Decentralized optimization over slowly time-varying graphs: algorithms and lower bounds

Decentralized Optimization Over Slowly Time-Varying Graphs: Algorithms and Lower Bounds

Dmitry Metelev[1], Aleksandr Beznosikov[1,2,3], Alexander Rogozin[1,2,4], Alexander Gasnikov[1,2,5], Anton Proskurnikov[6]

[1]Moscow Institute of Physics and Technology, Moscow, Russia.
[2]Skolkovo Institute of Science and Technology, Moscow, Russia.
[3]Mohamed bin Zayed University of Artificial Intelligence, Abu Dhabi, United Arab Emirates.
[4]HSE University, Moscow, Russia.
[5]Institute for Information Transmission Problems, Moscow, Russia.
[6]Politecnico di Torino, Turin, Italy.

Contributing authors: metelev.ds@phystech.edu; beznosikov.an@phystech.edu; aleksandr.rogozin@phystech.edu; gasnikov@yandex.ru; anton.p.1982@ieee.org;

Abstract

We consider a decentralized convex unconstrained optimization problem, where the cost function can be decomposed into a sum of strongly convex and smooth functions, associated with individual agents, interacting over a static or time-varying network. Our main concern is the convergence rate of first-order optimization algorithms as a function of the network's graph, more specifically, of the condition numbers of gossip matrices. We are interested in the case when the network is time-varying but the rate of changes is restricted. We study two cases: randomly changing network satisfying Markov property and a network changing in a deterministic manner. For the random case, we propose a decentralized optimization algorithm with accelerated consensus. For the deterministic scenario, we show that if the graph is changing in a worst-case way, accelerated consensus is not possible even if only two edges are changed at each iteration. The fact that such a low rate of network changes is sufficient to make accelerated consensus impossible is novel and improves the previous results in the literature.

Keywords: convex optimization, decentralized optimization, time-varying network, consensus, convergence rate

1 Introduction

The purpose of this paper is to study the problem of distributed unconstrained optimization problem, where the cost function is constructed as the average

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

of n strongly convex functions $\{f_i\}_{i=1}^n$, associated to n autonomous agents.

Following the standard framework of distributed convex optimization [1–3], we assume that agents communicate synchronously and can transmit real numbers of vectors to their teammates; the effects of communication delays and packet losses are ignored. At each iteration of the algorithm agent i updates its state by applying some first-order[1] algorithm aiming at minimizing of the function f; this algorithm can use the internal variables of agent i and information obtained from some of the other agents through a communication network. This communication network is represented by a graph whose vertices (nodes) are in one-to-one correspondence with the agents and whose edges represent communication channels available at the current iteration. In various settings, the communication network may remain static or change in a certain way, in our case in particular, we study the problem (1), imposing specific constraints on the change of the communication network.

Decentralized optimization has emerged as an essential tool for managing sum-type problems of type (1). Decentralized algorithms have found significant applications in areas where centralized coordination is limited due to data volume or privacy restrictions. Agents in these decentralized systems maintain local optimization objectives and participate in a network whose structure may evolve over time. Such problems find application in wireless sensor networks [4], resource allocation problems [5], distributed averaging [6, 7], distributed sensing [8], vehicle coordination and control [9], formation control [10–12], distributed data analysis [13–15], power system control [16, 17].

Related Work. In the literature, the complexity of decentralized optimization algorithms is typically represented by condition number of the network χ and condition number of objective functions $\kappa = L/\mu$. Survey [18] gives an introduction of decentralized optimization. For the case of a static network, this problem is relatively well-studied. In the work [3] a communication complexity lower bound of $\Omega\left(\sqrt{\chi\kappa}\log\left(\frac{1}{\varepsilon}\right)\right)$ was established and the optimal algorithm called MSDA was proposed, assuming access to the dual oracle. In the case of the primal oracle, the optimal algorithm OPAPC [19] was suggested, reaching lower bounds from [3].

[1] The exact definition of the first-order optimization algorithm will be given below.

Discussing methods for addressing non-static graphs in decentralized optimization, it is worth to highlight "gradient tracking" [20, 21]. Its main concept is for each node to store an approximation of the average gradient over the network. At the same time, ADOM [22, 23], which uses saddle-point reformulation and error-feedback. There is also the "inexact oracle" approach [24, 25]. Here, a standard non-decentralized algorithm is modified into a decentralized one. After each gradient iteration, a consensus process is started. This makes convergence analysis straightforward, but introduces an additional logarithmic factor. ADMM-based approaches should also be noted. They are usually used in the static case, but can also be presented in more dynamic setups, as demonstrated, for example, in papers [26, 27].

In the non-static case, when the network can arbitrarily change over time, a lower bound of $\Omega\left(\chi\sqrt{\kappa}\log\left(\frac{1}{\varepsilon}\right)\right)$ was established in [28]. Corresponding optimal algorithms were also derived: ADOM+ [28], Acc-GT [20], considering the primal oracle, and ADOM [22], considering the dual oracle.

Considering non-static slowly-changing case, the paper [29] adresses the monotonic mode of network change and proposes a consensus algorithm. This algorithm produces a near-optimal optimization method in the given setting, achieving a convergence rate comparable to the optimal algorithm in the static setting, although with an additional logarithmic factor dependent on χ. Moreover, in the same paper, regarding the lower bounds in the case of a slowly time-varying network (when constraints are imposed on its rate of change), there were obtained three different lower bounds, each depending on the degree of constraint on the rate of edge changes per temporal iteration. Specifically, they correspond to the following regimes:

- The mode with $\mathcal{O}(n^\alpha)$ $(\alpha > 0)$ edge changes yields a lower bound of $\Omega(\chi\sqrt{\kappa}\log\frac{1}{\varepsilon})$.
- The mode with $\mathcal{O}(\log(n))$ edge changes corresponds to $\Omega(\frac{\chi}{\log\chi}\sqrt{\kappa}\log\frac{1}{\varepsilon})$.
- The mode with $c = const$ edge changes corresponds to $\Omega(\chi^{d(c)}\sqrt{\kappa}\log\frac{1}{\varepsilon})$, where $c \geq 12$ and $\frac{1}{2} < d(c) < 1$.

Our contribution. The contribution of this paper is twofold.

Firstly, we study consensus algorithms over time-varying graphs with restricted changes that change randomly and satisfy Markov condition. We treat consensus problem as a stochastic optimization problem and propose an accelerated consensus method that is based on accelerated stochastic gradient method. After that, we propose an accelerated method for decentralized optimization under our assumptions.

Secondly, we show that accelerated consensus is not attainable for decentralized optimization over time-varying networks with worst-case changes. Our lower bounds are based on a counterexample graph in which no more than two edges are altered at each iteration. Previously lower bounds were provided in [29], and our results make a significant improvement over that.

2 Preliminaries

2.1 Smoothness and strong convexity

In this paper, by H we denote any Hilbert space over $\mathbb{R}$, such as $\mathbb{R}^n$ or ℓ_2.

Definition 1 (μ-Strongly Convex Function)**.** *A function $h : H \to \mathbb{R}$ is called μ-strongly convex if for any $x, y \in H$, the following inequality holds*

$$h(y) \geq h(x) + \langle \nabla h(x), y - x \rangle + \frac{\mu}{2}\|y - x\|^2.$$

Definition 2 (L-Smooth Function)**.** *A function $h : H \to \mathbb{R}$ is called L-smooth if for any $x, y \in H$, it satisfies*

$$\|\nabla h(y) - \nabla h(x)\|_* \leq L\|y - x\|.$$

We will refer to the functions f_i at the nodes of the network as *local functions*. The function f in (1) will be referred to as *the global function*.

2.2 Laplacians

Further on in the paper, we consider only loop-less undirected graphs.
Definition 3 (Weighted Graph)**.** *Let $\mathcal{G}_A = (\mathcal{V}, \mathcal{E}, A)$ denote an undirected weighted graph with nodes $\mathcal{V}$, edges $\mathcal{E}$ and edge weights represented by adjacency matrix $A = [a_{ij}]_{i,j=1}^n$. Weight a_{ij} is positive if $(i, j) \in \mathcal{E}$ and zero otherwise.*
Definition 4 (Laplacian of a Weighted Graph)**.** *Let $\mathcal{G}_A = (\mathcal{V}, \mathcal{E}, A)$ be a weighted graph. The Laplacian of $\mathcal{G}_A$ is defined as*

$$[L(\mathcal{G})]_{ij} = \begin{cases} \sum_{(k,i)\in\mathcal{E}} a_{ik}, & \text{if } i = j, \\ -a_{ij}, & \text{if } (i, j) \in \mathcal{E}, \\ 0, & \text{else.} \end{cases}$$

The unweighted or standard Laplacian of unweighted graph $\mathcal{G} = \mathcal{G}(\mathcal{V}, \mathcal{E})$ is simply the weighted Laplacian of the weighted graph $\mathcal{G}_A = (\mathcal{V}, \mathcal{E}, A)$ with all weights set to 1, i.e. $a_{ij} = 1$ if $(i, j) \in \mathcal{E}$ and $a_{ij} = 0$ if $(i, j) \notin \mathcal{E}$.
Example: Consider a graph $\mathcal{G}_A = (\mathcal{V}, \mathcal{E}, A)$ with 3 vertices $\mathcal{V} = (1, 2, 3)$ and edges $\mathcal{E} = \{(1, 2), (2, 3)\}$ with weights $a_{12} = 2$ and $a_{23} = 1$. The weighted Laplacian matrix for this graph is given by:

$$L(\mathcal{G}_A) = \begin{bmatrix} 2 & -2 & 0 \\ -2 & 3 & -1 \\ 0 & -1 & 1 \end{bmatrix}.$$

It is worth mentioning the well-known
Lemma 1. *For a weighted graph $\mathcal{G}_A$ with positive weights $a_{ij} > 0 \ \forall (i, j) \in \mathcal{E}$, the Laplacian $L(\mathcal{G}_A)$ is a positive semidefinite symmetric matrix whose kernel contains the column of ones. Furthermore,* $\ker L(\mathcal{G}_A) = span\{1\}$ *if and only if the graph $\mathcal{G}_A$ is connected.*

For the proof, see [30, Lemma 2].

Moreover, we introduce a mini-Laplacian

Definition 5. *Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a graph. The mini-Laplacian ℓ_{ij} is a $|\mathcal{V}| \times |\mathcal{V}|$ matrix defined as follows:*

$$[\ell_{ij}]_{kl} = \begin{cases} 1, & \text{if } (k,l) = (i,i) \text{ or } (k,l) = (j,j), \\ -1, & \text{if } (k,l) = (i,j) \text{ or } (k,l) = (j,i), \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

2.3 Gossip matrices

Definition 6 (Gossip Matrix)**.** *Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a graph with n nodes. We call matrix $W(\mathcal{G}) \in \mathbb{R}^{n \times n}$ a gossip matrix if*

1. $[W(\mathcal{G})]_{i,j} = 0$, *if* $i \neq j$ *and* $(i,j) \notin \mathcal{E}$.
2. $\ker W(\mathcal{G}) = \{(x_1, \ldots, x_n) \in \mathbb{R}^n : x_1 = \cdots = x_n\}$.

Note that a Laplacian of a weighted graph satisfies Definition 6.

3 Algorithms complexity for graphs with Markovian changes

In this section, we show that one can organize an accelerated consensus procedure for communication networks changing slowly and according to Markovian law. Using this procedure, we can achieve an improvement in the number of communications in decentralized optimization algorithms.

3.1 Consensus for networks with Markovian changes

Since communication networks change over time, the gossip matrices corresponding to these networks also time-varying. We define G as the set of all possible graphs that can occur through time and W_G as as the set of gossip matrices for G. For simplicity we can consider that each graph $\mathcal{G} \in G$ corresponds to exactly one matrix from $W \in W_G$. But one can note that for the graph $\mathcal{G}$ it is possible to define different gossip matrices at different moments of time (depending on the needs), therefore in general $|W_G| \geq |G|$. This case is also suitable for further reasoning and analysis.

Let us also introduce additional properties of graph change. In particular, we assume that the sequence of gossip matrices $\{W(\mathcal{G}_i)\}_{i=0}^{\infty}$ is a time-homogeneous Markov chain. We define W_σ as σ-field on W_G. We also denote by Q the corresponding Markov kernel and impose the following assumption on the mixing properties of Q:

Assumption 1. *$\{W(\mathcal{G}_k)\}_{k=0}^{\infty}$ is a stationary Markov chain on (W_G, W_σ) with Markov kernel Q and unique invariant distribution π. Moreover, Q is uniformly geometrically ergodic with mixing time $\tau \in \mathbb{N}$, i.e., for every $m \in \mathbb{N}$,*

$$\Delta(\mathrm{Q}^m) = \sup_{W, W' \in W_G} (1/2) \left\| \mathrm{Q}^m(W, \cdot) - \mathrm{Q}^m(W', \cdot) \right\|_{\mathsf{TV}} \leq (1/4)^{\lfloor m/\tau \rfloor}.$$

We also assume that

Assumption 2. *For all $k \in \mathbb{N} \cup \{0\}$, it holds $\mathbb{E}_\pi[W(\mathcal{G}_k)] = \tilde{W}$.*

The matrix $\tilde{W}$ is, in some sense, the keystone for the sequence $\{W(\mathcal{G}_k)\}_{k=0}^{\infty}$. Therefore, we focus on it and introduce some properties of $\tilde{W}$. In particular, we assume that

Assumption 3. *The matrix $\tilde{W}$ satisfies Definition 6, i.e. there exists undirected connected graph $\tilde{\mathcal{G}}$ such that $\tilde{W}$ is a gossip matrix of $\tilde{\mathcal{G}}$.*

For the sake of brevity let us introduce:

$$\lambda_{\max} = \lambda_{\max}(\tilde{W}),\ \lambda_{\min}^{+} = \lambda_{\min}^{+}(\tilde{W}),\ \chi = \frac{\lambda_{\max}(\tilde{W})}{\lambda_{\min}^{+}(\tilde{W})}.$$

Finally, we make the following assumption:

Assumption 4. *For any graph $\mathcal{G}$ of the set G it holds:*

$$\|W(\mathcal{G}) - \tilde{W}\| \leq \rho.$$

To understand what value ρ can take, let us consider the following example.

Example 1. *Let us take gossip matrix of the graph as its Laplacian, i.e. $W(\mathcal{G}) = L(\mathcal{G})$. Also assume that $\mathcal{G}$ and $\mathcal{G}'$ differ in no more than Δ edges, i.e. $|(\mathcal{E} \setminus \mathcal{E}') \cup (\mathcal{E}' \setminus \mathcal{E})| \leq \Delta$. Then we have*

$$W(\mathcal{G}) - W(\mathcal{G}') = \sum_{(i,j)\in\mathcal{E}\setminus\mathcal{E}'} \ell_{ij} - \sum_{(i,j)\in\mathcal{E}'\setminus\mathcal{E}} \ell_{ij},$$

where ℓ_{ij} denotes the mini-Laplacian defined in (2). Note that $\|\ell_{ij}\| \leq 2$. We have

$$\|W(\mathcal{G}) - W(\mathcal{G}')\| \leq \sum_{(i,j)\in(\mathcal{E}\setminus\mathcal{E}')\cup(\mathcal{E}'\setminus\mathcal{E})} \|\ell_{ij}\| \leq 2\Delta.$$

Therefore, Assumption 4 holds with $\rho = 2\Delta$.

This example shows that ρ can be is proportional to the number of distinct edges in graphs. With Assumption 4, one can prove that for any $x \in \mathbb{R}^n$

$$\|W(\mathcal{G})x - \tilde{W}x\| = \left\| \left(W(\mathcal{G}) - \tilde{W}\right)(x - x^*) \right\| \leq \rho \|x - x^*\|, \tag{3}$$

where $x^*_{(i)} = \frac{1}{n}\sum_{j=1}^{n} x_{(j)}$ for $i = 1, \ldots, n$.

Based on the W_0 matrix, we write down the consensus search problem:

$$\begin{aligned} &\min_{x\in\mathbb{R}^n} \left[r(x) = \|\sqrt{\tilde{W}}x\|^2 \right] \\ &\text{s.t.} \sum_{j=1}^{n} x_{(j)} = \sum_{j=1}^{n} x^0_{(j)} \end{aligned}, \tag{4}$$

where x is a vector of local variables, x^0 is an initial vector of local variables. Here we consider that locally each device stores a scalar variable, it is clear that the result can be easily generalize to vectors of local variables.

For the problem (4), we can apply Algorithm 1 from [31] (for convenience, we list it here – see Algorithm 1), which is designed to solve stochastic optimization problems with Markovian nature of randomness. The essence of this method is the use of an unusual random batches (lines 5-7). Note that to calculate such g^k it is necessary to communicate $2^{J_k}B$ times in a row, but send the same values from vector x_g^k. Then it is possible not to additionally send values of x_g^k to a neighbor, which has already been communicated with before.

In terms of convergence we can use Theorem 1 from [31]: the target function of (4) is $\lambda_{\max}$-smooth, Assumptions 1, 2, 4 plunges us in the setting of A3-4 of [31]. But there are also problems that need to be solved. In particular, we need to deal with the fact that the target function from (4) is not strongly convex on $\ker \tilde{W}$. The key problem is that A4 of [31] uses that $\|\nabla F(x,z) - \nabla f(x)\|^2 \leq \sigma^2 + \rho^2\|\nabla f(x)\|^2$, in our case (see (3)), we have $\|\nabla F(x,z) - \nabla f(x)\|^2 \leq \rho^2\|x - x^*\|^2$, then we need to modify the proof of Theorem 1 from [31].

Algorithm 1 Accelerated consensus over graphs with Markovian changes (adapted Algorithm 1 from [31])

1: **Parameters:** stepsize $\gamma > 0$, momentums θ, η, β, p, number of iterations N, batchsize limit M
2: **Initialization:** choose $x_f^0 = x^0$, $T^0 = 0$, set the same random seed for generating $\{J_k\}$ on all devices
3: **for** $k = 0, 1, 2, \ldots, N-1$ **do**
4: $\quad x_g^k = \theta x_f^k + (1-\theta)x^k$
5: $\quad$ Sample $J_k \sim \text{Geom}\,(1/2)$
6: $\quad$ Send x_g^k to neighbors in the networks $\{\mathcal{G}_{T^k+i}\}_{i=1}^{2^{J_k}B}$
7: $\quad$ Compute $g^k = g_0^k + \begin{cases} 2^{J_k}\left(g_{J_k}^k - g_{J_k-1}^k\right), & \text{if } 2^{J_k} \leq M \\ 0, & \text{otherwise} \end{cases}$
$\quad$ with $g_j^k = 2^{-j}B^{-1}\sum_{i=1}^{2^j B} W(\mathcal{G}_{T^k+i})x_g^k$
8: $\quad x_f^{k+1} = x_g^k - p\gamma g^k$
9: $\quad x^{k+1} = \eta x_f^{k+1} + (p-\eta)x_f^k + (1-p)(1-\beta)x^k + (1-p)\beta x_g^k$
10: $\quad T^{k+1} = T^k + 2^{J_k}B$
11: **end for**

Theorem 1. *Let Assumptions 1, 2, 3, 4 hold. Let problem (4) be solved by Algorithm 1. Then for any $b \in \mathbb{N}$,*

$$\gamma \in \left(0; \min\left\{\frac{3}{4\lambda_{\max}}; \frac{\lambda_{\min}^3}{[1800\rho^2(\tau b^{-1} + \tau^2 b^{-2})]^2}\right\}\right),$$

and $\beta, \theta, \eta, p, M, B$ satisfying

$$p = \frac{1}{4}, \quad \beta = \sqrt{\frac{4p^2\mu\gamma}{3}}, \quad \eta = \frac{3\beta}{p\mu\gamma} = \sqrt{\frac{12}{\mu\gamma}}, \quad \theta = \frac{p\eta^{-1} - 1}{\beta p \eta^{-1} - 1},$$

$$M = \max\{2; \sqrt{\tfrac{1}{4}\left(1 + \tfrac{2}{\beta}\right)}\}, \quad B = \lceil b \log_2 M \rceil,$$

it holds that

$$\mathbb{E}\left[\|x^N - x^*\|^2 + \frac{24}{\lambda_{\min}}(r(x_f^N) - r(x^*))\right] = \mathcal{O}\left(\exp\left(-N\sqrt{\frac{p^2\lambda_{\min}\gamma}{3}}\right)\left[\|x^0 - x^*\|^2 + \frac{24}{\lambda_{\min}}(r(x^0) - r(x^*))\right]\right),$$

where $x^*_{(i)} = \frac{1}{n}\sum_{j=1}^n x_{(j)}$ *for* $i = 1, \ldots, n$..

The proof of the theorem are given further in Section 3.3. From Theorem 1 immediately follows the next corollary.

Corollary 1.1. *Under the conditions of Theorem 1, choosing* $b = \tau$ *and* $\gamma \simeq \min\left\{\frac{1}{\lambda_{\max}}; \frac{\lambda_{\min}^3}{\rho^4}\right\}$, *in order to achieve* ε*-approximate solution (in terms of* $\mathbb{E}[\|x - x^*\|^2] \lesssim \varepsilon$*) it takes*

$$\tilde{\mathcal{O}}\left(\tau\left[\sqrt{\chi} + \frac{\rho^2}{\lambda_{\min}^2}\right]\log\frac{1}{\varepsilon}\right) \quad \textit{communications}.$$

3.2 Decentralized optimization with new consensus procedure

Based on Algorithm 1, it is possible to develop a decentralized algorithm for solving the distributed optimization problem (1). The essence of the approach is to use the classical non-distributed algorithm. One can adapt it to a decentralized setup by applying a consensus procedure to the full global gradient calculations. In particular, we take the classical optimal method for smooth convex optimization problems – the accelerated gradient method [32] (Algorithm 2). At each iteration of Algorithm 2, Algorithm 1 is applied when the nodes exchange local gradients with each other (line 5). This approach does not achieve exact consensus, but by making a sufficient number of iterations T it is possible to obtain $v_{i_1}^k \approx v_{i_2}^k$ with high accuracy.

Algorithm 2 Accelerated gradient algorithm for graphs with Markovian changes

1: **Parameters:** stepsize $\gamma > 0$, momentums η, number of iterations N, number of communications T
2: **Initialization:** choose $y_i^0 = x_i^0 = x^0$
3: **for** $k = 0, 1, 2, \ldots, N-1$ **do**
4: Locally compute $\nabla f_i(y_i^k)$
5: Communicate by running T iterations of Algorithm 1 with initialization $\{\nabla f_i(y_i^k)\}_{i=1}^n$ and output $\{v_i^k\}_{i=1}^n$
6: Locally make update: $x_i^{k+1} = y_i^k - \gamma v_i^k$
7: Locally make update: $y_i^{k+1} = x_i^{k+1} + \eta(x_i^{k+1} - x_i^k)$
8: **end for**

The analysis of this kind of algorithms is technical, namely, one need to add small inexactness to the analysis of the basic non-distributed method [24, 33–35]. If we want to solve the optimization problem (1) with precision ε, then by requiring consensus from Algorithm 1 to precision ε^2 or ε^3, we do not feel the effect of consensus inexactness. And therefore the following corollary holds.

Corollary 1.2. *Let the function f from (1) is μ-strongly convex and L-smooth and let Assumptions 1, 2, 3, 4 hold. Let problem (1) be solved by Algorithm 2. Then for*

$$\gamma = \frac{1}{L}, \quad \eta = \frac{\sqrt{L}-\sqrt{\mu}}{\sqrt{L}+\sqrt{\mu}}, \quad T = \tilde{\mathcal{O}}\left(\tau\left[\sqrt{\chi}+\frac{\rho^2}{\lambda_{\min}^2}\right]\log\frac{1}{\varepsilon}\right),$$

it holds that to achieve ε-approximate solution (in terms of $\mathbb{E}[f(x)-f(x^*)] \lesssim \varepsilon$) it takes

$$\tilde{\mathcal{O}}\left(\tau\left[\sqrt{\chi}+\frac{\rho^2}{\lambda_{\min}^2}\right]\log\frac{1}{\varepsilon}\cdot\sqrt{\frac{L}{\mu}}\log\frac{1}{\varepsilon}\right) \quad \textit{communications and}$$

$$\mathcal{O}\left(\sqrt{\frac{L}{\mu}}\log\frac{1}{\varepsilon}\right) \quad \textit{local computations on each node}.$$

From the point of view of local calculations this result is optimal [32]. The situation with communication complexity is more tricky. In the general case the estimate $\tilde{\mathcal{O}}(\chi\cdot\sqrt{L/\mu})$ from [24, 28] is optimal [28]. But our result $\tilde{\mathcal{O}}(\tau[\sqrt{\chi}+(\rho/\lambda_{\min})^2]\cdot\sqrt{L/\mu})$ for the special stochastic Markovian setting can break through the lower bounds from [28], e.g., when τ and $\rho/\lambda_{\min}$ are quite small. In Section 4, we show that deterministic graph changes are more adversarial, and even with the appearance or missing of several edges the lower bounds remain $\tilde{\Omega}(\chi\cdot\sqrt{L/\mu})$, which means that no acceleration in terms of communications is possible.

3.3 Proof of Theorem 1

Before proving Theorem 1, we give the following lemmas.

Lemma 2. *For x^k, x_g^k, x_f^k from Algorithm 1 it holds that $\sum_{j=1}^n x_{(j)}^k = \sum_{j=1}^n (x_f^k)_{(j)} = \sum_{j=1}^n (x_g^k)_{(j)} = \sum_{j=1}^n x_{(j)}^0$.*

Proof. Let us prove by induction. For x^0, x_g^0, x_f^0 the statement of Lemma follows from the initialization of $x_f^0 = x^0$ and line 4. Suppose that $\sum_{j=1}^n x_{(j)}^k = \sum_{j=1}^n (x_f^k)_{(j)} = \sum_{j=1}^n (x_g^k)_{(j)} = \sum_{j=1}^n x_{(j)}^0$. Let us prove that this is also valid for $x^{k+1}, x_g^{k+1}, x_f^{k+1}$. Using the definition of the gossip matrix, we get $\mathbf{1} \in \ker W^T(\mathcal{G}_{T^k+i})$. It means that for $y = W(\mathcal{G}_{T^k+i})x_g^k$, we have $\sum_{j=1}^n y_{(j)} = \mathbf{1}^T y = \mathbf{1}^T W(\mathcal{G}_{T^k+i})x_g^k = 0$. This fact guarantees that $\sum_{j=1}^n (x_f^{k+1})_{(j)} = \sum_{i=1}^n (x_g^k)_{(j)}$. The fact $\sum_{j=1}^n x_{(j)}^{k+1} = \sum_{j=1}^n (x_g^{k+1})_{(j)} = \sum_{i=1}^n x_{(j)}^0$ follows from lines 4 and 5. □

Lemma 3. *For any $x, y \in \mathbb{R}^n$ such that $\sum_{j=1}^n x_{(j)} = \sum_{j=1}^n y_{(j)}$, it holds*

$$r(x) \leq r(y) - \langle \nabla r(x), y - x \rangle - \frac{\lambda_{\min}}{2}\|x - y\|^2.$$

Proof. If $x = y$, the statement of Lemma follows automatically. In the further course of the proof, we assume that $x \neq y$.

Let us prove by contradiction that $(x - y) \notin \ker W_0$. If $(x - y) \in \ker \tilde{W}$, then $x_{(1)} - y_{(1)} = \ldots = x_{(j)} - y_{(j)} = \ldots = x_{(n)} - y_{(n)}$. From the condition of Lemma it is known that $\sum_{j=1}^n x_{(j)} = \sum_{j=1}^n y_{(j)}$, hence we have that $\sum_{j=1}^n [x_{(j)} - y_{(j)}] = n[x_{(1)} - y_{(1)}] = 0$ and $x_{(j)} - y_{(j)} = 0$ for all $j \in [n]$. We come to a contradiction, since $x \neq y$.

Finally, we have that $(x - y) \notin \ker \tilde{W}$. For such x and y, the function $r([x - y]) = \|\sqrt{\tilde{W}}[x - y]\|^2$ is $\lambda_{\min}$-strongly convex. This completes the proof. □

Also to prove Theorem 1, we need Lemmas 4, 5 and 6 from [31].

Lemma 4 (Lemma 4 from [31])**.** *Let Assumptions 4, 1, 2 hold. Then for the gradient estimates g^k from Algorithm 1 it holds that $\mathbb{E}_k[g^k] = \mathbb{E}_k[g^k_{\lfloor \log_2 M \rfloor}]$. Moreover,*

$$\mathbb{E}_k[\|\nabla r(x_g^k) - g^k\|^2] \leq 102\left(\tau B^{-1}\log_2 M + \tau^2 B^{-2}\right)\rho^2\|x_g^k - x^*\|^2,$$
$$\|\nabla r(x_g^k) - \mathbb{E}_k[g^k]\|^2 \leq 86\tau^2 M^{-2}B^{-2}\rho^2\|x_g^k - x^*\|^2.$$

Lemma 5 (Lemma 5 from [31])**.** *For the iterates of Algorithm 1 with $\theta = (p\eta^{-1} - 1)/(\beta p\eta^{-1} - 1)$, $\theta > 0$, $\eta \geq 1$, it holds that*

$$\begin{aligned}\mathbb{E}_k[\|x^{k+1} - x^*\|^2] \leq & (1 + \alpha\gamma\eta)(1 - \beta)\|x^k - x^*\|^2 + (1 + \alpha\gamma\eta)\beta\|x_g^k - x^*\|^2 \\ & + (1 + \alpha\gamma\eta)(\beta^2 - \beta)\|x^k - x_g^k\|^2 + p^2\eta^2\gamma^2\mathbb{E}_k[\|g^k\|^2] \\ & - 2\eta^2\gamma\langle \nabla r(x_g^k), x_g^k + \left(\frac{p}{\eta} - 1\right)x_f^k - \frac{p}{\eta}x^*\rangle \\ & + \frac{p\eta\gamma}{\alpha}\|\mathbb{E}_k[g^k] - \nabla r(x_g^k)\|^2,\end{aligned}$$

where $\alpha > 0$ is any positive constant.

To use the following lemma, we proved Lemmas 2 and 3.

Lemma 6 (Lemma 6 from [31])**.** *Let problem (4) be solved by Algorithm 1. Then for any $u \in \mathbb{R}^n$ such that $\sum_{i=1}^n u_{(i)} = \sum_{i=1}^n x_{(i)}^0$, we get*

$$\begin{aligned}\mathbb{E}_k[r(x_f^{k+1})] \leq & r(u) - \langle \nabla r(x_g^k), u - x_g^k\rangle - \frac{\lambda_{\min}}{2}\|u - x_g^k\|^2 - \frac{\gamma}{2}\|\nabla r(x_g^k)\|^2 \\ & + \frac{\gamma}{2}\|\mathbb{E}_k[g^k] - \nabla r(x_g^k)\|^2 + \frac{\lambda_{\max}\gamma^2}{2}\mathbb{E}_k[\|g^k\|^2].\end{aligned}$$

Proof of Theorem 1. With Lemmas 2 and 3, one can use Lemma 3, Lemma 3 5 with $u = x^*$, $u = x_f^k$ and get

$$\begin{aligned}\mathbb{E}_k[r(x_f^{k+1})] \leq & r(x^*) - \langle \nabla r(x_g^k), x^* - x_g^k\rangle - \frac{\lambda_{\min}}{2}\|x^* - x_g^k\|^2 - \frac{p\gamma}{2}\|\nabla r(x_g^k)\|^2 \\ & + \frac{p\gamma}{2}\|\mathbb{E}_k[g^k] - \nabla r(x_g^k)\|^2 + \frac{\lambda_{\max}p^2\gamma^2}{2}\mathbb{E}_k[\|g^k\|^2],\end{aligned}$$

$$\begin{aligned}\mathbb{E}_k[r(x_f^{k+1})] \leq & r(x_f^k) - \langle \nabla r(x_g^k), x_f^k - x_g^k\rangle - \frac{\lambda_{\min}}{2}\|x_f^k - x_g^k\|^2 - \frac{p\gamma}{2}\|\nabla r(x_g^k)\|^2 \\ & + \frac{p\gamma}{2}\|\mathbb{E}_k[g^k] - \nabla r(x_g^k)\|^2 + \frac{\lambda_{\max}p^2\gamma^2}{2}\mathbb{E}_k[\|g^k\|^2].\end{aligned}$$

Summing the first inequality with coefficient $2p\gamma\eta$, the second with coefficient $2\gamma\eta(\eta - p)$ and the estimate from Lemma 5, we obtain

$$\begin{aligned}&\mathbb{E}_k\left[\|x^{k+1} - x^*\|^2 + 2\gamma\eta^2 r(x_f^{k+1})\right] \\ &\quad \leq (1 + \alpha\gamma\eta)(1 - \beta)\|x^k - x^*\|^2 + (1 + \alpha\gamma\eta)\beta\|x_g^k - x^*\|^2 \\ &\qquad + (1 + \alpha\gamma\eta)(\beta^2 - \beta)\|x^k - x_g^k\|^2 \\ &\qquad - 2\eta^2\gamma\langle \nabla r(x_g^k), x_g^k + \left(\frac{p}{\eta} - 1\right)x_f^k - \frac{p}{\eta}x^*\rangle \\ &\qquad + p^2\eta^2\gamma^2\mathbb{E}_k[\|g^k\|^2] + \frac{p\eta\gamma}{\alpha}\|\mathbb{E}_k[g^k] - \nabla r(x_g^k)\|^2 \\ &\qquad + 2p\gamma\eta\Big(r(x^*) - \langle \nabla r(x_g^k), x^* - x_g^k\rangle - \frac{\lambda_{\min}}{2}\|x^* - x_g^k\|^2 - \frac{p\gamma}{2}\|\nabla r(x_g^k)\|^2 \\ &\qquad + \frac{p\gamma}{2}\|\mathbb{E}_k[g^k] - \nabla r(x_g^k)\|^2 + \frac{\lambda_{\max}p^2\gamma^2}{2}\mathbb{E}_k[\|g^k\|^2]\Big) \\ &\qquad + 2\gamma\eta(\eta - p)\Big(r(x_f^k) - \langle \nabla r(x_g^k), x_f^k - x_g^k\rangle - \frac{\lambda_{\min}}{2}\|x_f^k - x_g^k\|^2 - \frac{p\gamma}{2}\|\nabla r(x_g^k)\|^2 \\ &\qquad + \frac{p\gamma}{2}\|\mathbb{E}_k[g^k] - \nabla r(x_g^k)\|^2 + \frac{\lambda_{\max}p^2\gamma^2}{2}\mathbb{E}_k[\|g^k\|^2]\Big) \\ &\quad = (1 + \alpha\gamma\eta)(1 - \beta)\|x^k - x^*\|^2 + 2\gamma\eta\left(\eta - p\right)r(x_f^k) - 2p\gamma\eta r(x^*) \\ &\qquad + \left((1 + \alpha\gamma\eta)\beta - p\gamma\eta\lambda_{\min}\right)\|x_g^k - x^*\|^2 \\ &\qquad + (1 + \alpha\gamma\eta)(\beta^2 - \beta)\|x^k - x_g^k\|^2 - p\gamma^2\eta^2\|\nabla r(x_g^k)\|^2 \\ &\qquad + \left(\frac{p\eta\gamma}{\alpha} + p\gamma^2\eta^2\right)\|\mathbb{E}_k[g^k] - \nabla r(x_g^k)\|^2 + \left(p^2\eta^2\gamma^2 + p^2\gamma^3\eta^2\lambda_{\max}\right)\mathbb{E}_k[\|g^k\|^2] \\ &\quad \leq (1 + \alpha\gamma\eta)(1 - \beta)\|x^k - x^*\|^2 + 2\gamma\eta\left(\eta - p\right)r(x_f^k) - 2p\gamma\eta r(x^*) \\ &\qquad + \left((1 + \alpha\gamma\eta)\beta - p\gamma\eta\lambda_{\min}\right)\|x_g^k - x^*\|^2 \\ &\qquad + (1 + \alpha\gamma\eta)(\beta^2 - \beta)\|x^k - x_g^k\|^2 - p\gamma^2\eta^2\|\nabla r(x_g^k)\|^2 \\ &\qquad + p\eta\gamma\left(\frac{1}{\alpha} + \gamma\eta\right)\|\mathbb{E}_k[g^k] - \nabla r(x_g^k)\|^2\end{aligned}$$

$$+ 2p^2\eta^2\gamma^2\left(1+\gamma\lambda_{\max}\right)\mathbb{E}_k[\|g^k-\nabla r(x_g^k)\|^2]$$
$$+ 2p^2\eta^2\gamma^2\left(1+\gamma\lambda_{\max}\right)\mathbb{E}_k[\|\nabla r(x_g^k)\|^2]\,.$$

In the last step we also used Cauchy Schwartz inequality. The choice of $\alpha=\frac{\beta}{2\eta\gamma}$ gives $(1+\alpha\eta\gamma)(1-\beta)=\left(1+\frac{\beta}{2}\right)(1-\beta)\leq\left(1-\frac{\beta}{2}\right)$ and then

$$\begin{aligned}
&\mathbb{E}_k[\|x^{k+1}-x^*\|^2+2\gamma\eta^2 r(x_f^{k+1})]\\
&\quad\leq\left(1-\frac{\beta}{2}\right)\|x^k-x^*\|^2+2\gamma\eta\left(\eta-p\right)r(x_f^k)-2p\gamma\eta r(x^*)\\
&\qquad+\left(\left(1+\frac{\beta}{2}\right)\beta-p\gamma\eta\lambda_{\min}\right)\|x_g^k-x^*\|^2+\left(1+\frac{\beta}{2}\right)(\beta^2-\beta)\|x^k-x_g^k\|^2\\
&\qquad+p\eta^2\gamma^2\left(1+\frac{2}{\beta}\right)\|\mathbb{E}_k[g^k]-\nabla r(x_g^k)\|^2\\
&\qquad+2p^2\eta^2\gamma^2\left(1+\gamma\lambda_{\max}\right)\mathbb{E}_k[\|g^k-\nabla r(x_g^k)\|^2]\\
&\qquad-p\eta^2\gamma^2\left(1-2p\left(1+\gamma\lambda_{\max}\right)\right)\mathbb{E}_k[\|\nabla r(x_g^k)\|^2]\,.
\end{aligned}$$

Subtracting $2\gamma\eta^2 r(x^*)$ from both sides, we get

$$\begin{aligned}
&\mathbb{E}_k[\|x^{k+1}-x^*\|^2+2\gamma\eta^2(r(x_f^{k+1})-r(x^*))]\\
&\quad\leq\left(1-\frac{\beta}{2}\right)\|x^k-x^*\|^2+2\gamma\eta^2\left(1-\frac{p}{\eta}\right)(r(x_f^k)-r(x^*))\\
&\qquad+\left(\left(1+\frac{\beta}{2}\right)\beta-p\gamma\eta\lambda_{\min}\right)\|x_g^k-x^*\|^2+\left(1+\frac{\beta}{2}\right)(\beta^2-\beta)\|x^k-x_g^k\|^2\\
&\qquad+p\eta^2\gamma^2\left(1+\frac{2}{\beta}\right)\|\mathbb{E}_k[g^k]-\nabla r(x_g^k)\|^2\\
&\qquad+2p^2\eta^2\gamma^2\left(1+\gamma\lambda_{\max}\right)\mathbb{E}_k[\|g^k-\nabla r(x_g^k)\|^2]\\
&\qquad-p\eta^2\gamma^2\left(1-2p\left(1+\gamma\lambda_{\max}\right)\right)\mathbb{E}_k[\|\nabla r(x_g^k)\|^2]\,.
\end{aligned}$$

Applying Lemma 4, one can obtain

$$\begin{aligned}
&\mathbb{E}_k[\|x^{k+1}-x^*\|^2+2\gamma\eta^2(r(x_f^{k+1})-r(x^*))]\\
&\quad\leq\left(1-\frac{\beta}{2}\right)\|x^k-x^*\|^2+2\gamma\eta^2\left(1-\frac{p}{\eta}\right)(r(x_f^k)-r(x^*))\\
&\qquad+\left(\left(1+\frac{\beta}{2}\right)\beta-p\gamma\eta\lambda_{\min}\right)\|x_g^k-x^*\|^2+\left(1+\frac{\beta}{2}\right)(\beta^2-\beta)\|x^k-x_g^k\|^2\\
&\qquad+p\eta^2\gamma^2\left(1+\frac{2}{\beta}\right)\cdot 86\tau^2M^{-2}B^{-2}\rho^2\|x_g^k-x^*\|^2\\
&\qquad+2p^2\eta^2\gamma^2\left(1+\gamma\lambda_{\max}\right)\cdot 102\left(\tau B^{-1}\log_2 M+\tau^2B^{-2}\right)\rho^2\|x_g^k-x^*\|^2\\
&\qquad-p\gamma^2\eta^2(1-2p(1+\gamma\lambda_{\max}))\|\nabla r(x_g^k)\|^2\,.
\end{aligned}$$

With $M \geq \sqrt{\frac{1+2/\beta}{p}}$, we have

$$\begin{aligned}
&\mathbb{E}_k[\|x^{k+1} - x^*\|^2 + 2\gamma\eta^2(r(x_f^{k+1}) - r(x^*))] \\
&\quad \leq \left(1 - \frac{\beta}{2}\right)\|x^k - x^*\|^2 + 2\gamma\eta^2\left(1 - \frac{p}{\eta}\right)(r(x_f^k) - r(x^*)) \\
&\quad + \left(\left(1 + \frac{\beta}{2}\right)\beta - p\gamma\eta\lambda_{\min}\right)\|x_g^k - x^*\|^2 + \left(1 + \frac{\beta}{2}\right)(\beta^2 - \beta)\|x^k - x_g^k\|^2 \\
&\quad + 300p^2\eta^2\gamma^2\rho^2\left(1 + \gamma\lambda_{\max}\right)\left(\tau B^{-1}\log_2 M + \tau^2 B^{-2}\right)\|x_g^k - x^*\|^2 \\
&\quad - p\gamma^2\eta^2(1 - 2p(1 + \gamma\lambda_{\max}))\|\nabla r(x_g^k)\|^2 .
\end{aligned}$$

With $\gamma \leq \frac{3}{4\lambda_{\max}}$, using that $p = \frac{1}{4}$, $\beta = \sqrt{\frac{4p^2\lambda_{\min}\gamma}{3}}$, and $p\lambda_{\min}\gamma\eta = 3\beta$, one can obtain

$$\begin{gathered}
\beta = \sqrt{\tfrac{4p^2\lambda_{\min}\gamma}{3}} \leq \sqrt{\tfrac{p^2\lambda_{\min}}{\lambda_{\max}}} \leq 1, \\
1 + \gamma\lambda_{\max} \leq 2, \quad -(1 - 2p(1 + \gamma\lambda_{\max})) \leq 0, \\
\left(\left(1 + \tfrac{\beta}{2}\right)\beta - p\gamma\eta\lambda_{\min}\right) = \left(\beta + \tfrac{\beta^2}{2} - p\lambda_{\min}\gamma\eta\right) \leq \left(\tfrac{3\beta}{2} - p\lambda_{\min}\gamma\eta\right) \leq -\tfrac{p\lambda_{\min}\gamma\eta}{2}.
\end{gathered}$$

and, therefore,

$$\begin{aligned}
&\mathbb{E}_k[\|x^{k+1} - x^*\|^2 + 2\gamma\eta^2(r(x_f^{k+1}) - r(x^*))] \\
&\quad \leq \left(1 - \frac{\beta}{2}\right)\|x^k - x^*\|^2 + 2\gamma\eta^2\left(1 - \frac{p}{\eta}\right)(r(x_f^k) - r(x^*)) \\
&\quad + \left(-\frac{p\gamma\eta\lambda_{\min}}{2} + 300p^2\eta^2\gamma^2\rho^2\left(\tau B^{-1}\log_2 M + \tau^2 B^{-2}\right)\right)\|x_g^k - x^*\|^2 .
\end{aligned}$$

Since $p = \frac{1}{4}$, $\eta = \sqrt{\frac{12}{\lambda_{\min}\gamma}}$, $\gamma \leq \frac{\lambda_{\min}^3}{[1800\rho^2(\tau b^{-1} + \tau^2 b^{-2})]^2}$ and $B = \lceil b\log_2 M \rceil$, we get

$$\begin{gathered}
\gamma \leq \tfrac{\lambda_{\min}^3}{[1800\rho^2(\tau b^{-1} + \tau^2 b^{-2})]^2} \leq \tfrac{\lambda_{\min}^3}{[1800\rho^2(\tau B^{-1}\log_2 M + \tau^2 B^{-2})]^2}, \\
\left(-\tfrac{p\gamma\eta\lambda_{\min}}{2} + 300p^2\eta^2\gamma^2\rho^2\left(\tau B^{-1}\log_2 M + \tau^2 B^{-2}\right)\right) \\
= \tfrac{p\gamma\eta}{2}\left(-\lambda_{\min} + 150\eta\gamma\rho^2\left(\tau B^{-1}\log_2 M + \tau^2 B^{-2}\right)\right) \\
= \tfrac{p\gamma\eta}{2}\left(-\lambda_{\min} + \sqrt{\tfrac{12\cdot 150^2\gamma}{\lambda_{\min}}}\cdot\rho^2\left(\tau B^{-1}\log_2 M + \tau^2 B^{-2}\right)\right) \leq 0,
\end{gathered}$$

and, then,

$$\begin{aligned}
&\mathbb{E}_k[\|x^{k+1} - x^*\|^2 + 2\gamma\eta^2(r(x_f^{k+1}) - r(x^*))] \\
&\qquad \leq \left(1 - \frac{\beta}{2}\right)\|x^k - x^*\|^2 + 2\gamma\eta^2\left(1 - \frac{p}{\eta}\right)(r(x_f^k) - r(x^*)) .
\end{aligned}$$

Using that $\eta = \sqrt{\frac{12}{\lambda_{\min}\gamma}}$ and $\frac{\beta}{2} = \frac{p}{\eta}$, we have

$$\mathbb{E}_k\left[\|x^{k+1} - x^*\|^2 + \frac{24}{\lambda_{\min}}(r(x_f^{k+1}) - r(x^*))\right] \\ \leq \left(1 - \frac{\beta}{2}\right)\left[\|x^k - x^*\|^2 + \frac{24}{\lambda_{\min}}(r(x_f^k) - r(x^*))\right].$$

Substituting of $\beta = \sqrt{\frac{4p^2\lambda_{\min}\gamma}{3}}$, we have

$$\mathbb{E}_k\left[\|x^{k+1} - x^*\|^2 + \frac{24}{\lambda_{\min}}(r(x_f^{k+1}) - r(x^*))\right] \\ \leq \left(1 - \sqrt{\frac{p^2\lambda_{\min}\gamma}{3}}\right)\left[\|x^k - x^*\|^2 + \frac{24}{\lambda_{\min}}(r(x_f^k) - r(x^*))\right].$$

Taking the full expectation and running the recursion finish the proof. □

4 Lower bounds for slowly time-varying graphs

4.1 First-order decentralized algorithms

Similar to the works [3, 28, 29] we will impose some conditions on the optimization algorithm. We will call the class of algorithms satisfying these conditions *first-order decentralized algorithms*. Each algorithm in this class has two types of iterations: communicational and local. In the communicational iteration, the nodes communicate with each other, while in the local iteration, they perform computations on their local memory. For each time step $k \in \mathbb{N}$ we will call $\mathcal{H}_i(k)$ the local memory of the node i. Also for each time step $k \in \mathbb{N}$, denote the last preceding communication time by $q(k)$.
1. If nodes perform a local computation at step k, local information is updated as

$$\mathcal{H}_i(k+1) \subseteq \text{span}\left(\{x, \nabla f_i(x), \nabla f_i^*(x) : \ x \in \mathcal{H}_i(k)\}\right)$$

for all $i = 1, \ldots, n$. Here $f^*(y) = \sup_x\{y^T x - f(x)\}$ is the Fenchel's dual function.

2. If the nodes perform a communication round at time step k, local information is updated as

$$\mathcal{H}_i(k+1) \subseteq \text{span}\left(\bigcup_{j \in \mathcal{N}_i^{q(k)} \cup \{i\}} \mathcal{H}_j(k)\right)$$

for all $i = 1, \ldots, n$. Here $\mathcal{N}_i^{q(k)}$ is a set of neighbors of node i at time step $q(k)$, i.e. at the time of last communication.

4.2 Overview of main result

In order to establish a new lower bound, we will employ a slightly different concept than that found in previous works [3, 28, 29]. The fundamental idea behind such lower bound approaches is to construct a counterexample of a time-varying network, in which information flows slowly from one large cluster to another, while maintaining a modest characteristic number for the network. In previous studies [3, 28], this was achieved by utilizing classical (unweighted) graph Laplacians corresponding to the network. In our novel approach, we employ a weighted Laplacian to construct a counterexample.

To estimate the characteristic number of a graph, it is necessary to evaluate both the largest and smallest nonzero eigenvalues of the Laplacian. Although maximal eigenvalue can be easily estimated, difficulties arise in determining the minimal positive eigenvalue. The literature contains numerous typical graphs (such as paths, stars, and complete binary trees) for which lower nonzero eigenvalues of Laplacians have been calculated, and these can be employed for counterexamples. Our approach, which is based on weighted Laplacians, allows to utilize previously inaccessible topologies.

Now we formulate our result on lower bounds as follows. The proof will be provided in the forthcoming sections.

Theorem 2. *For any $\chi \geq 56$, $L > 16\mu > 0$ and any first-order optimization method $\mathcal{M}$ there exists a set of L-smooth and μ-strongly convex functions $\{f_i\}_{i=1}^m$, a sequence of graphs $\{\mathcal{G}_k = (\mathcal{V}, \mathcal{E}_k)\}_{k=0}^\infty$ and a sequence of corresponding gossip matrices $\{W(\mathcal{G}_k)\}_{k=0}^\infty$ such that for each $k = 0, 1, \ldots$ it holds $\chi(W(\mathcal{G}_k)) = \chi$ and*

$$\|x_k - x_*\|^2 \geq \left(1 - 4\sqrt{\frac{\mu}{L}}\right)^{\frac{72k}{7\chi}+2} \|x_0 - x_*\|^2.$$

Corollary 2.1. *For any $\chi \geq 56$, $L > 16\mu > 0$ and any first-order optimization method $\mathcal{M}$ there exists a set of L-smooth and μ-strongly convex functions $\{f_i\}_{i=1}^m$, a sequence of graphs $\{\mathcal{G}_k = (\mathcal{V}, \mathcal{E}_k)\}_{k=0}^\infty$ and a sequence of corresponding gossip matrices $\{W(\mathcal{G}_k)\}_{k=0}^\infty$ such that for each $k = 0, 1, \ldots$ it holds $\chi(W(\mathcal{G}_k)) = \chi$ and method $\mathcal{M}$ requires at least $\Omega(\chi\sqrt{L/\mu}\log(1/\varepsilon))$ communication rounds.*

4.3 Auxiliary lemmas for weighted Laplacians

The weighted Laplacian enables us to adjust the edge weights in such a manner that the smallest nonzero eigenvalue can be easily estimated while simultaneously maintaining control over the largest eigenvalue. We upper bound the largest eigenvalue through the following lemma.

Lemma 7. *Let $\mathcal{G}_A = (\mathcal{V}, \mathcal{E}, A)$ be a weighted graph. Let $d_{\max}(\mathcal{G}_A) = \max_{i\in\mathcal{V}} \sum_{(i,j)\in\mathcal{E}} a_{ij}$ denote the maximum vertex degree of $\mathcal{G}_A$. Then $\lambda_{\max}(L(\mathcal{G}_A)) \leq 2d_{\max}(\mathcal{G}_A)$.*

Proof. This is a classical result, let us give a proof of it. Let $d_i = \sum_{(i,k)\in\mathcal{E}} a_{ik}$.

First, we observe that $\sum_{j=0}^n [L(\mathcal{G}_A)]_{ij} = 0$ for each i. This implies that the row sum of the i-th row coincides with the diagonal element d_i:

$$\sum_{j\neq i} |[L(\mathcal{G}_A)]_{ij}| = \sum_{(i,j)\in\mathcal{E}} a_{ij} = d_i.$$

Let $\overline{B}(a,R)$ denote a closed disc in complex plane with center at a and radius R. By the Gershgorin circle theorem, we have

$$\lambda_1 \in \bigcup_{i=1}^{n} \overline{B}(d_i,d_i) = \overline{B}(d_{max}(\mathcal{G}_A), d_{max}(\mathcal{G}_A)).$$

As a result, $\lambda_{\max}(\mathcal{G}_A) \leq 2d_{\max}(\mathcal{G}_A)$. □

Lemma 8. *For any unweighted graph $\mathcal{G} = (\mathcal{V},\mathcal{E})$ there exists a weighted graph $\mathcal{G}_A = (\mathcal{V},\mathcal{E},A)$ such that $\chi(L(\mathcal{G}_A)) \leq 2Dn$, where $n = |\mathcal{V}|$ and D is the diameter of $\mathcal{G}$.*

Proof. For each edge $(i,j) \in \mathcal{E}$, let s_{ij} denote an arbitrary shortest path from i to j (in the unweighted graph $\mathcal{G}$). Let us build a weighted graph $\mathcal{G}_A = (\mathcal{V},\mathcal{E},A)$ in the following way. Assign each of the edges $(i,j) \in \mathcal{E}$ a weight a_{ij} equal to the number of shortest paths traversing edge (i,j), i.e.

$$a_{ij} = |s_{kl} :\ (i,j) \in s_{kl}|,$$

and set $a_{ij} = 0$ if $(i,j) \notin \mathcal{E}$. Note that the Laplacian of $\mathcal{G}_A$ has the form

$$L(\mathcal{G}_A) = \sum_{(i,j)\in E} a_{ij}\ell_{ij}.$$

For a fixed shortest path s_{ij} of length d in the graph, let $s_{ij} = \{i = v_0,\ldots,v_d = j\}$, and let S_{ij} denote the sum of mini-Laplacians corresponding to edges in the path s_{ij}, i.e.,

$$S_{ij} = \ell_{iv_1} + \ell_{v_1v_2} + \ldots + \ell_{v_{d-1}j}.$$

Note that the Laplacian of $\mathcal{G}_A$ can be also written as

$$L(\mathcal{G}_A) = \sum_{(i,j)\in\mathcal{E}} S_{ij}.$$

To estimate $\lambda_{n-1}(L(\mathcal{G}_A))$, we use the theorem stating that if A,B are symmetric matrices and $A \succeq B$, then $\lambda_i(A) \geq \lambda_i(B)$ for all $i \in 1,\ldots,n$, where $\lambda_i(A)$ and $\lambda_i(B)$ are sorted in a descending order. We will show that for any shortest path s_{ij} with length d,

$$S_{ij} = \ell_{iv_1} + \ell_{v_1v_2} + \cdots + \ell_{v_{d-1}j} \succeq \frac{1}{d}\ell_{ij}.$$

Let $x \in \mathbb{R}^n$. Then

$$\begin{aligned} x^T S_{ij} x &= x^T(\ell_{iv_1} + \ell_{v_1v_2} + \cdots + \ell_{v_{d-1}j})x = \frac{1}{d}\left(d\sum_{l=0}^{d-1}(x_{v_l} - x_{v_{l+1}})^2\right) \\ &= \frac{1}{d}\left(\left(\sum_{l=0}^{d-1}1^2\right)\sum_{l=0}^{d-1}(x_{v_l} - x_{v_{l+1}})^2\right) \overset{①}{\geq} \frac{1}{d}\left(\sum_{l=0}^{d-1}1\cdot(x_{v_l} - x_{v_{l+1}})\right)^2 \\ &\geq \frac{1}{d}(x_{v_0} - x_{v_d})^2 = \frac{1}{d}x^T\ell_{ij}x, \end{aligned}$$

where ① holds by Cauchy–Schwarz inequality. Since $x^\top s_{ij} x \geq \frac{1}{d}x^\top \ell_{ij} x$ holds for any $x \in \mathbb{R}^n$, we have

$$S_{ij} \succeq \frac{1}{d}\ell_{ij}.$$

Now, we sum these lower bounds for all pairs of distinct vertices i and j in the graph, obtaining

$$L(\mathcal{G}_A) = \sum_{(i,j)\in\mathcal{E}} S_{ij} \succeq \frac{1}{D}\sum_{(i,j)\in\mathcal{E}} \ell_{ij}.$$

Thus, our matrix $L(\mathcal{G}_A)$ is not smaller than the Laplacian corresponding to the complete graph divided by D. As it is known, the spectrum of the complete graph consists of the number n with multiplicity $n-1$ and 0 with multiplicity 1. Therefore,

$$\lambda^+_{\min}(L(\mathcal{G}_A)) \geq \frac{n}{D}. \tag{5}$$

Next, by Lemma 7 we have that $\lambda_1(L(\mathcal{G}_A)) \leq 2d_{\max}(\mathcal{G}_A)$. To estimate $d_{\max}(\mathcal{G}_A)$, note that each shortest path from i to j passes through a fixed vertex v at most once, so the adjacent edges to v participate in the sum of mini-Laplacians for the path from i to j at most twice. There are $\frac{n(n-1)}{2}$ pairs of vertices. Therefore,

$$d_{\max}(\mathcal{G}_A) \leq 2\frac{n(n-1)}{2} < n^2.$$

As a result, we have

$$\frac{\lambda_{\max}(L(\mathcal{G}_A))}{\lambda^+_{\min}(L(\mathcal{G}_A))} \leq \frac{2n^2}{n/D} \leq 2nD,$$

which completes the proof. □

4.4 Counterexample graph sequence

The structure of the further proof will then be identical to the structures in the articles [3, 28, 29]. Let us describe the structure of the counterexample network.

Definition 7. *Consider two star graphs with the number of vertices a and b, respectively. Let us add an additional isolated vertex to these two graphs and connect it with edges to the centers of the stars. We denote the resulting graph by $T_{a,b}$.*

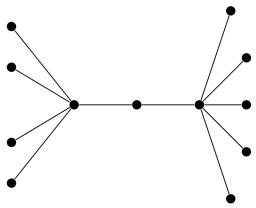

Fig. 1 Graph $T_{4,5}$.

Consider the graph $T_{n,n}$, which consists of two "glued-together stars": the left and the right. We will refer to the set of pendant vertices adjacent to the center of the left star as the left partition. The right partition is defined similarly. Let us take the left partition and select $\left\lceil \frac{n}{2} \right\rceil$ vertices from it, denoting them $\mathcal{V}_1$. In the right partition, also select $\left\lceil \frac{n}{2} \right\rceil$ vertices and denote them $\mathcal{V}_2$. Next, we will introduce functions on the vertices of this graph.

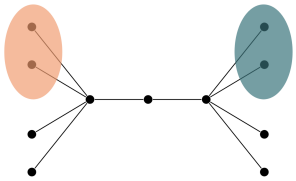

Fig. 2 An example of the graph $T_{4,4}$ with highlighted sets $\mathcal{V}_1$ and $\mathcal{V}_2$. The vertices in the red region belong to the set $\mathcal{V}_1$, while the vertices in the blue region belong to the set $\mathcal{V}_2$.

From now on, let us consider a graph $T_{n,n}$ with $n \geq 2$.

Denote the vertex functions $f_v : \ell_2 \to \mathbb{R}$ depending on vertex type:

$$f_v(x) = \begin{cases} \frac{\mu}{2n} \|x\|^2 + \frac{L-\mu}{4|\mathcal{V}_2|} \sum_{k=1}^{\infty} (x_{2k-1} - x_{2k})^2, & v \in \mathcal{V}_1, \\ \frac{\mu}{2n} \|x\|^2 + \frac{L-\mu}{4|\mathcal{V}_1|} \left[(x_1 - 1)^2 + \sum_{k=1}^{\infty} (x_{2k} - x_{2k+1})^2 \right], & v \in \mathcal{V}_2, \\ \frac{\mu}{2n} \|x\|^2, & v \in \mathcal{V} \setminus \mathcal{V}_1 \setminus \mathcal{V}_2. \end{cases} \tag{6}$$

From the definition of $\mathcal{V}_1$ and $\mathcal{V}_2$ we can deduce that

$$|\mathcal{V}_1| = |\mathcal{V}_2| \geq \frac{n}{4}. \tag{7}$$

Let us estimate the network's global characteristic number using the local one

$$\kappa_l = \frac{\frac{L-\mu}{|\mathcal{V}_1|} + \frac{\mu}{n}}{\frac{\mu}{n}} = \frac{n}{|\mathcal{V}_1|}(\kappa_g - 1) + 1 \leq 4(\kappa_g - 1) + 1.$$

Thus, we have

$$\kappa_g \geq \frac{\kappa_l - 1}{4} + 1. \tag{8}$$

In the following, we will describe the structure of changes in our graph. In total, our graph sequence will be divided into two alternating phases: Phase 1 - "flow of vertices from left to right" and Phase 2 - "flow of vertices from right to left." Let us consider the very first graph in our sequence. We will take the previously described graph $T_{n,n}$ with highlighted sets $\mathcal{V}_1, \mathcal{V}_2$ and transfer all unmarked vertices of the right partition to the left partition. The resulting graph will have the form $T_{2n-\left[\frac{n}{2}\right],\left[\frac{n}{2}\right]}$, which will be the first element of our sequence.

Next, let us consider the first phase and define it by induction. We have defined the first graph of the first phase. Suppose that we have the n-th element of the first phase and let it have the form $T_{a,b}$, where $a+b=2n$, $a \geq [n/2]+1$, $b \geq [n/2]$, and we assume that the whole set $\mathcal{V}_1$ belongs to the left partition and the set $\mathcal{V}_2$ to the right partition. Let us denote the left center of the star as l, and the right center as r. Let us denote the vertex connecting l and r by v, and denote by u any unmarked vertex of the left partition. We then transfer the vertex v to the right partition and place the vertex u in the position of v (that is, make it the connecting vertex). As a result, we removed the edge (v, l) and added the edge (u, r). As a result, we get a graph of the form $T_{a-1,b+1}$.

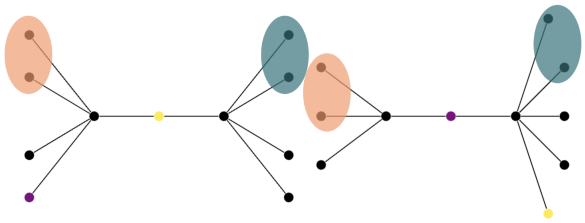

Fig. 3 Here is an example of the graph changing scheme. In the left image, the 3rd iteration of Phase 1 is illustrated, and in the right image, the 4th iteration is presented. The red and blue areas respectively consist of the sets $\mathcal{V}_1$ and $\mathcal{V}_2$. The vertex v is marked in yellow, while the vertex u is indicated in purple.

The first phase continues until the graph takes the form $T_{\left[\frac{n}{2}\right],2n-\left[\frac{n}{2}\right]}$. At this iteration, the first phase ends and the second phase begins (i.e. they intersect at one element). The second phase is defined similarly but symmetrically, with vertices now "flowing" from the right partition to the left partition.

Now, let us consider a model where we assume that the vertices in $\mathcal{V}_2$ contain some information in their local memory, and they need to transfer this information to the vertices in $\mathcal{V}_1$. Vertices perform a communication iteration after each graph change and

can share information with their neighbors. If we start to change the graph from the first iteration of the first phase, it is not difficult to see that a minimum of $2n - 2\left[\frac{n}{2}\right]$ communication iterations will be required for the information transfer (that is, in the graph at the number $2n - 2\left[\frac{n}{2}\right]$, the vertices $\mathcal{V}_1$ will not yet possess the information). Similarly, the reasoning for transferring information from vertices $\mathcal{V}_1$ to $\mathcal{V}_2$ can be applied during the second phase.

Let us define t as the time it takes to transfer information from one partition to the other one. We have

$$t = 2n - 2\left[\frac{n}{2}\right] \geq n. \tag{9}$$

4.5 Proof of Theorem 2

The next explanation is taken from [29].

Let $x_0 = 0$ be the initial point for the first-order decentralized algorithm. For every $m \geq 1$, we define $l_m = \min\{k \geq 1 | \exists v : \exists x \in \mathcal{H}_v(k) : x_m \neq 0\}$ as the first moment when we can get a nonzero element at the m-th place at any node.

Considering the functions on vertices from $\mathcal{V}_1$ and $\mathcal{V}_2$, we can conclude that functions on vertices from $\mathcal{V}_1$ can "transfer" (by calculating the gradient) information (nonzero element) from the odd positions $(1, 3, 5, \ldots)$ to the next even positions $(2, 4, 6, \ldots$ correspondingly). At the same time, functions on vertices from $\mathcal{V}_2$ can transfer information from the even positions $(2, 4, 6, \ldots)$ to the next odd positions $(3, 5, 7, \ldots)$. Therefore, for the network to get a new nonzero element at the next position, a whole phase is required, that is, t communication iterations.

To reach the m-th nonzero element, we need to make at least m local steps and $(m-1)t$ communication steps to transfer information from gradients between $\mathcal{V}_1$ and $\mathcal{V}_2$ sets. Therefore, the time l_m at which m-th element becomes nonzero can be estimated as

$$l_m \geq (m-1)t + m. \tag{10}$$

The solution of the global optimization problem is $x_k^* = \left(\frac{\sqrt{\kappa_g}-1}{\sqrt{\kappa_g}+1}\right)^k$.

For any m, k such that $l_m > k$

$$\|x_k - x_*\|^2 \geq (x_*)_m^2 + (x_*)_{m+1}^2 + \ldots = \left(\frac{\sqrt{\kappa_g}-1}{\sqrt{\kappa_g}+1}\right)^m \|x_0 - x_*\|^2.$$

Using (10) we can take $m = \lceil \frac{k}{t+1} \rceil + 1$. From (8) we conclude that $\frac{\sqrt{\kappa_g}-1}{\sqrt{\kappa_g}+1} \geq 1 - \frac{4}{\sqrt{\kappa_l}}$

Therefore using (8) and (9) we get

$$\|x_k - x_*\|^2 \geq \left(1 - 4\sqrt{\frac{\mu}{L}}\right)^{\lceil k/(n+1) \rceil + 1} \|x_0 - x_*\|^2.$$

For each graph in our sequence we map a weighted Laplacian from (8), so $\chi \leq 8n$.

Rearranging an expression, we get

$$\|x_k - x_*\|^2 \geq \left(1 - 4\sqrt{\frac{\mu}{L}}\right)^{\frac{8k}{\chi}+2} \|x_0 - x_*\|^2 . \tag{11}$$

Note that our reasoning holds for any value χ, L, μ satisfying the following conditions: $\chi = 8(2n+3), n \in \mathbb{N}, n \geq 2,\ L > 16\mu > 0$. Let us explain why such an estimation is appropriate for any $\chi \geq 56$. Let $\chi \geq 56$, take the closest to it from below $\chi_0 = 8(2n+3) \geq 56$. Then it is not difficult to see that

$$\chi_0/\chi \geq \frac{7}{9}. \tag{12}$$

Let us take our sequence of graph counterexamples for values χ_0, L, μ. Take a lower bound for them

$$\|x_k - x_*\|^2 \geq \left(1 - 4\sqrt{\frac{\mu}{L}}\right)^{\frac{8k}{\chi_0}+2} \|x_0 - x_*\|^2 . \tag{13}$$

Then let us "tweak" the weights of the edges so that the characteristic numbers of their weighted Laplacean numbers increase up to χ. This can always be done by taking any edge and decreasing its weight to 0, then the smallest positive eigenvalue will go to infinity. Using (12) and (13) we obtain

$$\|x_k - x_*\|^2 \geq \left(1 - 4\sqrt{\frac{\mu}{L}}\right)^{\frac{72k}{7\chi}+2} \|x_0 - x_*\|^2 .$$

5 Conclusion

In this paper, we study the lower bounds of decentralized optimization problems in the case of a slowly changing communication network. Specifically, we consider the case of changing at most two edges per iteration. A lower bound $\chi\sqrt{L/\mu}\log(1/\varepsilon)$ was obtained, which coincides with the lower bound in the class of problems with arbitrary change of edges per iteration. Thus, the question raised in [29] can be considered closed in the formulation under discussion. However, there are some open questions, such as whether it is possible to obtain acceleration in the case when the graph changes even more slowly (for example, when no more than $\log n$ or *const* edges change per n iterations), or whether it is possible to obtain acceleration on average in the case when changes of the edges are random.

In the case of a Markovian-varying network, we also obtain a consensus result that allows for the construction of an optimization algorithm whose convergence rate is similar (up to a logarithmic factor) to that of the static case, but with the addition of a term characterized by a Markovian property. Perhaps the extra logarithm could be avoided by constructing a more sophisticated method.

Although the results of the lower bounds are quite pessimistic and argue that no acceleration can be achieved with arbitrary slow changes, some examples (including

results in the Markov setting) show that under certain conditions an improvement can be made, and finding such conditions is of interest.

Our work is rooted in a theoretical and mathematical framework, therefore it does not involve the analysis or generation of any datasets.

6 Acknowledgements

This work was supported by a grant for research centers in the field of artificial intelligence, provided by the Analytical Center for the Government of the Russian Federation in accordance with the subsidy agreement (agreement identifier 000000D730321P5Q0002) and the agreement with the Moscow Institute of Physics and Technology dated November 1, 2021 No. 70-2021-00138.

References

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

[2] Nedić, A., Ozdaglar, A.: Subgradient methods for saddle-point problems. Journal of optimization theory and applications **142**(1), 205–228 (2009)

[3] Scaman, K., Bach, F., Bubeck, S., Lee, Y.T., Massoulié, L.: Optimal algorithms for smooth and strongly convex distributed optimization in networks. In: Proceedings of the 34th International Conference on Machine Learning-Volume 70, pp. 3027–3036 (2017). JMLR.org

[4] Iacca, G.: Distributed optimization in wireless sensor networks: an island-model framework. Soft Computing **17**(12), 2257–2277 (2013) https://doi.org/10.1007/s00500-013-1091-x

[5] Chen, T., Luo, J., Deng, Z., Zuo, X., Zhou, X., Liu, Y.-m.: Distributed algorithm design for resource allocation problems of second-order multi-agent systems. In: 2022 41st Chinese Control Conference (CCC), pp. 4538–4542 (2022). https://doi.org/10.23919/CCC55666.2022.9901830

[6] Cai, K., Ishii, H.: Average consensus on arbitrary strongly connected digraphs with time-varying topologies. IEEE Transactions on Automatic Control **59**(4), 1066–1071 (2014)

[7] Xiao, L., Boyd, S.: Fast linear iterations for distributed averaging. Systems & Control Letters **53**(1), 65–78 (2004) https://doi.org/10.1016/j.sysconle.2004.02.022

[8] Bazerque, J.A., Giannakis, G.B.: Distributed spectrum sensing for cognitive radio networks by exploiting sparsity. IEEE Transactions on Signal Processing **58**(3), 1847–1862 (2009)

[9] Ren, W., Beard, R.W.: Distributed Consensus in Multi-vehicle Cooperative Control vol. 27. Springer (2008)

[10] Olshevsky, A.: Efficient information aggregation strategies for distributed control and signal processing. arXiv preprint arXiv:1009.6036 (2010)

[11] Ren, W.: Consensus based formation control strategies for multi-vehicle systems. In: 2006 American Control Conference, p. 6 (2006). IEEE

[12] Jadbabaie, A., Lin, J., Morse, A.S.: Coordination of groups of mobile autonomous agents using nearest neighbor rules. IEEE Transactions on automatic control **48**(6), 988–1001 (2003)

[13] Rabbat, M., Nowak, R.: Distributed optimization in sensor networks. In: Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks, pp. 20–27 (2004)

[14] Forero, P.A., Cano, A., Giannakis, G.B.: Consensus-based distributed support vector machines. Journal of Machine Learning Research **11**(5) (2010)

[15] Nedić, A., Olshevsky, A., Uribe, C.A.: Fast convergence rates for distributed non-bayesian learning. IEEE Transactions on Automatic Control **62**(11), 5538–5553 (2017)

[16] Ram, S.S., Veeravalli, V.V., Nedic, A.: Distributed non-autonomous power control through distributed convex optimization. In: IEEE INFOCOM 2009, pp. 3001–3005 (2009). IEEE

[17] Gan, L., Topcu, U., Low, S.H.: Optimal decentralized protocol for electric vehicle charging. IEEE Transactions on Power Systems **28**(2), 940–951 (2012)

[18] Nedić, A., Olshevsky, A., Rabbat, M.G.: Network topology and communication-computation tradeoffs in decentralized optimization. Proceedings of the IEEE **106**(5), 953–976 (2018) https://doi.org/10.1109/JPROC.2018.2817461

[19] Kovalev, D., Salim, A., Richtárik, P.: Optimal and practical algorithms for smooth and strongly convex decentralized optimization. Advances in Neural Information Processing Systems **33** (2020)

[20] Li, H., Lin, Z.: Accelerated gradient tracking over time-varying graphs for decentralized optimization. arXiv preprint arXiv:2104.02596 (2021)

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

[22] Kovalev, D., Shulgin, E., Richtárik, P., Rogozin, A.V., Gasnikov, A.: Adom: Accelerated decentralized optimization method for time-varying networks. In:

International Conference on Machine Learning, pp. 5784–5793 (2021). PMLR

[23] Kovalev, D., Beznosikov, A., Sadiev, A., Persiianov, M., Richtárik, P., Gasnikov, A.: Optimal algorithms for decentralized stochastic variational inequalities. Advances in Neural Information Processing Systems **35**, 31073–31088 (2022)

[24] Rogozin, A., Lukoshkin, V., Gasnikov, A., Kovalev, D., Shulgin, E.: Towards accelerated rates for distributed optimization over time-varying networks. In: International Conference on Optimization and Applications, pp. 258–272 (2021). Springer

[25] Li, H., Fang, C., Yin, W., Lin, Z.: Decentralized accelerated gradient methods with increasing penalty parameters. IEEE Transactions on Signal Processing **68**, 4855–4870 (2020)

[26] Zhang, C., Ahmad, M., Wang, Y.: Admm based privacy-preserving decentralized optimization. IEEE Transactions on Information Forensics and Security **14**(3), 565–580 (2019) https://doi.org/10.1109/TIFS.2018.2855169

[27] Li, S.E., Wang, Z., Zheng, Y., Sun, Q., Gao, J., Ma, F., Li, K.: Synchronous and asynchronous parallel computation for large-scale optimal control of connected vehicles. Transportation Research Part C: Emerging Technologies **121**, 102842 (2020) https://doi.org/10.1016/j.trc.2020.102842

[28] Kovalev, D., Gasanov, E., Gasnikov, A., Richtarik, P.: Lower bounds and optimal algorithms for smooth and strongly convex decentralized optimization over time-varying networks. Advances in Neural Information Processing Systems **34** (2021)

[29] Metelev, D., Rogozin, A., Kovalev, D., Gasnikov, A.: Is Consensus Acceleration Possible in Decentralized Optimization over Slowly Time-Varying Networks? (2023)

[30] Olfati-Saber, R., Fax, J.A., Murray, R.M.: Consensus and cooperation in networked multi-agent systems. Proceedings of the IEEE **95**(1), 215–233 (2007)

[31] Beznosikov, A., Samsonov, S., Sheshukova, M., Gasnikov, A., Naumov, A., Moulines, E.: First order methods with markovian noise: from acceleration to variational inequalities. arXiv preprint arXiv:2305.15938 (2023)

[32] Nesterov, Y.: Introductory Lectures on Convex Optimization: A Basic Course vol. 87. Springer (2003)

[33] Beznosikov, A., Samokhin, V., Gasnikov, A.: Distributed saddle-point problems: Lower bounds, optimal and robust algorithms. arXiv preprint arXiv:2010.13112 (2020)

[34] Rogozin, A., Bochko, M., Dvurechensky, P., Gasnikov, A., Lukoshkin, V.: An

accelerated method for decentralized distributed stochastic optimization over time-varying graphs. Conference on decision and control (2021)

[35] Beznosikov, A., Rogozin, A., Kovalev, D., Gasnikov, A.: Near-optimal decentralized algorithms for saddle point problems over time-varying networks. In: Optimization and Applications: 12th International Conference, OPTIMA 2021, Petrovac, Montenegro, September 27–October 1, 2021, Proceedings 12, pp. 246–257 (2021). Springer