messages can be exchanged simultaneously between adjacent nodes $i \sim j$ in an interval of time of length τ_{ij};
3) *Communication Capacity, node-wise limitations:* Node i can only send q_i^{comm} messages in any interval of time of length $\tau_i^{\text{comm}} = \max_{j\sim i}\tau_{ij}$.

Taking into account these constraints in the analysis boils down to replacing *P.p.p.* processes $(\mathcal{P}_{ij})_{(ij)\in E}$, $(\mathcal{P}_i^{\text{comp}})_{i\in V}$ of intensities (p_{ij}), (p_i^{comp}) in the *DDO* algorithm, by *truncated Poisson point processes* $(\tilde{\mathcal{P}}_{ij}, \tilde{\mathcal{P}}_i^{\text{comp}})$ (see Definition 2).

More precisely, for every edge $(ij) \in E$ (*resp.* node $i \in V$), let $n_{ij}(t)$ be the number of communications occurring along (ij) between times $t - \tau_{ij}$ and t (*resp.* $n_{i,j}^{\text{comm}}$ the number of communications node i is involved in between times t and $t - \tau_{ij}$, n_i^{comp} the number of computations node i is involved with between times t and $t - \tau_i^{\text{comp}}$). Without capacity constraints, these quantities are discrete Poisson random variables (of mean $p_{ij}\tau_{ij}$ for $n_{ij}(t)$, *e.g.*).

B. Convergence guarantees

As in Section IV, we consider communication and computation update rules as in Algorithm 3 (*DDO* algorithm). In the presence of capacity constraints, a communication alongside edge $(ij) \in E$ at a clock ticking $t \in \mathcal{P}_{ij}$ occurs and *does not break the communication capacity constraints* if and only if $n_{ij}(t) < q_{ij}$ (for edge-wise limitations), $n_{i,j}^{\text{comm}}(t) < q_i^{\text{comm}}$ and $n_{j,i}^{\text{comm}}(t) < q_j^{\text{comm}}$ (for node-wise limitations) are satisfied.

Under capacity constraints, we have the following guarantees for our algorithm, defined as in Algorithm 3 (Algorithm 1 for communications and Algorithm 2 for local computations), where communications and computations that violate the capacity constraints are dropped.

Theorem 4: Assume for any $i \in V$ and $(ij) \in E$:

$$\begin{aligned} &cp_i^{\text{comp}}\tau_i^{\text{comp}} \le q_i^{\text{comp}}, \\ &cp_{ij}\tau_{ij} \le q_{ij}, \\ &c\sum_{j\sim i} p_{ij}\tau_i^{\text{comm}} \le q_i^{\text{comm}}, \end{aligned} \tag{32}$$

where $c = 1/(1 - \sqrt{\ln(6)/2})$ is a numerical constant. Then, if the assumptions of Theorem 3 described in Equation 26 additionally hold, for γ verifying

$$\gamma \le \min\left(\frac{\sigma}{8L}\lambda_2\big(\Delta_G(\nu_{ij} = K_{ij})\big), \frac{1}{\tau_{\max}}\right),$$

we have:

$$\frac{\int_0^T e^{\gamma t}\mathbb{E}\left[\left\|\frac{\sigma}{2}x(t) - \bar{x}^\star\right\|^2\right]\mathrm{d}t}{\int_0^T e^{\gamma t}\left\|\frac{\sigma}{2}x(0) - \bar{x}^\star\right\|^2\mathrm{d}t} \le e^{-\frac{\gamma T}{2}}\frac{L}{\sigma}\frac{1 + \frac{\tau_{\max}}{T}}{1 - \gamma\tau_{\max}}.$$

The same guarantees as without the capacity constraints thus hold, up to a constant factor $1/2$ in the rate of convergence. The conditions on the activation intensities (32) suggest that graph sparsity is beneficial: for q_i^{comm} small, $2\sum_{j\sim i} p_{ij}\tau_i^{\text{comm}} \le q_i^{\text{comm}}$ translates into p_{ij} scaling with the inverse of the edge-degree of (ij), so large degrees thus

[6] In fact, it is strongly convex on the orthogonal of Ker$A$, which suffices for us to conclude since the dynamics are restricted to this subspace.

slow down the convergence. The new conditions (32) are easily enforced with the natural choice of intensities p_{ij} (*resp.* $p_i^{\rm comp}$) of order $1/\tau_{ij}$ (*resp.* $\tau_i^{\rm comp}$).

Taking $q_i^{\rm comm}=1$, we recover the behavior of *loss networks* [28], where a node cannot concurrently communicate with different neighbors. Gossip on loss networks was previously studied in [19], to obtain some form of asynchronous speedup. Comparatively, our present algorithms are structurally simpler and their analysis in the continuized framework yields faster convergence speeds.

C. Proof of Theorem 4

Proof: The algorithm under capacity constraints is obtained by applying coordinate gradient descent in the continuized framework to the same dual problem as in Section IV, but with random variables "$\varepsilon_k(t)$" that are not taken constant equal to 1. Here, for $(ij)\in E$ and $t\in\mathcal{P}_{ij}$, we have

$$\varepsilon_{ij}(t)=1_{\left\{n_{ij}(t)<q_{ij}\,,\,n_i^{\rm comm}(t)<q_i^{\rm comm}\,,\,n_j^{\rm comm}(t)<q_j^{\rm comm}\right\}},$$

while for $i\in V$ and $t\in\mathcal{P}_i^{\rm comp}$,

$$\varepsilon_{ii^{\rm comp}}(t)=1_{\left\{n_i^{\rm comp}<q_i^{\rm comp}\right\}}.$$

We apply Theorem 2 as in the proof of Theorem 3, leading to the same stability conditions on the step sizes $K_{ij}, K_i^{\rm comp}$, while the rate of convergence is multiplied by a lower bound ε on all $\mathbb{E}\left[\varepsilon_{ij}(t)\right]$ and $\mathbb{E}\left[\varepsilon_{ii^{\rm comp}}(t)\right]$. Let us finally compute such a lower bound ε.

For $(ij)\in E$, $n_{ij}(t)$ is stochastically dominated by Z_{ij} a Poisson random variable of parameter $p_{ij}\tau_{ij}$, while $n_i^{\rm comm}(t)$ and $n_j^{\rm comm}(t)$ are respectively dominated by Z_i and Z_j, Poisson random variables of parameters $\tau_{ij}\sum_{k\sim i}p_{ki}$ and $\tau_{ij}\sum_{\ell\sim j}p_{\ell j}$, so that:

$$\begin{aligned}\mathbb{E}\left[\varepsilon_{ij}(t)\right]&\geq\mathbb{P}\left(Z_{ij}<q_{ij}\,,\,Z_i<q_i^{\rm comm}\,,\,Z_j<q_j^{\rm comm}\right)\\&\geq 1-\mathbb{P}(Z_{ij}\geq q_{ij})-\mathbb{P}(Z_i\geq q_i^{\rm comm})-P(Z_j\geq q_j^{\rm comm}).\end{aligned}$$

We now prove that $\mathbb{P}(Z_{ij}\geq q_{ij}), \mathbb{P}(Z_i\geq q_i^{\rm comm}), P(Z_j\geq q_j^{\rm comm})$ are all inferior to $1/6$. For $\mathbb{P}(Z_{ij}\geq q_{ij})$, using that for Z a Poisson variable of parameter μ and $x\geq 0$,

$$\mathbb{P}(Z_\mu\geq\mu+x)\leq e^{\frac{-x^2}{\mu+x}},$$

we have $\mathbb{P}(Z_{ij}\geq q_{ij})\leq e^{-\frac{(q_{ij}-p_{ij}\tau_{ij})^2}{q_{ij}}}\leq e^{-2(1-1/c)^2}$ if $q_{ij}\geq 2$, and this quantity is equal to $1/6$, by definition of c. Then, if $q_{ij}=1$, using $\mathbb{P}(Z_\mu\geq 1)=1-e^{-\mu}$, we have that $\mathbb{P}(Z_{ij}\geq q_{ij})\leq p_{ij}\tau_{ij}\leq 1/c\leq 1/6$. We proceed in the same way for $\mathbb{P}(Z_i\geq q_i^{\rm comm}), P(Z_j\geq q_j^{\rm comm})$. Hence, $\mathbb{E}\left[\varepsilon_{ij}(t)\right]\geq 1/2$ under our assumptions on the Poisson intensities. Similarly, we prove that $\mathbb{E}\left[\varepsilon_{ii^{\rm comp}}(t)\right]\geq 1/2$, and this concludes the proof. ∎

VI. Braess's Paradox and Experiments

In this section, we investigate how the local step sizes K_{ij} and Poisson intensities p_{ij} used in Theorems 1, 3 and 4 should be tuned for a fixed choice of communication delays. Consider the line graph with constant delays $\tau_{i,i+1}=\tau$. Add edge $(1,n)$ in order to close the line, with a delay $\tau_{1n}=\tau'$ with arbitrarily large τ'. If the added Poisson intensity p_{1n} satisfies $\tau_{1n}p_{1n}\to\infty$, then according to Theorem 1, we have $K_{12}\to 0$ and $K_{n-1,n}\to 0$. Consequently, since $\gamma=\mathcal{O}(\Delta_G(K)$, we have $\gamma\to 0$: the weighted graph becomes close to disconnected. By adding an edge to the graph, the convergence speed of delayed randomized gossip is degraded.

In order to alleviate the phenomenon, we would need to virtually delete the edge, by setting $p_{1n}=0$. Figure 1 illustrates this phenomenon in the more general setting: one can sparsify the communication graph by solving a regularized optimization problem over the p_{ij} in order to maximize $\lambda_2(\Delta_G(K))$ (K being a function of p), leading to both faster consensus and smaller communication complexity (and thus lower energy footprint).

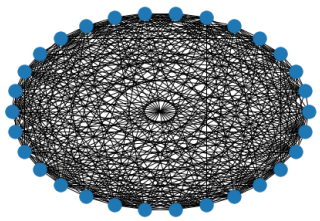
In road-traffic, removing one or more roads in a road network can speed up the overall traffic flow. This phenomenon, called *Braess's paradox* [16], also arises in loss networks [6]. In our problem, this translates to removing an edge (ij) with a non-negligible Poisson intensity p_{ij}. We take G_1 a dense Erdős-Rényi random graph (Figure 1(a)) of parameters $n=30, p=0.75$. Delays τ_{ij} are taken equal to 0.01 with probability 0.9, and to 1 with probability 0.1. Initially, intensities are set as $p_{ij}^{(1)}=1/\tau_{ij}$. Maximizing:

$$\lambda_2\left(\Delta_G\left(\frac{p_{ij}}{1+\sum_{kl\sim ij}p_{kl}(\tau_{ij}+e\tau_{kl})}\right)\right)-\omega\sum_{(ij)\in E}p_{ij}\tau_{ij}$$

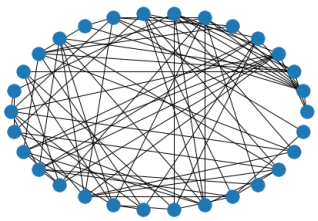
over $(p_{ij})_{ij}$, we obtain intensities $p^{(2)}$ and a graph G_2 (Figure 1(b)), sparser than G_1: we delete edges that have a null intensity (*i.e.* such that $p_{ij}^{(2)}=0$). We then run our delayed gossip algorithm for initialization x_0 a Dirac mass ($x_0(i)=I_{i=i_0}$), on G_1 (blue curves) and G_2, for the choice of K_{ij} as in Theorem 1. The green curve is the synchronous gossip algorithm [15] on G_1, to illustrate the asynchronous speedup, where each iteration takes a time $\tau_{\max}=1$. In Figure 1(c), the error to the consensus is measured as a function of the continuous time, while it is measured in terms of number of updates in Figure 1(d) and in terms of energy (defined as $\sum_{k:T_k<t}\tau_{i_kj_k}$ at time t: the energy consumed by a communication is assumed to be proportional to the time the communication took) in Figure 1(e).

As expected, in terms of number of updates in the whole graph and energy spent, the sparser graph is more effective: slow and costly edges were deleted. Perhaps more surprising, but supported by our theory (Theorem 3) and the resulting Braess's paradox, this also holds in Figure 1(c): even though in the same amount of time, less updates are made in the sparser graph G_2 than in G_1, delayed randomized gossip is still faster on G_2 than G_1. Making less updates and deleting some communications make all other communications more efficient.

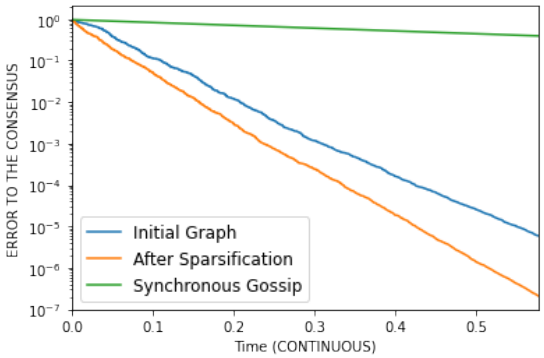
We believe that this phenomenon could be exploited for efficient design of large scale networks, beyond the maximization the spectral gap regardless of physical constraints as in [68] for instance.



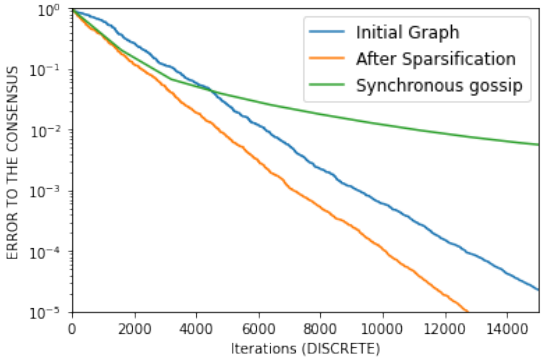
(a) Graph G_1 (E-R)



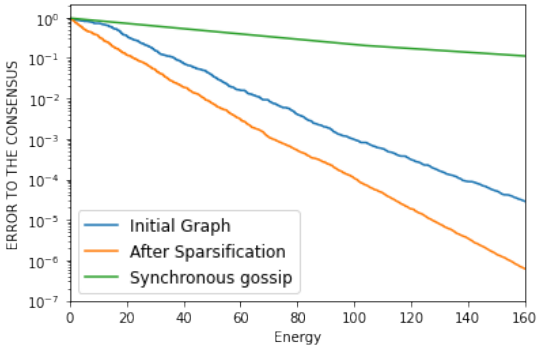
(b) Graph G_2 (after sparsification)



(c) Continuous time



(d) Number of updates in the whole graph



(e) Energy spent

Fig. 1. **Experiments and Braess's paradox.**

CONCLUSION

We introduced a novel analysis framework for the study of algorithms in the presence of delays, establishing that an *asynchronous speedup* can be achieved in decentralized optimization. Our results hold for explicit choices of algorithm parameters based on local network characteristics. They derive from the continuous-time analysis and assumptions handled in our continuized framework. The explicit conditions and convergence rates we obtain allow us to further discuss counter-intuitive effects akin to the Braess paradox, such as the possibility to speed up convergence by suppressing communication links. Although the algorithm requires dual updates, a fully primal algorithm could be obtained by using Bregman gradients [25] or a primal-dual formulation [34].

APPENDIX

For σ-strongly convex and L-smooth functions $f_1, \dots, f_n$ on $\mathbb{R}^d$ and for $A \in \mathbb{R}^{V \times E}$ such that $Ae_{ij} = \mu_{ij}(e_i - e_j)$ for $(ij) \in E$, define $F_A^* : \mathbb{R}^{E \times d} \to \mathbb{R}$ as:

$$F_A^*(\lambda) = \frac{1}{n} \sum_{i \in V} f_i((A\lambda_i)), \quad \lambda \in \mathbb{R}^{E \times d}.$$

Lemma 2: For any $(ij) \in E$, F_A^* is $L_{ij} := 4\mu_{ij}^2 \sigma^{-1}$-smooth on E_{ij} the subspace of coordinates $(ij) \in E$.

Proof: Let $h_{ij} \in \mathbb{R}^d$ and $\lambda \in \mathbb{R}^{E \times d}$. Using the σ^{-1}-smoothness of f_i^* and f_j^*:

$$\begin{aligned} F_A^*(\lambda + e_{ij}h_{ij}^\top) - F_A^*(\lambda) &= f_i^*((A(\lambda + e_{ij}h_{ij})^\top)_i) - f_i^*((A\lambda)_i) \\ &+ f_j^*((A(\lambda + e_{ij}h_{ij}^\top))_j) - f_j^*((A\lambda)_j) \\ &\leq \langle \nabla_{ij} F_A^*(\lambda), e_{ij}h_{ij}^\top \rangle \\ &+ \frac{\sigma^{-1}}{2} \left\| (Ae_{ij}h_{ij}^\top)_i \right\|^2 + \frac{\sigma^{-1}}{2} \left\| (Ae_{ij}h_{ij}^\top)_j \right\|^2, \end{aligned}$$

concluding the proof, as $\left\| (Ae_{ij}h_{ij}^\top)_i \right\|^2 = 2\mu_{ij}^2 \|h_{ij}\|^2$. ∎

Lemma 3: For any $(ij) \in E$, any $\lambda, \lambda' \in \mathbb{R}^{E^+ \times d}$:

$$\|\nabla_{ij} F_A^*(\lambda) - \nabla_{ij} F_A^*(\lambda')\| \leq \sum_{(kl) \sim (ij)} M_{(ij),(kl)} \|\lambda_{kl} - \lambda'_{kl}\|, \tag{33}$$

where $M_{(ij),(kl)} = \sqrt{L_{ij} L_{kl}}$ and $L_{ij} = 4\mu_{ij}^2 \sigma^{-1}, L_{kl} = 4\mu_{kl}^2 \sigma^{-1}$.

Proof: Since $\nabla_{ij} F_A^*(\lambda) = (Ae_{ij})^\top ((\nabla g_i^*((A\lambda)_i) -$

$\nabla g_j^*((A\lambda)_j))$, we have:

$$\begin{aligned}
&\|\nabla_{ij}F_A^*(\lambda) - \nabla_{ij}F_A^*(\lambda')\| \\
&= \big\|(Ae_{ij})^\top((\nabla f_i^*((A\lambda)_i) - \nabla f_i^*((A\lambda')_i) \\
&\quad - \nabla f_j^*((A\lambda)_j) + \nabla f_j^*((A\lambda')_j))\big\| \\
&\leq \|Ae_{ij}\|\big(\|\nabla f_i^*((A\lambda)_i) - \nabla f_i^*((A\lambda')_i)\| \\
&\quad + \big\|\nabla f_j^*((A\lambda)_j) - \nabla f_j^*((A\lambda')_j)\big\|\big) \\
&\leq \sqrt{2}|\mu_{ij}|\big(\sigma_i^{-1}\|(A\lambda)_i - (A\lambda')_i\| + \sigma_j^{-1}\|(A\lambda)_j - (A\lambda')_j\|\big) \\
&= \sqrt{2}|\mu_{ij}|\left(\sigma_i^{-1}\left\|\sum_{k\sim i}\mu_{ik}(\lambda_{ik} - \lambda'_{ik})\right\|\right. \\
&\quad \left.+\sigma_j^{-1}\left\|\sum_{l\sim j}\mu_{jl}(\lambda_{jl} - \lambda'_{jl})\right\|\right) \\
&\leq \sqrt{2}|\mu_{ij}|\left(\sqrt{\sigma_i^{-1}+\sigma_j^{-1}}\sqrt{\sigma_i^{-1}+\sigma_k^{-1}}\sum_{k\sim i}|\mu_{ik}|\|\lambda_{ik} - \lambda'_{ik}\|\right) \\
&\quad + \sqrt{\sigma_i^{-1}+\sigma_j^{-1}}\sqrt{\sigma_l^{-1}+\sigma_j^{-1}}\sum_{l\sim j}|\mu_{jl}|\|\lambda_{jl} - \lambda'_{jl}\|\big)\Big) \\
&\leq \sum_{(kl)\sim(ij)}\sqrt{L_{ij}L_{kl}}\|\lambda_{kl} - \lambda'_{kl}\|,
\end{aligned}$$

where L_{ij}, L_{kl} as in Lemma 2. ∎

Lemma 4 (Strong convexity): The strong convexity parameter σ_A of F_A^* on the orthogonal of $\ker(A)$ is lower bounded by $L^{-1}\lambda_2(\Delta_G(\mu_{ij}^2))$, where we recall that $\lambda_2(\Delta_G(\mu_{ij}^2))$ is the graph Laplacian with weights μ_{ij}^2.

Proof: Let $\lambda, \lambda' \in \mathbb{R}^{E\times d}$. For $i \in V$, by L^{-1}-strong convexity of f_i^*, $f_i^*((A\lambda)_i) - f_i^*((A\lambda')_i) \geq \langle\nabla f_i^*((A\lambda')_i), (A(\lambda-\lambda'))_i\rangle + \frac{1}{2L}\|(A(\lambda-\lambda'))_i\|^2$. Summing over all $i \in V$ and using $\nabla F_A^*(\lambda') = A^\top(\nabla_i f_i^*((A\lambda')_i))_{i\in V}$ leads to:

$$\begin{aligned}
F_A^*(\lambda) - F_A^*(\lambda') &\geq \langle\nabla F_A^*(\lambda'), \lambda-\lambda'\rangle + \frac{1}{2L}\|A(\lambda'-\lambda\|^2 \\
&\geq \langle\nabla F_A^*(\lambda'), \lambda-\lambda'\rangle + \frac{\lambda_{\min}^+(A^TA)}{4}\|\lambda-\lambda'\|^{*2}.
\end{aligned}$$

where $\|.\|^*$ is the euclidean norm on the orthogonal of $Ker(A)$. Finally, notice that $AA^\top = \Delta_G(\mu_{ij}^2)$ and has same eigenvalues as $A^\top A$. ∎

Let $G = (V, E)$ be the "physical" graph, augmented as $G^+ = (V^+, E^+)$, where $V^+ = V \cup \{i^{\text{comp}}, i \in V\}$ and $E^+ = E \cup \{(ii^{\text{comp}}), i \in V\}$ as in Section IV.

Lemma 5: For $\nu^+ = (\nu_{ij})_{(ij)\in E^+}$ non negative weights, the smallest positive eigenvalue of the Laplacian of the augmented graph G^+ with weights ν^+ satisfies:

$$\lambda_2\big(\Delta_{G^+}(\nu^+)\big) \geq \frac{1}{4}\min\left(\lambda_2\big(\Delta_G(\nu)\big), \min_{i\in V}\nu_{ii^{\text{comp}}}\right),$$

where $\lambda_2\big(\Delta_G(\nu)\big)$ is the smallest eigenvalue of the original graph, with weights $\nu = (\nu_{ij})_{(ij)\in E}$.

Proof: Let $m = \min_{i\in V}\nu_{ii^{\text{comp}}}$ and $\lambda = \lambda_2\big(\Delta_G(\nu)\big)$. For any $X = (x, y) \in \mathbb{R}^{V^+}$, we have that $X^\top\Delta_{G^+}(\nu^+)X = \sum_{(ij)\in E^+}\nu_{ij}(X_i - X_j)^2 = x^\top\Delta_G(\nu)x + \sum_{i\in V}\nu_{ii^{\text{comp}}}(x_i - y_i)^2 \geq \lambda\|x - \bar{X}\|^2 + m\|x - y\|^2$. Then, for $c > 0$ sufficiently small such that for any $z, z' \in \mathbb{R}$, $\lambda z^2 + m(z - z')^2 \geq cz^2 + cz'^2$, we have $X^\top\Delta_{G^+}(\nu^+)X \geq c\|X - \bar{X}\|^2$ and so $\lambda_2\big(\Delta_{G^+}(\nu^+)\big) \geq c$. Let us now compute such a value c, to conclude this proof. For $z, z' \in \mathbb{R}$, $\lambda z^2 + m(z - z')^2 - cz^2 - cz'^2 = \left(\sqrt{\lambda+m-c}z - \frac{m}{\sqrt{\lambda+m-c}}z'\right)^2 + \left(m - c - \frac{m^2}{\lambda+m-c}\right)z'^2$, and this quantity is non-negative as long as $c \leq \min(\lambda, m)/4$. ∎

REFERENCES

[1] A. Agarwal and J. C. Duchi. Distributed delayed stochastic optimization. *NeurIPS*, 24, 2011.

[2] D. Alistarh, D. Grubic, J. Z. Li, R. Tomioka, and M. Vojnovic. Qsgd: Communication-efficient sgd via gradient quantization and encoding. In *NeurIPS*, 2017.

[3] M. Assran, A. Aytekin, H. R. Feyzmahdavian, M. Johansson, and M. G. Rabbat. Advances in asynchronous parallel and distributed optimization. *Proceedings of the IEEE*, 2020.

[4] M. S. Assran and M. G. Rabbat. Asynchronous gradient push. *IEEE Transactions on Automatic Control*, 2021.

[5] G. M. Baudet. Asynchronous iterative methods for multiprocessors. *Journal of the ACM (JACM)*, 25(2):226–244, 1978.

[6] N. G. Bean, F. P. Kelly, and P. G. Taylor. Braess's paradox in a loss network. *Journal of Applied Probability*, 34(1):155–159, 1997.

[7] T. Ben-Nun and T. Hoefler. Demystifying parallel and distributed deep learning: An in-depth concurrency analysis. *ACM Computing Surveys (CSUR)*, 52(4):1–43, 2019.

[8] M. Bornstein, T. Rabbani, E. Z. Wang, A. Bedi, and F. Huang. SWIFT: Rapid decentralized federated learning via wait-free model communication. In *ICLR*, 2023.

[9] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah. Randomized gossip algorithms. *IEEE transactions on information theory*, 2006.

[10] S. Bubeck. Convex optimization: Algorithms and complexity. *Found. Trends Mach. Learn.*, 8(3–4):231–357, Nov. 2015.

[11] J. Chen, X. Pan, R. Monga, S. Bengio, and R. Jozefowicz. Revisiting distributed synchronous SGD. *arXiv preprint arXiv:1604.00981*, 2016.

[12] A. Chowdhery and coauthors. Palm: Scaling language modeling with pathways, 2022.

[13] E. Cyffers, M. Even, A. Bellet, and L. Massoulié. Muffliato: Peer-to-peer privacy amplification for decentralized optimization and averaging. In *NeurIPS*, 2022.

[14] M. H. A. Davis. Piecewise-deterministic markov processes: A general class of non-diffusion stochastic models. *Journal of the Royal Statistical Society. Series B (Methodological)*, 46(3):353–388, 1984.

[15] A. G. Dimakis, S. Kar, J. M. F. Moura, M. G. Rabbat, and A. Scaglione. Gossip algorithms for distributed signal processing. *Proceedings of the IEEE*, 98(11):1847–1864, 2010.

[16] D. Easley and J. Kleinberg. *Networks, Crowds, and Markets: Reasoning about a Highly Connected World*. Cambridge University Press, 2010.

[17] M. Even. Stochastic gradient descent under markovian sampling schemes. In *ICML*, ICML'23, 2023.

[18] M. Even, R. Berthier, F. Bach, N. Flammarion, H. Hendrikx, P. Gaillard, L. Massoulié, and A. Taylor. Continuized accelerations of deterministic and stochastic gradient descents, and of gossip algorithms. In *NeurIPS*, 2021.

[19] M. Even, H. Hendrikx, and L. Massoulié. Asynchrony and acceleration in gossip algorithms. *arXiv preprint arXiv:2011.02379*, 2020.

[20] H. Feyzmahdavian and M. Johansson. Asynchronous iterations in optimization: New sequence results and sharper algorithmic guarantees. In *JMLR, 2023*, 2021.

[21] Y. Geng, S. Liu, Z. Yin, A. Naik, B. Prabhakar, M. Rosenblum, and A. Vahdat. Exploiting a natural network effect for scalable, fine-grained clock synchronization. In *USENIX Symposium on Networked Systems Design and Implementation*, 2018.

[22] K. Gu and Y. Liu. Lyapunov–krasovskii functional for uniform stability of coupled differential-functional equations. *Automatica*, 2009.

[23] H. Hendrikx. A principled framework for the design and analysis of token algorithms. In *AISTATS*, 2023.

[24] H. Hendrikx, F. Bach, and L. Massoulié. An accelerated decentralized stochastic proximal algorithm for finite sums. In *NeurIPS*, 2019.

[25] H. Hendrikx, F. Bach, and L. Massoulié. Dual-free stochastic decentralized optimization with variance reduction. In *NeurIPS*, 2020.

[26] S. Horváth, S. Laskaridis, M. Almeida, I. Leontiadis, S. I. Venieris, and N. D. Lane. FjORD: Fair and accurate federated learning under heterogeneous targets with ordered dropout. *NeurIPS*, 2021.

[27] P. Kairouz and coauthors. Advances and open problems in federated learning, 2019.
[28] F. P. Kelly. Loss networks. *The Annals of Applied Probability*, 1991.
[29] A. Klenke. *The Poisson Point Process*. Springer London, 2014.
[30] A. Koloskova, N. Loizou, S. Boreiri, M. Jaggi, and S. Stich. A unified theory of decentralized sgd with changing topology and local updates. In *International Conference on Machine Learning*, 2020.
[31] A. Koloskova, S. Stich, and M. Jaggi. Decentralized stochastic optimization and gossip algorithms with compressed communication. In *36th International Conference on Machine Learning*, 2019.
[32] A. Koloskova, S. U. Stich, and M. Jaggi. Sharper convergence guarantees for asynchronous SGD for distributed and federated learning. In *NeurIPS*, 2022.
[33] D. Kovalev, E. Gasanov, A. Gasnikov, and P. Richtárik. Lower bounds and optimal algorithms for smooth and strongly convex decentralized optimization over time-varying networks. In *NeurIPS*, 2021.
[34] D. Kovalev, A. Salim, and P. Richtárik. Optimal and practical algorithms for smooth and strongly convex decentralized optimization. *NeurIPS*, 33, 2020.
[35] R. Leblond, F. Pedregosa, and S. Lacoste-Julien. Improved asynchronous parallel optimization analysis for stochastic incremental methods. *Journal of Machine Learning Research*, 19(81):1–68, 2018.
[36] J. Li, G. Chen, Z. Y. Dong, and Z. Wu. Distributed mirror descent method for multi-agent optimization with delay. *Neurocomputing*, 2016.
[37] X. Lian, Y. Huang, Y. Li, and J. Liu. Asynchronous parallel stochastic gradient for nonconvex optimization. *NeurIPS*, 28, 2015.
[38] X. Lian, C. Zhang, H. Zhang, C.-J. Hsieh, W. Zhang, and J. Liu. Can decentralized algorithms outperform centralized algorithms? a case study for decentralized parallel stochastic gradient descent. In *Neural Information Processing Systems*, 2017.
[39] X. Lian, W. Zhang, C. Zhang, and J. Liu. Asynchronous decentralized parallel stochastic gradient descent. In *ICML*, 2018.
[40] Q. Liu, B. Yang, Z. Wang, D. Zhu, X. Wang, K. Ma, and X. Guan. Asynchronous decentralized federated learning for collaborative fault diagnosis of pv stations. *IEEE Transactions on Network Science and Engineering*, 9(3):1680–1696, 2022.
[41] Q. Luo, J. He, Y. Zhuo, and X. Qian. Prague: High-performance heterogeneity-aware asynchronous decentralized training. In *Proceedings of the Twenty-Fifth International Conference on Architectural Support for Programming Languages and Operating Systems*, 2020.
[42] H. Mania, X. Pan, D. Papailiopoulos, B. Recht, K. Ramchandran, and M. I. Jordan. Perturbed iterate analysis for asynchronous stochastic optimization. *SIAM Journal on Optimization*, 27(4):2202–2229, 2017.
[43] L. Massoulié. Stability of distributed congestion control with heterogeneous feedback delays. *IEEE Transactions on Automatic Control*, 2002.
[44] K. Mishchenko, F. Bach, M. Even, and B. Woodworth. Asynchronous sgd beats minibatch sgd under arbitrary delays, 2022.
[45] K. Mishchenko, F. Iutzeler, J. Malick, and M.-R. Amini. A delay-tolerant proximal-gradient algorithm for distributed learning. In *International Conference on Machine Learning*, pages 3584–3592, 2018.
[46] G. Nadiradze, A. Sabour, P. Davies, S. Li, and D. Alistarh. Asynchronous decentralized SGD with quantized and local updates. In *NeurIPS*, 2021.
[47] A. Nedic and A. Ozdaglar. Distributed subgradient methods for multi-agent optimization. *IEEE Transactions on Automatic Control*, 54(1):48–61, 2009.
[48] S.-I. Niculescu. *Delay effects on stability: a robust control approach*, volume 269. Springer Science & Business Media, 2001.
[49] B. Recht, C. Re, S. Wright, and F. Niu. Hogwild!: A lock-free approach to parallelizing stochastic gradient descent. *NeurIPS*, 24, 2011.
[50] M. Ryabinin, E. Gorbunov, V. Plokhotnyuk, and G. Pekhimenko. Moshpit SGD: Communication-efficient decentralized training on heterogeneous unreliable devices. *NeurIPS*, 2021.
[51] K. Scaman, F. Bach, S. Bubeck, Y. T. Lee, and L. Massoulié. Optimal algorithms for smooth and strongly convex distributed optimization in networks. In *International Conference on Machine Learning*, 2017.
[52] W. Shi, Q. Ling, G. Wu, and W. Yin. EXTRA: An exact first-order algorithm for decentralized consensus optimization. *SIAM Journal on Optimization*, 25(2):944–966, 2015.
[53] B. Sirb and X. Ye. Decentralized consensus algorithm with delayed and stochastic gradients. *SIAM Journal on Optimization*, 2018.
[54] S. Sra, A. W. Yu, M. Li, and A. J. Smola. Adadelay: Delay adaptive distributed stochastic optimization. In *Artificial Intelligence and Statistics*, pages 957–965. PMLR, 2016.
[55] S. U. Stich. Local SGD converges fast and communicates little. In *International Conference on Learning Representations*, 2019.
[56] S. U. Stich and S. P. Karimireddy. The error-feedback framework: Better rates for SGD with delayed gradients and compressed updates. *Journal of Machine Learning Research*, 21:1–36, 2020.
[57] Y. Tian, Y. Sun, and G. Scutari. Asy-sonata: Achieving linear convergence in distributed asynchronous multiagent optimization. In *2018 56th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 543–551, 2018.
[58] Y. Tian, Y. Sun, and G. Scutari. Achieving linear convergence in distributed asynchronous multi-agent optimization. *IEEE Transactions on Automatic Control*, PP:1–1, 03 2020.
[59] H. Touvron and coauthors. Llama: Open and efficient foundation language models, 2023.
[60] J. Tsitsiklis, D. Bertsekas, and M. Athans. Distributed asynchronous deterministic and stochastic gradient optimization algorithms. *IEEE transactions on automatic control*, 31(9):803–812, 1986.
[61] H. Wang, X. Liao, T. Huang, and C. Li. Cooperative distributed optimization in multiagent networks with delays. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 45(2):363–369, 2015.
[62] J. Wang, A. K. Sahu, Z. Yang, G. Joshi, and S. Kar. Matcha: Speeding up decentralized sgd via matching decomposition sampling. In *2019 Sixth Indian Control Conference (ICC)*, pages 299–300, 2019.
[63] B. Woodworth, K. K. Patel, S. Stich, Z. Dai, B. Bullins, B. Mcmahan, O. Shamir, and N. Srebro. Is local SGD better than minibatch SGD? In *International Conference on Machine Learning*, 2020.
[64] T. Wu, K. Yuan, Q. Ling, W. Yin, and A. H. Sayed. Decentralized consensus optimization with asynchrony and delays. *IEEE Transactions on Signal and Information Processing over Networks*, 2018.
[65] T. Wu, K. Yuan, Q. Ling, W. Yin, and A. H. Sayed. Decentralized consensus optimization with asynchrony and delays. *IEEE Transactions on Signal and Information Processing over Networks*, 4(2):293–307, 2018.
[66] X. Wu, C. Liu, S. Magnússon, and M. Johansson. Delay-agnostic asynchronous coordinate update algorithm. In *Proceedings of the 40th International Conference on Machine Learning*, 2023.
[67] C. Xi, V. S. Mai, R. Xin, E. H. Abed, and U. A. Khan. Linear convergence in optimization over directed graphs with row-stochastic matrices. *IEEE Transactions on Automatic Control*, 2018.
[68] B. Ying, K. Yuan, Y. Chen, H. Hu, P. Pan, and W. Yin. Exponential graph is provably efficient for decentralized deep training. In *NeurIPS*, 2021.
[69] J. Zhang and K. You. Fully asynchronous distributed optimization with linear convergence in directed networks, 2021.
[70] S. Zheng, Q. Meng, T. Wang, W. Chen, N. Yu, Z.-M. Ma, and T.-Y. Liu. Asynchronous stochastic gradient descent with delay compensation. In *International Conference on Machine Learning*, pages 4120–4129, 2017.

Mathieu Even is a PhD student at Inria Paris in the Dyogene team, under the supervision of Laurent Massoulié. His work focuses on optimization and statistics for machine learning, with a focus on distributed and federated systems.

Hadrien Hendrikx is a research scientist at Inria Grenoble. Before that, he was a post-doc in the MLO team from EPFL, working with Martin Jaggi. He completed a Ph.D. at Inria Paris, under the supervision of Francis Bach and Laurent Massoulié. His research focuses on optimization for machine learning, and on decentralized methods in particular.

Laurent Massoulié is research director at Inria, head of the Microsoft Research – Inria Joint Centre, and professor at the Applied Maths Centre of Ecole Polytechnique. His research interests are in machine learning, probabilistic modelling and algorithms for networks. He has held research scientist positions at: France Telecom, Microsoft Research, Thomson-Technicolor, where he headed the Paris Research Lab. He obtained best paper awards at IEEE INFOCOM 1999, ACM SIGMETRICS 2005, ACM CoNEXT 2007, NeurIPS 2018, NeurIPS 2021, was elected "Technicolor Fellow" in 2011, received the "Grand Prix Scientifique" of the Del Duca Foundation delivered by the French Academy of Science in 2017, and is a Fellow of the "Prairie" Institute.