На правах рукописи

Рогозин Александр Викторович

Децентрализованная оптимизация на меняющихся со временем сетях

Специальность:
1.2.2. Математическое моделирование, численные методы и комплексы программ

АВТОРЕФЕРАТ
диссертации на соискание ученой степени кандидата физико-математических наук

Moscow — 2023

Работа выполнена в федеральном государственном автономном образовательном учреждении высшего образования «Московский физико-технический институт (национальный исследовательский университет)».

Научный руководитель: доктор физико-математических наук, доцент
Гасников Александр Владимирович

Ведущая организация: Федеральное государственное бюджетное учреждение науки Институт проблем машиноведения Российской академии наук

Защита состоится **30 августа 2023 г. в 12 часов 00 минут** на заседании диссертационного совета **ФПМИ.1.2.2.020**, созданного на базе федерального государственного автономного образовательного учреждения высшего образования «Московский физико-технический институт (национальный исследовательский университет)» (МФТИ, Физтех).

по адресу: 141701, Московская область, г. Долгопрудный, Институтский переулок, д. 9.

С диссертацией можно ознакомиться в библиотеке МФТИ, Физтех и на сайте организации https://mipt.ru.

Автореферат разослан «___»_____ 2023

Ученый секретарь
диссертационного совета,
Кандидат технических наук, доцент Войтиков Константин Юрьевич

As a manuscript

Rogozin Alexander Viktorovich

Decentralized optimization over time-varying networks

Speciality:
1.2.2 Mathematical Modeling, Numerical Methods and Software Systems

ABSTRACT
of the dissertation for the Degree of
Candidate of Physical-Mathematical Sciences

Moscow — 2023

The dissertation was prepared in Moscow Physics and Technology Institute.

Scientific supervisor: Doctor of physical-mathematical sciences, associate professor
Gasnikov Alexander Vladimirovich

Leading organization: Federal State Budgetary Institution of Science Institute of Mechanical Engineering Problems of the Russian Academy of Sciences

The defense will take place on **30 August 2023 at 12 hours 00 minutes** at the meeting of the Dissertation Council **ФПМИ.1.2.2.020**, based at Moscow Physics and Technology Institute (MIPT, Phystech).

address: 141700, Moscow region, Dolgoprudny, Institutskiy pereulok, 9.

The text of the dissertation is available in the library MIPT, Phystech or on the website https://mipt.ru.

Abstract was sent «___»_____ 2023

Scientific secretary of the Dissertation Council,
Candidate of technical sciences, associate professor — Voitikov Konstantin Yurievich

General Description of the Subject of Work

Introduction

Decentralized optimization over time-varying networks has a wide range of applications in distributed learning, signal processing and various distributed control problems. The agents of the distributed system locally hold optimization objectives and can communicate to their immediate neighbors over a network. Moreover, the network may change from time to time due to technical malfunctions such as a loss of connection in wireless communications.

In decentralized optimization, the following sum-type optimization problems are considered.

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

Each node/agent locally holds f_i and can compute its values and gradients at given points. The agents can exchange information through a decentralized communication network, which may be time-varying. Each node is connected to several others via communication links and can communicate to them. A time-varying network is represented as a sequence of graphs (we focus on undirected graphs). Since there is no centralized aggregator (server node, master node), each agent locally holds a copy x_i of the decision vector x. The vectors held by the nodes should be synchronized, but the agents can communicate only to their immediate neighbors.

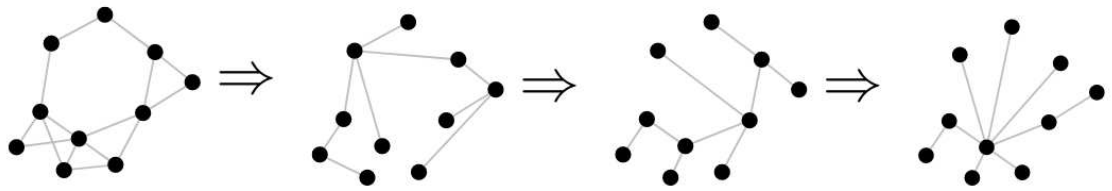

Figure 1: Time-varying network. The set of nodes stays the same, but edges change from time to time.

As an example, suppose that each agent has a set of observations and collaboratively the agents need to approximate a linear dependence of the observations. Let node i hold $\left\{a_{ij} \in \mathbb{R}^d\right\}_{j=1}^{n}$, $\left\{b_{ij} \in \mathbb{R}\right\}_{j=1}^{n}$ and denote $\mathbf{A}_i = [a_{i1} \dots a_{in}]^\top$, $\mathbf{b}_i = [b_{i1} \dots b_{in}]^\top$. Then each node holds a loss function $f_i(x) = 1/2 \left\|\mathbf{A}_i x - \mathbf{b}_i\right\|_2^2$. Note that in this example index i characterizes the part of the dataset.

Problems of type (1) arise in many applications where centralized aggregation is limited due to the structure of the network, privacy constraints or large amounts of data. Applications include vehicle coordination and control [42], distributed statistical inference and machine learning [39, 18, 31], power system control [40, 19], distributed averaging [11, 36, 52], distributed tracking [21], formation control [35, 41, 22, 5], distributed spectrum sensing [7],

distributed load balancing [4, 3]. See surveys [32, 29] for additional examples and [20, 15] for reviews of decentralized optimization over time-static graphs. Also see survey [45] for a review of decentralized methods for time-varying networks.

Each f_i is assumed to be convex and stored at a separate computational agent (or node). Moreover, each f_i is equipped with a first-order oracle (either stochastic or deterministic). That means that each computational node can compute gradients or stochastic gradients of the function it holds. During the computation process, the agents can exchange their decision vectors and gradients.

A decentralized optimization algorithm should be designed in such a way that the sum of functions is minimized while the decision vectors held at different computational nodes stay approximately the same. Assuming that node i holds x_i, the optimal point in the decentralized sense should be consensual and optimal, i.e.

$$x_1 = \ldots = x_m = x^* = \arg\min_{x \in \mathbb{R}^d} \frac{1}{m} \sum_{i=1}^{m} f_i(x).$$

Decentralized algorithms for time-varying graphs are built using several techniques. We can roughly point out three techniques. The first one is referred to as *gradient tracking* [30, 25, 49, 28]. In gradient tracking approach, an auxiliary variable is introduced in order to track the quantity that approximates the average gradient over the nodes in the network. We refer to the second technique as ADOM (after the method in which it was originally proposed) and it is presented in the works [27, 26]. ADOM uses a specific reformulation of the optimization problem, tackles decentralized communication as a compression operator and uses an error feedback mechanism. The third technique is consensus subroutine [23, 46, 44, 10] and it is the focus of the dissertation. This approach performs several consensus iterations after each oracle call. It is relatively simple in usage but produces suboptimal methods (up to a logarithmic term).

Required information

Goals of the work

The main goal of the work is to develop a technique that allows to obtain decentralized analogues of non-distributed optimization methods. Typically, developing a new decentralized optimization method can be viewed as some art, but we focus on a universal technique. The method is based on running a consensus algorithm for several iterations after each gradient step. The analysis is based on the inexact oracle concept. Moreover, an algorithm for distributed optimization with affine constraints is proposed. The goals of the thesis are summarized as follows:

- Develop a decentralized accelerated deterministic method for optimization over time-varying graphs.
- Develop a decentralized accelerated stochastic method with mini-batching for optimization over time-varying graphs.

- Develop a decentralized accelerated method for composite optimization with proximal-friendly term over time-varying graphs.

Scientific novelty

We underline the three main points of the thesis:

- Near optimal algorithms for decentralized optimization over time-varying networks.
- Dependence on average constants of objective functions instead of worst case constants.
- Accelerated algorithm for decentralized composite optimization.

Every point is novel and below we discuss the points in detail.

We propose the first near optimal algorithms for decentralized strongly-convex optimization with deterministic and stochastic primal oracle in the time-varying setup. Prior to our results, the optimal methods were proposed only using the dual oracle [48] and for time-static graphs [23]. Usage of the dual oracle assumes that the objective functions are dual friendly (i.e. admit the computation of Fenchel conjugate and its gradient either in a closed form of via a cheap numerical procedure), which may not always hold. Our approach is based on a consensus subroutine that allows to obtain a gradient of the objective with a given accuracy. After that, we analyze the methods with inexact gradient computation. Our approach works both for time-static and time-varying graphs.

An interesting and valuable outcome of our approach is the dependence on average constants instead of worst case ones. Let us describe what we mean by average and worst-case. Let each function f_i be L_i-smooth and μ_i-strongly convex. By average constants we understand $L_g = 1/m \sum_{i=1}^m L_i,\ \mu_g = 1/m \sum_{i=1}^m \mu_i$ and by worst-case we mean $L_\ell = \max_{i=1,\dots,m} L_i,\ \mu_\ell = \min_{i=1,\dots,m} \mu_i$. Note that it is possible that $L_g/\mu_g \gg L_\ell/\mu_\ell$. Our analysis yields the dependence on the average constants, which may be significantly better in some scenarios.

Finally, we develop an accelerated algorithm for decentralized composite optimization. Each of the objective functions held at the nodes is a sum of a smooth term and a possibly non-smooth composite term. The composite term is assumed to be proximal-friendly, its proximal operator can be easily computed. Our approach allows to work with constrained sets, while previous works [53] only support unconstrained minimization.

Theoretical and practical value

The theoretical value of the thesis is a relatively simple technique of converting a non-distributed optimization method into a decentralized method. The proposed approach builds an analogue of centralized optimization method with inexact gradients by running consensus algorithm for a sufficient number of iterations to obtain an approximate projection onto the consensual set. The usage of consensus subroutine results in a complexity optimal up to an additional logarithmic factor. We apply the technique to obtain near-optimal algorithms in deterministic and stochastic setups. Moreover, the complexity of our methods depends on average constants of smoothness and strong convexity instead of worst-case ones.

From theoretical perspective, we enhance the dependence on the parameters of optimization objectives (which may be significant) at the cost of an additional logarithmic factor (that is usually treated as not so significant).

From practical point of view, decentralized optimization has numerous applications in distributed signal processing and sensing, power system control, control of satellite and drone networks, coordination of automated vehicle groups and distributed statistical inference and machine learning. The loss of connection between computational agents may happen, and therefore the development of fault-tolerant algorithms has a large practical impact. We once again note that the proposed technique is simple enough to be implemented. Moreover, the potential user of the proposed algorithms is not obliged to set the number of communication rounds after each optimization step according to the theory; instead, one may regulate the number of consensus steps according to practical purposes.

Statements to be defended

We propose the following statements for the defense.

- A near-optimal method for decentralized optimization over time-varying networks was developed.
- A near-optimal method for decentralized stochastic optimization over time-varying network was proposed.
- A near-optimal method for decentralized composite optimization over time-varying graphs was developed.

We note that in both deterministic and stochastic cases the algorithms complexity depends on average smoothness and strong convexity constants instead of worst-case ones.

Presentations and validation of research results

The results were presented at the following conferences:

1. Towards accelerated rates for distributed optimization over time-varying networks. XII International Conference on Optimization and Applications. Petrovac, Montenegro, September 27, 2021 (online).
2. An Accelerated Method for Decentralized Distributed Stochastic Optimization Over Time-Varying Graphs. 2021 60th IEEE Conference on Decision and Control (CDC). Austin, USA, December 15, 2021 (online).
3. Decentralized Strongly-Convex Optimization with Affine Constraints: Primal and Dual Approaches. XIII International Conference on Optimization and Applications. Petrovac, Montenegro, September 26, 2022 (online).

Publications

The results in the thesis are presented in the following papers.
Published papers.

1. Alexander Rogozin, Vladislav Lukoshkin, Alexander Gasnikov, Dmitry Kovalev, Egor Shulgin (2021, September). *Towards accelerated rates for distributed optimization over time-varying networks*. In International Conference on Optimization and Applications (pp. 258-272). Springer, Cham.

2. Alexander Rogozin, Mikhail Bochko, Pavel Dvurechensky, Alexander Gasnikov, Vladislav Lukoshkin (2021, December). *An accelerated method for decentralized distributed stochastic optimization over time-varying graphs*. In 2021 60th IEEE Conference on Decision and Control (CDC) (pp. 3367-3373). IEEE.

3. Rogozin, A., Yarmoshik, D., Kopylova, K., and Gasnikov, A. (2023, January). *Decentralized Strongly-Convex Optimization with Affine Constraints: Primal and Dual Approaches*. In Advances in Optimization and Applications: 13th International Conference, OPTIMA 2022, Petrovac, Montenegro, September 26–30, 2022, Revised Selected Papers (pp. 93-105). Cham: Springer Nature Switzerland.

Personal contribution

[46], The author maid all theoretical analysis of the paper and most of the writing work. The author also provided base library functions for numerical experiments.

[44], The author provided the proof of the main result (Theorem 4.2 and Lemmas 4.3–4.6), carried numerical experiments and did the writing for presentation of main theoretical and numerical results.

[47], The author proved the results in Sections 4 and 5 and did the writing for Sections 1-5.

Structure of the thesis

The thesis consists of an introduction, three main chapters, list of references and two chapters in the Appendix.

The content of the work

Time-varying consensus

Let us first describe the process of reaching a consensus over time-varying networks.

The nodes communicate through a time-varying communication network (we focus on *undirected* graphs). The network is represented as a sequence of graphs $\{\mathcal{G}^k = (\mathcal{V}, \mathcal{E}^k)\}_{k=0}^{\infty}$. Sequence $\{\mathcal{G}^k\}_{k=0}^{\infty}$ is also referred to as a *time-varying graph*. The problems of reaching a consensus over time-varying graphs has been studied since 1980's (see i.e. seminal works [51] and [8]). More recent works include [43, 42, 38, 34, 24].

In order to maintain consensus constraints $x_1 = \ldots = x_m$, a sequence of communication matrices is assigned to the time-varying graph. The two most used types of communication matrices are *mixing matrices* and *gossip matrices*. Typically, mixing matrices are utilized in methods that use primal oracle and gossip matrices are employed in dual algorithms. Both types of matrices are defined in such way that a matrix-vector multiplication corresponds to one synchronized communication round.

Mixing matrix

Assumption 1. *Mixing matrix sequence $\{W^k\}_{k=0}^{\infty}$ satisfies the following properties.*

(Decentralized property) $[W^k]_{ij} = 0$ if $(i,j) \notin \mathcal{E}^k$ and $i \neq j$.

(Double stochasticity) $W^k\mathbf{1} = \mathbf{1}$ and $\mathbf{1}^\top W^k = \mathbf{1}^\top$.

(Spectrum property) There exists a positive $\tau \in \mathbb{Z}$ and $\chi > 0$ such that for any $k \geq \tau - 1$ and any $x \in \mathbb{R}^d$ it holds

$$\left\| \left(W_\tau^k - \frac{1}{m}\mathbf{1}\mathbf{1}^\top \right) x \right\|^2 \leq \left(1 - \frac{\tau}{\chi}\right) \|x\|^2,$$

where $W_\tau^k = W^k \ldots W^{k-\tau+1}$.

Also for each k introduce $\mathbf{W}^k = W^k \otimes \mathbf{I}$. The spectrum property in Assumption 1 ensures geometric convergence of consensus iterates $\mathbf{x}^{k+1} = \mathbf{W}^k\mathbf{x}^k$ to consensual point $\overline{\mathbf{x}}^0 = 1/m \sum_{i=1}^m [\mathbf{x}^0]_i$. After $N = O\left(\chi \log\left(\frac{1}{\varepsilon}\right)\right)$ iterations it holds $\left\|\mathbf{x}^N - \overline{\mathbf{x}}^0\right\|^2 \leq \varepsilon$. Here and below the dependence from different parameters except ε under $\log(\,\cdot\,)$ is skipped.

Typically mixing matrices are used in primal algorithms.

Remark 2 (Sufficient conditions for Assumption 1)**.** *Paper [30] gives sufficient conditions for Assumption 1 to hold. Firstly, the graph sequence $\{\mathcal{G}^k\}_{k=0}^{\infty}$ should be τ-connected. That means that for any $k \geq 0$ the union of τ consequent graphs $\widehat{\mathcal{G}}^k = \{\mathcal{V}, \bigcup_{i=k}^{k+\tau-1} \mathcal{E}^i\}$ is connected. Secondly, the following restrictions on the mixing matrix weights are imposed:*

1. *(Double stochasticity) $W^k\mathbf{1} = \mathbf{1}$, $\mathbf{1}^\top W^k = \mathbf{1}^\top$.*
2. *(Positive diagonal) $[W^k]_{ii} > 0$ for $i = 1, \ldots, m$.*
3. *(Edge utilization) If $(i,j) \in \mathcal{E}^k$, then $[W^k]_{ij} > 0$, else $[W^k]_{ij} = 0$.*
4. *(Nonvanishing weights) There exists $\theta > 0$ such that if $[W^k]_{ij} > 0$, then $[W^k]_{ij} \geq \theta$.*

In other words, the term τ in the Spectrum property of Assumption 1 describes the number of iterations such that the union of τ consequent graphs is connected.

A possible way to build mixing matrices satisfying Assumption 1 is to use Metropolis weights [30]:

$$[W^k]_{ij} = \begin{cases} \frac{1}{\max(\deg(i), \deg(j)) + 1} & \text{if } (i,j) \in \mathcal{E}^k, \\ 0 & \text{if } (i,j) \notin \mathcal{E}^k \text{ and } i \neq j, \\ 1 - \sum_{j \neq i} [W^k]_{ij}, & i = j \end{cases}$$

Gossip matrix

Dual methods typically use a notation of a *gossip matrix* sequence $\{\mathcal{L}^k\}_{k=0}^{\infty}$.

Assumption 3. *Gossip matrix sequence $\{\mathcal{L}^k\}_{k=0}^{\infty}$ satisfies the following properties.*

1. $[\mathcal{L}^k]_{ij} = 0$ *if* $i \neq j$ *and* $(i,j) \notin \mathcal{E}^k$.
2. $\operatorname{Ker} \mathcal{L}^k \supseteq \operatorname{span}(\mathbf{1})$.
3. $\operatorname{Im} \mathcal{L}^k \subseteq \{x \in \mathbb{R}^m : \ x_1 + \ldots + x_m = 0\}$.
4. *There exists a positive* $\tau \in \mathbb{Z}$ *and* $\chi > 0$ *such that for any* $k \geq \tau - 1$ *it holds*

$$\left\|\mathcal{L}_\tau^k x - x\right\|^2 \leq \left(1 - \frac{\tau}{\chi}\right) \|x\|^2$$

for all $x \in \mathbb{R}^m$ *such that* $x_1 + \ldots + x_m = 0$. *Here* $\mathcal{L}_\tau^k$ *is defined as*

$$\mathbf{I} - \mathcal{L}_\tau^k = (\mathbf{I} - \mathcal{L}^k) \ldots (\mathbf{I} - \mathcal{L}^{k-\tau+1}).$$

For time-static networks, let $\mathcal{L}$ denote the gossip matrix. Consensus constraints $x_1 = \ldots = x_m$ can be written as $\mathcal{L}\mathbf{x} = 0$. Therefore, distributed optimization is formulated as an affinely constrained problem. The dual approach build upon optimization of the function dual to F s.t. $\mathcal{L}\mathbf{x} = 0$.

A possible way to obtain a gossip matrix is $\mathcal{L}^k = \mathbf{I} - W^k$. As noted in [26], mixing matrix sequence $\{W^k\}_{k=0}^{\infty}$ satisfies Assumption 1 if and only if gossip matrix sequence $\{\mathcal{L}^k = \mathbf{I} - W^k\}_{k=0}^{\infty}$ satisfies Assumption 3.

Alternatively, a gossip matrix can be built using a graph Laplacian. Matrix $\mathbf{L}(\mathcal{G}^k)$ is called a Laplacian of $\mathcal{G}^k$ if

$$\mathbf{L}(\mathcal{G}^k) = \begin{cases} \deg(i), & i = j, \\ -1, & (i,j) \in \mathcal{E}^k, \\ 0, & (i,j) \notin \mathcal{E}^k \text{ and } i \neq j. \end{cases}$$

Then $\left\{\mathcal{L}^k = \mathbf{L}(\mathcal{G}^k)/\lambda_{\max}(\mathbf{L}^k)\right\}_{k=0}^{\infty}$ satisfies Assumption 3. Moreover, in the case $\tau = 1$ one can equivalently define χ as

$$\chi = \sup_{k \geq 0} \frac{\lambda_{\max}(\mathbf{L}(\mathcal{G}^k))}{\lambda_{\min}^+(\mathbf{L}(\mathcal{G}^k))}.$$

Reaching the consensus over time-varying networks can be viewed as a quadratic optimization problem with a time-varying objective function $\{c^k(\mathbf{x}) = \frac{1}{2}\mathbf{x}^\top(\mathbf{I} - \mathbf{W}^k)\mathbf{x}\}_{k=0}^{\infty}$. All functions $c^k(\mathbf{x})$ have the same minimizer $\overline{\mathbf{x}}^0$. Non-accelerated gradient descent corresponds to a consensus algorithm. On each iteration of gradient descent, there is a contraction of potential function $\Phi^k = \frac{1}{2}\left\|\mathbf{x}^k - \overline{\mathbf{x}}^0\right\|^2$. This contraction is robust to graph changes, which makes non-accelerated consensus converge over time-varying graphs.

On the contrary, accelerated gradient methods build upon a potential function of type $\Phi^k = a_k(c(\mathbf{y}^k) - c^*) + b_k\left\|\mathbf{z}^k - \overline{\mathbf{x}}^0\right\|^2$ [6], where $\mathbf{y}^k, \mathbf{z}^k$ are additional extrapolation sequences

and $a_k, b_k > 0$ are scalars (note that in the case of consensus problems optimal value $c^* = 0$). The first summand in Φ^k contains the function value and therefore is not robust to network changes. This observation illustrates that accelerated consensus is not reachable over time-varying networks. This fact was proved in [26] by building a lower complexity bound.

Remark 4. *Over time-static networks, accelerated consensus is possible. One can apply an accelerated gradient method to minimizing* $c(\mathbf{x}) = \frac{1}{2}\mathbf{x}^\top(\mathbf{I} - \mathbf{W})\mathbf{x}$ *(see Section 2.1 of [20]). An alternative way is to replace* $\mathbf{W}$ *by a Chebyshev polynomial* $P(\mathbf{W})$ *as shown in [48].*

Consensus subroutine

Introduce $\mathbf{x} = \text{col}[x_1, \ldots, x_m] = (x_1^\top \ldots x_m^\top)^\top \in \mathbb{R}^{md}$ and let $\mathcal{L} = \{\mathbf{x} \in \mathbb{R}^{md} : \ x_1 = \ldots = x_m\}$ denote the consensus set. Note that projection of $\mathbf{x}$ on $\mathcal{L}$ is $\mathbf{P}\mathbf{x}$.

Algorithm 1 Consensus

Require: Initial guess $\mathbf{x}^0$, number of iterations T.
  for $t = 0, \ldots, T-1$ **do**
    $\mathbf{x}^{t+1} = \mathbf{W}^t\mathbf{x}^t$
  end for

Spectral properties of mixing matrices give the following simple convergence result.

Lemma 5. *Algorithm 1 requires* $T = O(\chi \log(1/\varepsilon))$ *communication rounds to yield* $\mathbf{x}^T$ *such that* $\left\|\mathbf{x}^T - \mathbf{P}\mathbf{x}^0\right\|_2 \leq \varepsilon$.

Proof. Let us put $T = n\tau$. It holds $\mathbf{x}^T = \mathbf{W}_{n\tau}^{n\tau-1}\mathbf{x}^0$. Note that for all $t \geq 0$ we have $\mathbf{P}\mathbf{W}^t = \mathbf{P}$ and therefore $\mathbf{P}\mathbf{x}^T = \mathbf{x}^0$. Let us put $T = n\tau$, where n is an integer and $n\tau \geq \chi \log\left(\left\|\mathbf{x}^0 - \mathbf{P}\mathbf{x}^0\right\|/\varepsilon\right)$. We have

$$
\begin{aligned}
\|\mathbf{W}_{n\tau}^{n\tau-1}\mathbf{x}^0 - \mathbf{P}\mathbf{x}^0\|_2 &= \left\|(\mathbf{W}_{n\tau}^{n\tau-1} - \mathbf{P})(\mathbf{x}^0 - \mathbf{P}\mathbf{x}^0)\right\| \\
&= \left\|(\mathbf{W}_{\tau}^{n\tau-1} - \mathbf{P})(\mathbf{W}_{\tau}^{n\tau-\tau-1} - \mathbf{P})\ldots(\mathbf{W}_{\tau}^{\tau-1} - \mathbf{P})(\mathbf{x}^0 - \mathbf{P}\mathbf{x}^0)\right\|_2 \\
&\leq \left(1 - \frac{\tau}{\chi}\right)^n \left\|\mathbf{x}^0 - \mathbf{P}\mathbf{x}^0\right\|_2 = \exp\left(n\log\left(1 - \frac{\tau}{\chi}\right)\right)\left\|\mathbf{x}^0 - \mathbf{P}\mathbf{x}^0\right\|_2 \\
&\leq \exp\left(-\frac{n\tau}{\chi}\right)\left\|\mathbf{x}^0 - \mathbf{P}\mathbf{x}^0\right\|_2 \leq \varepsilon.
\end{aligned}
$$

□

Lemma 5 illustrates that complexity of reaching a consensus with accuracy ε is $O(\chi \log(1/\varepsilon))$. As we will se in the following sections, this fact determines the communication complexity of decentralized algorithms.

Consensus subroutine approach

Inexact oracle

Consider function $h(x),\ x \in Q$ that we would like to minimize with an iterative method. It is possible that exact gradient of h is not known. Instead we have an access to approximate function characteristics.

Definition 6. *We call $(h_\delta(x), \psi_\delta(y,x))$ a (δ, L, μ) model of function h at point $x \in Q$ if for any $y \in Q$ it holds*

$$\frac{\mu}{2}\|y-x\|_2^2 \le h(y) - (h_\delta(x) + \psi_\delta(y,x)) \le \frac{L}{2}\|y-x\|_2^2 + \delta.$$

To illustrate the application of inexact oracle technique for design of decentralized methods, consider a setting when the gradient is computed at shifted points. Consider minimization problem

$$\min_{x\in Q}\ h(x) = f(x) + g(x),$$

where $f(x)$ is a convex smooth function, $g(x)$ is a proper convex closed function and $Q \subseteq \mathbb{R}^d$ is a closed convex set. Assume that f is L_f-smooth and μ_f-strongly convex, i.e. for any $x, y \in Q$ it holds

$$\frac{\mu_f}{2}\|y-x\|_2^2 \le f(y) - f(x) - \langle \nabla f(x), y-x\rangle \le \frac{L_f}{2}\|y-x\|_2^2.$$

For each point $y \in Q$, we do not have an opportunity to compute $f(y)$, $g(y)$ and $\nabla f(y)$ exactly. Instead, we have an access to a first-order oracle at point $x \in Q$ that lies in some neighborhood of y. This is known as "computation as shifted points" and is described in [13]. The inexact model is

$$\begin{aligned} h_\delta(y) &= f(x) + g(x) + \langle \nabla f(x), y-x\rangle - \frac{\mu_f}{2}\|x-y\|_2^2, \\ \psi_\delta(z,y) &= \langle \nabla f(x), z-y\rangle + g(z) - g(x), \end{aligned}$$

where $\delta = \left(L_f + \frac{\mu_f}{2}\right)\|z-y\|_2^2$.

Using the fact that $\|y-z\|_2^2 \le 2\|x-y\|_2^2 + 2\|x-z\|_2^2$. For strong convexity relation, we have

$$\begin{aligned} f(z) + g(z) &\ge f(x) + g(z) + \langle \nabla f(x), z-x\rangle + \frac{\mu_f}{2}\|z-x\|_2^2 \\ &\ge f(x) + g(z) + \langle \nabla f(x), z-x\rangle + \frac{\mu_f}{4}\|z-y\|_2^2 - \frac{\mu_f}{2}\|x-y\|_2^2 \\ &= \left(f(x) + g(x) + \langle \nabla f(x), y-x\rangle - \frac{\mu_f}{2}\|x-y\|_2^2\right) \\ &\quad + (\langle \nabla f(x), z-y\rangle + g(z) - g(x)) + \frac{\mu_f}{4}\|z-y\|_2^2. \end{aligned}$$

Using relation $\|x-z\|_2^2 \leq 2\|y-z\|_2^2 + 2\|x-y\|_2^2$, for smoothness relation we obtain

$$\begin{aligned} f(z)+g(z) &\leq f(x)+g(z)+\langle \nabla f(x), z-x\rangle + \frac{L_f}{2}\|z-x\|_2^2 \\ &\leq f(x)+g(z)+\langle \nabla f(x), z-x\rangle + L_f\|y-x\|_2^2 + L_f\|z-y\|_2^2 \\ &\leq \left(f(x)+g(x)+\langle \nabla f(x), y-x\rangle - \frac{\mu_f}{2}\|x-y\|_2^2\right) \\ &\quad + (\langle \nabla f(x), z-y\rangle + g(z) - g(x)) + L_f\|y-x\|_2^2 \\ &\quad + \left(L_f + \frac{\mu_f}{2}\right)\|z-y\|_2^2. \end{aligned}$$

As a result, we obtain that

$$\frac{\mu_f}{4}\|z-y\|_2^2 \leq h(z) - h_\delta(y) - \psi_\delta(z,y) \leq L_f\|z-y\|_2^2 + \delta.$$

That means that $(h_\delta(y), \psi_\delta(z,y))$ is a $(\delta, 2L_f, \mu_f/2)$-model of $h(x)$.

Similar Triangles Method

We propose accelerated gradient methods for several scenarios in decentralized optimization. Initially the ideas of accelerating gradient methods via momentum have been proposed by Polyak in heavy ball method [37]. An optimal acceleration was developed in 1980-s in a seminal work by Nesterov [33]. After that, more approaches to fast gradient algorithms were proposed, including linear coupling [2], proximal point method interpretation [1]. For an overview of acceleration techniques see [17, 12].

In our work we choose a variant of acceleration called Similar Triangles Method (STM) [16]. We use STM because it supports stochasticity and inexactness.

Algorithm 2 Similar Triangles Method

Require: x_0 is the starting point, $\mu \geq 0$, $\{\delta_k\}_{k=0}^{\infty}$, $L > 0$

Set $y_0 = x_0$, $u_0 = x_0$, $\alpha_0 = 0$, $A_0 = \alpha_0$

for $k \geq 0$ **do**

  Find α_{k+1} as the largest root of $(A_k + \alpha_{k+1})(1 + A_k\mu) = L\alpha_{k+1}^2$ and put $A_{k+1} = A_k + \alpha_{k+1}$.

  $y_{k+1} = \frac{\alpha_{k+1}u_k + A_k x_k}{A_{k+1}}$

  Define

$$\phi_{k+1}(x) = \alpha_{k+1}\psi_{\delta_k}(x, y_{k+1}) + \frac{1+A_k\mu}{2}\|x-u_k\|_2^2 + \frac{\alpha_{k+1}\mu}{2}\|x-y_{k+1}\|_2^2$$

  $u_{k+1} = \arg\min_{x\in Q} \phi_{k+1}(x)$

  $x_{k+1} = \frac{\alpha_{k+1}u_{k+1} + A_k x_k}{A_{k+1}}$

end for

STM has additional features that we do not use in the work. Firstly, it allows to choose a step-size according to a line search rule. However, line search procedure is not known to

be implemented in decentralized optimization algorithms, and therefore we do not use this feature of STM. Secondly, Line 6 of STM admits an approximate computation of $\arg\min$, but this feature is not used in our analysis. Thirdly, squared Euclidean norm in the definition of $\phi_{k+1}(x)$ in 5 can be replaced by a Bregman divergence $V(y, x)$. Our work does not cover non-Euclidean proximal setup, so this opportunity of STM is not used, as well.

Theorem 7. *After N iterations Algorithm 2 yields x_N such that*

$$f(x_N) - f^* \leq \frac{\|u_0 - x^*\|_2^2}{2A_N} + \frac{2\sum_{k=0}^{N-1} A_{k+1}\delta_k}{A_N}$$
$$\|u_N - x_*\|_2^2 \leq \frac{\|u_0 - x_*\|_2^2}{1 + A_N\mu} + \frac{4\sum_{k=0}^{N-1} A_{k+1}\delta_k}{1 + A_N\mu}.$$

Moreover, for constant error $\delta_k \equiv \delta$ it holds

$$\frac{\sum_{k=0}^{N-1} A_{k+1}\delta}{A_N} \leq \left(1 + \sqrt{\frac{L}{\mu}}\right)\delta.$$

This fact is used in our analysis to ensure that accumulated inexactness does not exceed $O(\varepsilon)$, where ε denotes the desired accuracy.

Inexactness for decentralized optimization with consensus subroutine

The idea of consensus subroutine is to approximately average the values held by the nodes in the network and make an iteration of some gradient method afterwards. Intuitively, if the iterates held by the nodes are approximately equal, then the inexact model based on this iterates will be a good approximation of the function. In other words, if the point of gradient computation is close to some consensual point, we have the case of gradient computation at shifted points. This case was described in Section **Inexact oracle** above.

Let us describe how to apply the approach to decentralized optimization in detail. We consider the most general case of composite optimization.

$$\min_{x \in Q} \; h(x) := \frac{1}{m}\sum_{i=1}^{m} f_i(x) + g(x) \tag{2}$$

where f_i are convex smooth functions and $g(x)$ is a proper closed convex function with easily computable proximal operator.

Assumption 8. *For each $i = 1, \ldots, m$ function f_i is L_i-smooth and μ_i-strongly convex, i.e. for any $x, y \in \mathbb{R}^d$ it holds*

$$f_i(y) \leq f_i(x) + \langle \nabla f_i(x), y - x\rangle + \frac{L_i}{2}\|y - x\|_2^2,$$
$$f_i(y) \geq f_i(x) + \langle \nabla f_i(x), y - x\rangle + \frac{\mu_i}{2}\|y - x\|_2^2.$$

Also introduce local and global constants characterizing problem optimization parameters.

$$L_\ell = \max_{i=1,\dots,m} L_i,\ \mu_\ell = \min_{i=1,\dots,m} \mu_i, \tag{3a}$$

$$L_g = \frac{1}{m}\sum_{i=1}^m L_i,\ \mu_g = \frac{1}{m}\sum_{i=1}^m \mu_i. \tag{3b}$$

In the literature, it has been shown that global and local constants may significantly differ [48, 45].

Lemma 9. *Consider* $y \in \mathbb{R}^d,\ z \in \mathbb{R}^d,\ \mathbf{x} = [x_1^\top, \dots, x_m^\top]^\top \in \mathbb{R}^{md}$. *Define*

$$\eta = \frac{1}{2m}\left(\frac{L_\ell^2}{L_g} + \frac{2L_\ell^2}{\mu_g} + L_\ell - \mu_\ell\right), \tag{4}$$

$$\delta = \eta \sum_{i=1}^m \|x_i - y\|_2^2, \tag{5}$$

$$f_\delta(y, \mathbf{x}) = \frac{1}{m}\sum_{i=1}^m \left[f_i(x_i) + \langle \nabla f_i(x_i), y - x_i\rangle + \frac{1}{2}\left(\mu_\ell - \frac{2L_\ell^2}{\mu_g}\right)\|y - x_i\|^2\right],$$

$$\psi_\delta(z, y, \mathbf{x}) = \frac{1}{m}\sum_{i=1}^m \left[\langle \nabla f_i(x_i), z - y\rangle + g(z) - g_i(x_i)\right].$$

Then $(f_\delta(y, \mathbf{x}), \psi_\delta(z, y, \mathbf{x}))$ *is a* $(\delta, 2L_g, \mu_g/2)$*-model of* f *at point* y, *i.e.*

$$\frac{\mu_g}{4}\|z - y\|^2 \leqslant f(z) - f_\delta(y, \mathbf{x}) - \psi_\delta(z, y, \mathbf{x}) \leq L_g\|z - y\|^2 + \delta.$$

Lemma 9 shows that oracle inexactness measure δ is proportional to the mean squared distance from set of points $[x_1, \dots, x_m]$ where gradient is computed to consensual point y. Generally speaking, gradient methods with inexact oracle converge to a δ-neighborhood of solution [14, 50] if the inexactness is δ. Therefore, in order to reach accuracy ε, we need to put $\delta = O(\varepsilon)$, and therefore the number of communication rounds after each oracle calls will be $T = O(\tau\chi\log(1/\varepsilon))$. This illustrated how communication complexity of first-order methods with consensus subroutine is obtained.

Moreover, it is worth noting that in our approach smoothness and strong convexity constants of inexact oracle are described by global problem parameters L_g, μ_g instead of local ones L_ℓ, μ_ℓ. This is a novel feature of the approach.

Decentralized Methods with Consensus Subroutine

In this section, we show how to apply consensus subroutine to different classes of decentralized problem. Our results build upon applying variations of Lemma 9 to each particular case.

Algorithm for Deterministic Optimization

In this section we describe the results in paper [46]. Consider minimization problem

$$\min_{x\in\mathbb{R}^d} \sum_{i=1}^{m} f_i(x). \tag{6}$$

We assume that each f_i $(i = 1, \ldots, m)$ is μ_i-strongly convex and has a L_i-Lipschitz gradient, i.e. Assumption 8 holds. Moreover, f_i is equipped with a first-order deterministic oracle.

Introduce $\mathbf{x} = \mathrm{col}[x_1, \ldots, x_m] \in \mathbb{R}^{md}$ and $\mathcal{L} = \{\mathbf{x} \in \mathbb{R}^{md} : \ x_1 = \ldots = x_m\}$. Problem (6) can be equivalently rewritten as

$$\begin{aligned} &\min_{\mathbf{x}\in\mathbb{R}^{md}} \sum_{i=1}^{m} f_i(x_i), \\ &\text{s.t. } x_1 = \ldots = x_m. \end{aligned} \tag{7}$$

We adapt STM (Algorithm 2) for solving problem (7).

Algorithm 3 Accelerated decentralized method with consensus subroutine

Require: Initial guess $\mathbf{x}^0 \in \mathcal{L}$, constants $L, \mu > 0$, $\mathbf{u}^0 = \mathbf{x}^0$, $\alpha^0 = A^0 = 0$

1: **for** $k = 0, 1, 2, \ldots$ **do**

2: Find α^{k+1} as the greater root of $(A^k + \alpha^{k+1})(1 + A^k\mu_g/2) = 2L_g(\alpha^{k+1})^2$

3: $A^{k+1} = A^k + \alpha^{k+1}$

4: $\mathbf{y}^{k+1} = \dfrac{\alpha^{k+1}\mathbf{u}^k + A^k\mathbf{x}^k}{A^{k+1}}$

5: $\mathbf{v}^{k+1} = \dfrac{\alpha^{k+1}(\mu_g/2)\mathbf{y}^{k+1} + (1 + A^k\mu_g/2)\mathbf{u}^k}{1 + A^{k+1}\mu_g/2} - \dfrac{\alpha^{k+1}\nabla F(\mathbf{y}^{k+1})}{1 + A^{k+1}\mu_g/2}$

6: $\mathbf{u}^{k+1} = \mathrm{Consensus}(\mathbf{v}^{k+1}, T^k)$

7: $\mathbf{x}^{k+1} = \dfrac{\alpha^{k+1}\mathbf{u}^{k+1} + A^k\mathbf{x}^k}{A^{k+1}}$

8: **end for**

Here Consensus in line 6 is Algorithm 1.

In fact, problem (6) is a specific case of problem (2) with $Q = \mathbb{R}^d$ composite term $g(x) \equiv 0$. Therefore, Lemma 9 is applicable. Here let us write the corresponding result.

Lemma 10. *Consider* $y \in \mathbb{R}^d,\ z \in \mathbb{R}^d,\ \mathbf{x} = [x_1^\top, \ldots, x_m^\top]^\top \in \mathbb{R}^{md}$. *Define*

$$\begin{aligned}
\eta &= \frac{1}{2m}\left(\frac{L_\ell^2}{L_g} + \frac{2L_\ell^2}{\mu_g} + L_\ell - \mu_\ell\right), \\
\delta &= \eta \sum_{i=1}^m \|x_i - y\|_2^2, \\
f_\delta(y, \mathbf{x}) &= \frac{1}{m}\sum_{i=1}^m \left[f_i(x_i) + \langle \nabla f_i(x_i), y - x_i\rangle + \frac{1}{2}\left(\mu_\ell - \frac{2L_\ell^2}{\mu_g}\right)\|y - x_i\|^2\right], \\
\psi_\delta(z, y, \mathbf{x}) &= \frac{1}{m}\sum_{i=1}^m \left[\langle \nabla f_i(x_i), z - y\rangle\right].
\end{aligned}$$

Then $(f_\delta(y, \mathbf{x}), \psi_\delta(z, y, \mathbf{x}))$ *is a* $(\delta, 2L_g, \mu_g/2)$*-model of* f *at point* y, *i.e.*

$$\frac{\mu_g}{4}\|z - y\|^2 \leqslant f(z) - f_\delta(y, \mathbf{x}) - \psi_\delta(z, y, \mathbf{x}) \leq L_g\|z - y\|^2 + \delta.$$

The complexity of the resulting method is derived from complexity of STM.

Theorem 11. *Choose some* $\varepsilon > 0$ *and set*

$$T_k = T = \frac{\chi}{2}\log\frac{D}{\delta'},\ \delta' = \frac{n\varepsilon}{32}\frac{\mu_g^{3/2}}{L_g^{1/2}L_\ell^2}$$

where $D = \left(\frac{D_1}{\sqrt{\varepsilon}} + D_2\right)^2$ *and*

$$\begin{aligned}
D_1 &= \frac{L_\ell}{L_g^{1/2}\mu_g}\left[8\sqrt{2}L_\ell\left\|\overline{u}^0 - x^*\right\|\left(\frac{L_g}{\mu_g}\right)^{3/4} + \frac{4\sqrt{2}\,\|\nabla F(\mathbf{X}^*)\|}{\sqrt{n}}\left(\frac{L_g}{\mu_g}\right)^{1/4}\right], \\
D_2 &= \frac{L_\ell}{L_g^{1/2}\mu_g}\left[3\sqrt{\mu_g} + 4\sqrt{2n}\left(\frac{L_g}{\mu_g}\right)^{1/4}\right].
\end{aligned}$$

Then Algorithm 3 requires

$$N_{comp} = 2\sqrt{\frac{L_g}{\mu_g}}\log\left(\frac{\|\overline{u}^0 - x^*\|^2}{2\varepsilon L_g}\right) \tag{9}$$

gradient computations at each node and

$$N_{comm} = N \cdot T = 2\sqrt{\frac{L_g}{\mu_g}}\chi \cdot \log\left(\frac{2L_g\|\overline{u}^0 - x^*\|^2}{\varepsilon}\right)\log\left(\frac{D_1}{\sqrt{\varepsilon}} + D_2\right)$$

communication steps to yield $\mathbf{x}^N$ *such that*

$$f(\overline{x}^N) - f(x^*) \leqslant \varepsilon,\ \left\|\mathbf{x}^N - \overline{\mathbf{X}}^N\right\|^2 \leqslant \delta'.$$

Algorithm for Stochastic Optimization

This section describes the results in paper [44]. Now the objective functions are equipped with a stochastic first-order oracle.

We consider the following optimization problem

$$\min_{x\in\mathbb{R}^d} \frac{1}{m}\sum_{i=1}^{m} f_i(x), \quad f_i(x) := \mathbb{E}_{\xi_i\sim\mathcal{D}_i}\mathbf{f}_i(x,\xi_i), \tag{10}$$

where ξ_i's are random variables with probability distributions $\mathcal{D}_i$. We make the following assumptions.

Assumption 12. *Almost sure w.r.t. distribution $\mathcal{D}_i$, the function $\mathbf{f}_i(x,\xi_i)$ has gradient $\nabla\mathbf{f}_i(x,\xi_i)$. Function $\mathbf{f}_i(x,\xi)$ is $L_i(\xi_i)$-smooth with respect to the Euclidean norm, i.e. for all $x,y\in\mathbb{R}^d$ we have*

$$\mathbf{f}_i(y,\xi_i) \le \mathbf{f}_i(x,\xi_i) + \langle\nabla\mathbf{f}_i(x,\xi_i), y-x\rangle + \frac{L_i(\xi_i)}{2}\|y-x\|_2^2.$$

For each $i=1,\dots,m$ there is a constant $L_i\ge 0$ such that $\sqrt{\mathbb{E}_{\xi_i}L_i^2(\xi_i)}\le L_i$.

Moreover, we assume that for each $i=1,\dots,m$ function f_i is μ_i-strongly convex, i.e.

$$f_i(y) \ge f_i(x) + \langle\nabla f_i(x), y-x\rangle + \frac{\mu_i}{2}\|y-x\|_2^2.$$

Under Assumption 12 we have that each f_i is L_i-smooth. We also assume bounded variance of the stochastic gradient.

Assumption 13. *For all $x\in\mathbb{R}^d$ and for each $i=1,\dots,m$ it holds*

$$\mathbb{E}_{\xi_i}[\|\nabla\mathbf{f}_i(x,\xi_i)-\nabla f_i(x)\|_2^2] \leqslant \sigma_i^2. \tag{11}$$

In the analysis we will use constants μ_ℓ,μ_g,L_g defined in (3). We will also need worst-case constants over stochastic realizations and aveage variance. Introduce

$$L_\xi = \max_{i=1,\dots,m}\max_{\xi} L_i(\xi),\ M_\xi = \max_{i=1,\dots,m}\max_{\xi}\|\nabla f_i(x^*,\xi)\|_2,\ \sigma_g^2 = \frac{1}{m}\sum_{i=1}^{m}\sigma_i^2.$$

We use the same inexact oracle construction as in Lemma 10. For brevity introduce inexact gradient

$$e_\delta(y,\mathbf{x}) = \frac{1}{m}\sum_{i=1}^{m}\nabla f_i(x).$$

Note that we have $\psi_\delta(z,y,\mathbf{x}) = \langle e(y,\mathbf{x}), z-y\rangle$. Also introduce stochastic batched inexact gradient

$$\tilde{e}_\delta(y,\mathbf{x}) = \frac{1}{m}\sum_{i=1}^{m}\frac{1}{r}\sum_{j=1}^{r}\nabla\mathbf{f}_i(x_i,\xi_i^j).$$

Lemma 14. *Stochastic gradient $\tilde{e}_\delta(y, \mathbf{x})$ satisfies*

$$\mathbb{E}\widetilde{e}_\delta(y, \mathbf{x}) = e_\delta(y, \mathbf{x}) \tag{12}$$

$$\mathbb{E}\|\widetilde{e}_\delta(y, \mathbf{x}) - e_\delta(y, \mathbf{x})\|^2 \leq \frac{\sum_{i=1}^n \sigma_i^2}{n^2 r} = \frac{\sigma_g^2}{nr}. \tag{13}$$

Algorithm 4 Decentralized Stochastic AGD

Require: Initial guess $\mathbf{x}^0 \in \mathcal{L}$, constants $L_g, \mu_g > 0$,
$\mathbf{u}^0 = \mathbf{x}^0$

1: **for** $k = 0, 1, 2, \ldots$ **do**
2: $\mathbf{y}^{k+1} = \dfrac{\alpha^{k+1}\mathbf{u}^k + A^k\mathbf{x}^k}{A^{k+1}}$
3: $\mathbf{v}^{k+1} = \dfrac{(\alpha^{k+1}\mu_g/2)\mathbf{y}^{k+1} + (1 + A^k\mu_g/2)\mathbf{u}^k}{1 + A^{k+1}\mu_g/2} - \dfrac{\alpha^{k+1}\nabla^r F(\mathbf{y}^{k+1})}{1 + A^{k+1}\mu_g/2}$
4: $\mathbf{u}^{k+1} = \text{Consensus}(\mathbf{v}^{k+1}, T^k)$
5: $\mathbf{X}^{k+1} = \dfrac{\alpha^{k+1}\mathbf{U}^{k+1} + A^k\mathbf{X}^k}{A^{k+1}}$
6: **end for**

Batched gradient allows to reduce the variance term in the complexity of algorithm. The convergence result builds upon complexity of STM.

Theorem 15 (Main result)**.** *Let $\varepsilon > 0$ be the desired accuracy. Set*

$$T_k = T = \frac{\chi}{2}\log\frac{D}{\delta'},\ \delta' = \frac{m\varepsilon}{32}\frac{\mu_g^{3/2}}{L_g^{1/2}L_\ell^2},\ r = \frac{2\sigma_g^2}{\varepsilon\sqrt{L_g\mu_g}},$$

where

$$\sqrt{D} = \left(\frac{2L_\ell}{\sqrt{L_g\mu_g}} + 1\right)\sqrt{\delta'} + \frac{2mM_\xi}{\sqrt{L_g\mu_g}} \tag{14}$$
$$+ \frac{2L_\ell}{\mu_g}\sqrt{m}\left(\left\|\overline{\mathbf{u}}^0 - \mathbf{x}^*\right\|^2 + \frac{2}{\sqrt{L_g\mu_g}}\left(\frac{\sigma_g^2}{4mL_g r^2} + \delta\right)\right)^{1/2}.$$

Then, to yield $\mathbf{x}^N$ such that

$$\mathbb{E}f\left(\frac{1}{m}\sum_{i=1}^m x_i^N\right) - f(x^*) \leqslant \varepsilon,\ \mathbb{E}\left\|\mathbf{x}^N - \mathbf{P}\mathbf{x}^N\right\|^2 \leqslant \delta' = O(\varepsilon),$$

Algorithm 3 requires no more than

$$N_{comp} = N \cdot r = \frac{6\sigma_g^2}{n\mu_g\varepsilon}\log\left(\frac{4L_g\left\|\overline{u}^0 - x^*\right\|^2}{\varepsilon}\right) \tag{15}$$

stochastic oracle calls at each node and no more than

$$N_{comm} = \frac{3}{2}\sqrt{\frac{L_g}{\mu_g}}\chi \cdot \log\left(\frac{4L_g \left\|\overline{u}^0 - x^*\right\|^2}{\varepsilon}\right) \log\frac{D}{\delta'},$$

communication rounds.

Algorithm for Proximal Optimization

In this section, we describe the result for (deterministic) problem of general form (1).

$$\min_{x \in Q} \; f(x) + g(x) = \frac{1}{m}\sum_{i=1}^{m} f_i(x) + g(x).$$

Introduce proximal operator of g w.r.t. set Q.

$$\text{prox}_g^{\gamma}(x) = \arg\min_{y \in Q}\left(g(y) + \frac{1}{2\gamma}\left\|y - x\right\|_2^2\right).$$

We assume that $g(x)$ is a proximal-friendly function, i.e. $\text{prox}_g^{\gamma}(x)$ can be computed easily.

Along with assumptions on smoothness and strong convexity of $f_i(x)$, we will need to bound the gradient of $f_i(x)$ uniformly over Q.

Assumption 16. *For each $i = 1, \ldots, m$ function f_i has a bounded gradient, i.e. there exists $M_i \geq 0$ such that*

$$\left\|\nabla f_i(x)\right\|_2 \leq M_i.$$

Algorithm 5 Accelerated decentralized proximal method with consensus subroutine

Require: Initial guess $\mathbf{x}^0 \in \mathcal{L}$, constants $L, \mu > 0$, $\mathbf{u}^0 = \mathbf{x}^0$, $\alpha^0 = A^0 = 0$

1: **for** $k = 0, 1, 2, \ldots$ **do**

2: Find α^{k+1} as the greater root of $(A^k + \alpha^{k+1})(1 + A^k\mu_g/2) = 2L_g(\alpha^{k+1})^2$

3: $A^{k+1} = A^k + \alpha^{k+1}$

4: $\mathbf{y}^{k+1} = \dfrac{\alpha^{k+1}\mathbf{u}^k + A^k\mathbf{x}^k}{A^{k+1}}$

5: $\mathbf{v}^{k+1} = \dfrac{\alpha^{k+1}(\mu_g/2)\mathbf{y}^{k+1} + (1 + A^k\mu_g/2)\mathbf{u}^k}{1 + A^{k+1}\mu_g/2} - \dfrac{\alpha^{k+1}\nabla F(\mathbf{y}^{k+1})}{1 + A^{k+1}\mu_g/2}$

6: $\mathbf{u}^{k+1} = \text{prox}_G^{\gamma_k}\left(\text{Consensus}(\mathbf{v}^{k+1}, T^k)\right)$

7: $\mathbf{x}^{k+1} = \dfrac{\alpha^{k+1}\mathbf{u}^{k+1} + A^k\mathbf{x}^k}{A^{k+1}}$

8: **end for**

We have the following convergence result.

Theorem 17. *Let Assumptions 8, 16 hold. Then accelerated method requires* $O\left(\sqrt{L_g/\mu_g}\log(1/\varepsilon)\right)$ *oracle calls per node and* $O\left(\sqrt{L_g/\mu_g}\chi\log^2(1/\varepsilon)\log\left(\log(1/\varepsilon)\right)\right)$ *communication rounds to reach* ε*-accuracy in function and consensus violation.*

The result on proximal method with bounded set Q is known for saddle-point problems [9]. Our work proposes analysis that does not require boundedness of Q at the cost of assuming bounded gradient.

References

[1] Kwangjun Ahn and Suvrit Sra. Understanding nesterov's acceleration via proximal point method. In *Symposium on Simplicity in Algorithms (SOSA)*, pages 117–130. SIAM, 2022.

[2] Zeyuan Allen-Zhu and Lorenzo Orecchia. Linear coupling: An ultimate unification of gradient and mirror descent. *arXiv:1407.1537*, 2014.

[3] Natalia Amelina, Alexander Fradkov, Yuming Jiang, and Dimitrios J Vergados. Approximate consensus in stochastic networks with application to load balancing. *IEEE Transactions on Information Theory*, 61(4):1739–1752, 2015.

[4] Natalia Amelina, Oleg Granichin, and Aleksandra Kornivetc. Local voting protocol in decentralized load balancing problem with switched topology, noise, and delays. In *52nd IEEE Conference on Decision and Control*, pages 4613–4618. IEEE, 2013.

[5] Cathrine Antal, Oleg Granichin, and Sergey Levi. Adaptive autonomous soaring of multiple uavs using simultaneous perturbation stochastic approximation. In *49th IEEE Conference on Decision and Control (CDC)*, pages 3656–3661. IEEE, 2010.

[6] Nikhil Bansal and Anupam Gupta. Potential-function proofs for gradient methods. *Theory of Computing*, 15(1):1–32, 2019.

[7] Juan Andrés Bazerque and Georgios B Giannakis. Distributed spectrum sensing for cognitive radio networks by exploiting sparsity. *IEEE Transactions on Signal Processing*, 58(3):1847–1862, 2009.

[8] Dimitri P Bertsekas and John N Tsitsiklis. *Parallel and distributed computation: numerical methods*, volume 23. Prentice hall Englewood Cliffs, NJ, 1989.

[9] Aleksandr Beznosikov, Valentin Samokhin, and Alexander Gasnikov. Distributed saddle-point problems: Lower bounds, optimal algorithms and federated gans. *arXiv preprint arXiv:2010.13112*, 2021.

[10] Alexander Beznosikov, Dmitry Kovalev, Abdurakhmon Sadiev, Peter Richtarik, and Alexander Gasnikov. Optimal distributed algorithms for stochastic variational inequalities. *arXiv preprint*, 2021.

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

[12] Bugra Can. *Theory and Methods for Stochastic, Accelerated, and Distributed Optimization*. PhD thesis, Rutgers The State University of New Jersey, Graduate School-Newark, 2022.

[13] O. Devolder, F. Glineur, and Yu. Nesterov. First-order methods with inexact oracle: the strongly convex case. *CORE Discussion Papers*, 2013016:47, 2013.

[14] O. Devolder, F. Glineur, and Yu. Nesterov. First-order methods of smooth convex optimization with inexact oracle. *Mathematical Programming*, 146(1-2):37–75, 2014.

[15] Darina Dvinskikh and Alexander Gasnikov. Decentralized and parallelized primal and dual accelerated methods for stochastic convex programming problems. *arXiv preprint arXiv:1904.09015*, 2019.

[16] Pavel Dvurechensky, Alexander Gasnikov, Sergey Omelchenko, and Alexander Tiurin. Adaptive similar triangles method: a stable alternative to sinkhorn's algorithm for regularized optimal transport. *arXiv:1706.07622*, 2017.

[17] Alexandre d'Aspremont, Damien Scieur, Adrien Taylor, et al. Acceleration methods. *Foundations and Trends® in Optimization*, 5(1-2):1–245, 2021.

[18] Pedro A Forero, Alfonso Cano, and Georgios B Giannakis. Consensus-based distributed support vector machines. *Journal of Machine Learning Research*, 11(5), 2010.

[19] Lingwen Gan, Ufuk Topcu, and Steven H Low. Optimal decentralized protocol for electric vehicle charging. *IEEE Transactions on Power Systems*, 28(2):940–951, 2012.

[20] Eduard Gorbunov, Alexander Rogozin, Aleksandr Beznosikov, Darina Dvinskikh, and Alexander Gasnikov. Recent theoretical advances in decentralized distributed convex optimization. In *High-Dimensional Optimization and Probability*, pages 253–325. Springer, 2022.

[21] Oleg Granichin and Natalia Amelina. Simultaneous perturbation stochastic approximation for tracking under unknown but bounded disturbances. *IEEE Transactions on Automatic Control*, 60(6):1653–1658, 2014.

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

[23] Dusan Jakovetić, Joao Xavier, and José MF Moura. Fast distributed gradient methods. *IEEE Transactions on Automatic Control*, 59(5):1131–1146, 2014.

[24] Solmaz S Kia, Bryan Van Scoy, Jorge Cortes, Randy A Freeman, Kevin M Lynch, and Sonia Martinez. Tutorial on dynamic average consensus: The problem, its applications, and the algorithms. *IEEE Control Systems Magazine*, 39(3):40–72, 2019.

[25] Júlia Komjáthy and Yuval Peres. Lecture notes for markov chains: Mixing times, hitting times, and cover times.

[26] Dmitry Kovalev, Elnur Gasanov, Alexander Gasnikov, and Peter Richtarik. Lower bounds and optimal algorithms for smooth and strongly convex decentralized optimization over time-varying networks. *Advances in Neural Information Processing Systems*, 34, 2021.

[27] Dmitry Kovalev, Egor Shulgin, Peter Richtárik, Alexander Rogozin, and Alexander Gasnikov. Adom: Accelerated decentralized optimization method for time-varying networks. *arXiv preprint arXiv:2102.09234*, 2021.

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

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

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

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

[32] Angelia Nedic and Asuman Ozdaglar. Cooperative distributed multi-agent optimization. *Convex optimization in signal processing and communications*, 340, 2010.

[33] Yurii Nesterov. A method of solving a convex programming problem with convergence rate $o(1/k^2)$. *Soviet Mathematics Doklady*, 27(2):372–376, 1983.

[34] Reza Olfati-Saber and Richard M Murray. Consensus problems in networks of agents with switching topology and time-delays. *IEEE Transactions on automatic control*, 49(9):1520–1533, 2004.

[35] Alex Olshevsky. Efficient information aggregation strategies for distributed control and signal processing. *arXiv preprint arXiv:1009.6036*, 2010.

[36] Alex Olshevsky. Linear time average consensus on fixed graphs and implications for decentralized optimization and multi-agent control. *arXiv preprint arXiv:1411.4186*, 2014.

[37] Boris T Polyak. Some methods of speeding up the convergence of iteration methods. *Ussr computational mathematics and mathematical physics*, 4(5):1–17, 1964.

[38] Anton V Proskurnikov and Giuseppe Carlo Calafiore. Delay robustness of consensus algorithms: Beyond the uniform connectivity (extended version). *arXiv preprint arXiv:2105.07183*, 2021.

[39] Michael Rabbat and Robert Nowak. Distributed optimization in sensor networks. In *Proceedings of the 3rd international symposium on Information processing in sensor networks*, pages 20–27, 2004.

[40] Sundhar Srinivasan Ram, Venugopal V Veeravalli, and Angelia Nedic. Distributed non-autonomous power control through distributed convex optimization. In *IEEE INFOCOM 2009*, pages 3001–3005. IEEE, 2009.

[41] Wei Ren. Consensus based formation control strategies for multi-vehicle systems. In *2006 American Control Conference*, pages 6–pp. IEEE, 2006.

[42] Wei Ren and Randal W Beard. *Distributed consensus in multi-vehicle cooperative control*, volume 27. Springer, 2008.

[43] Wei Ren and Yongcan Cao. *Distributed coordination of multi-agent networks: emergent problems, models, and issues*, volume 1. Springer, 2011.

[44] Alexander Rogozin, Mikhail Bochko, Pavel Dvurechensky, Alexander Gasnikov, and Vladislav Lukoshkin. An accelerated method for decentralized distributed stochastic optimization over time-varying graphs. *arXiv:2103.15598*, 2021.

[45] Alexander Rogozin, Alexander Gasnikov, Aleksander Beznosikov, and Dmitry Kovalev. Decentralized optimization over time-varying graphs: a survey. *arXiv preprint arXiv:2210.09719*, 2022.

[46] Alexander Rogozin, Vladislav Lukoshkin, Alexander Gasnikov, Dmitry Kovalev, and Egor Shulgin. Towards accelerated rates for distributed optimization over time-varying networks. *arXiv preprint arXiv:2009.11069*, 2020.

[47] Alexander Rogozin, Demyan Yarmoshik, Ksenia Kopylova, and Alexander Gasnikov. Decentralized strongly-convex optimization with affine constraints: Primal and dual approaches. *arXiv preprint arXiv:2207.04555*, 2022.

[48] Kevin Scaman, Francis Bach, Sébastien Bubeck, Yin Tat Lee, and Laurent Massoulié. Optimal algorithms for smooth and strongly convex distributed optimization in networks. In *International Conference on Machine Learning*, pages 3027–3036, 2017.

[49] Zhuoqing Song, Lei Shi, Shi Pu, and Ming Yan. Optimal gradient tracking for decentralized optimization. *arXiv preprint arXiv:2110.05282*, 2021.

[50] Fedor Stonyakin, Alexander Tyurin, Alexander Gasnikov, Pavel Dvurechensky, Artem Agafonov, Darina Dvinskikh, Mohammad Alkousa, Dmitry Pasechnyuk, Sergei Artamonov, and Victorya Piskunova. Inexact model: a framework for optimization and variational inequalities. *Optimization Methods and Software*, 36(6):1155–1201, 2021.

[51] John Nikolas Tsitsiklis. Problems in decentralized decision making and computation. Technical report, Massachusetts Inst of Tech Cambridge Lab for Information and Decision Systems, 1984.

[52] Lin Xiao, Stephen Boyd, and Seung-Jean Kim. Distributed average consensus with least-mean-square deviation. *Journal of parallel and distributed computing*, 67(1):33–46, 2007.

[53] Haishan Ye, Ziang Zhou, Luo Luo, and Tong Zhang. Decentralized accelerated proximal gradient descent. *Advances in Neural Information Processing Systems*, 33, 2020.