Decentralized and Parallel Primal and Dual Accelerated Methods for Stochastic Convex Programming Problems

Darina Dvinskikh, and Alexander Gasnikov *

Abstract

We introduce primal and dual stochastic gradient oracle methods for decentralized convex optimization problems. Both for primal and dual oracles, the proposed methods are optimal in terms of the number of communication steps. However, for all classes of the objective, the optimality in terms of the number of oracle calls per node takes place only up to a logarithmic factor and the notion of smoothness. By using mini-batching technique, we show that the proposed methods with stochastic oracle can be additionally parallelized at each node. The considered algorithms can be applied to many data science problems and inverse problems.

1 Introduction

We consider the stochastic convex optimization problem

$$\min_{x \in Q \subseteq \mathbb{R}^n} f(x) := \mathbb{E}[f(x, \xi)]. \tag{1}$$

Such kind of problems arise in many applications of data science [Shalev-Shwartz and Ben–David, 2014, Shapiro et al., 2009] and mathematical statistics [Spokoiny et al., 2012]. To solve this problem with the average precision ε in the function value (i.e., to find such x^N that $\mathbb{E}f(x^N) - \min\limits_{x \in Q} f(x) \leq \varepsilon$), one can use stochastic gradient (mirror) descent [Juditsky and Nemirovski, 2012] with

$$\min \left\{ O\left(\frac{M^2 R^2}{\varepsilon^2}\right), \; O\left(\frac{M^2}{\mu \varepsilon}\right) \right\} \tag{2}$$

*D. Dvinskikh (*darina.dvinskikh@wias-berlin.de)* is with Weierstrass Institute for Applied Analysis and Stochastics, and Moscow Institute of Physics and Technology, and Institute for Information Transmission Problems RAS. A. Gasnikov (*gasnikov@yandex.ru*) is with Moscow Institute of Physics and Technology, and Institute for Information Transmission Problems RAS, and Weierstrass Institute for Applied Analysis and Stochastics

number of calculations of unbiased stochastic subgradients $\nabla f(x,\xi)$. Here μ is the constant of strong convexity of f, $R = \|x^0 - x_*\|_2$ is the Euclidean distance between starting point x^0 and the solution x^* of (1) that corresponds to the minimum of this norm. We also used $\mathbb{E}[\|\nabla f(x,\xi)\|_2^2] \leq M^2$. Generally, we can parallelize (2) on no more than $\tilde{O}(1)$ processors by using batch-parallelization [Dvurechensky et al., 2018]. If we additionally assume that f has L-Lipschitz (continuous) gradient and $\mathbb{E}[\|\nabla f(x,\xi) - \nabla f(x)\|_2^2] \leq \sigma^2$, then (2) is replaced by

$$\min\left\{O\left(\sqrt{\frac{LR^2}{\varepsilon}}\right) + O\left(\frac{\sigma^2 R^2}{\varepsilon^2}\right),\ O\left(\sqrt{\frac{L}{\mu}}\ln\left(\frac{\mu R^2}{\varepsilon}\right)\right) + O\left(\frac{\sigma^2}{\mu\varepsilon}\right)\right\} \tag{3}$$

number of calculations of unbiased stochastic subgradients $\nabla f(x,\xi)$ by using batch parallelization [Devolder, 2013, Dvurechensky and Gasnikov, 2016, Gasnikov and Nesterov, 2018, Ghadimi and Lan, 2013]. In this case we can parallelize subgradients calculations on no more than

$$O\left(\frac{\sigma^2 R^2/\varepsilon^2}{\sqrt{LR^2/\varepsilon}}\right) \quad \text{or} \quad O\left(\frac{\sigma^2/(\mu\varepsilon)}{\sqrt{L/\mu}\ln\left(\mu R^2/\varepsilon\right)}\right)$$

processors (depending on where the minimum in (3) is reached). Notice that this is much better than in previous case. Since this result cannot be improved [Woodworth et al., 2018], it is the best possible way (in general) to solve (1) by using parallel architecture in online context [Shalev-Shwartz et al., 2009].

For many reasons, in some situations in practice, it can be impossible to organize model-based request[1] for calculating stochastic gradient $\nabla f(x^k,\xi^k)$ in online regime. Typically, in machine learning applications [Hastie et al., 2001, Shalev-Shwartz and Ben-David, 2014], instead of online access to $\left\{\nabla f(x^k,\xi^k)\right\}_{k=1}^m$ we have offline access. This means that the set of functions $\left\{f(x,\xi^k)\right\}_{k=1}^m$ are stored in the memory and to use them in algorithms, we need to request corresponding function and then calculate its gradient. This may significantly change the complexity of the problem. Indeed, it is known from [Guigues et al., 2017, Shalev-Shwartz et al., 2009, Shapiro et al., 2009] that with high probability the exact solution of problem

$$\min_{x\in Q\subseteq\mathbb{R}^n} \tilde{f}(x) := \frac{1}{m}\sum_{k=1}^m f(x,\xi^k) \tag{4}$$

is an ε-solution (in the function value) of problem (1) if

$$m = \min\left\{\tilde{O}\left(\frac{nM^2R^2}{\varepsilon^2}\right), \tilde{O}\left(\frac{M^2}{\mu\varepsilon}\right)\right\}.$$

[1]For desired x and independently generated ξ, a request returns $\nabla f(x,\xi)$. This allows not to keep the set of functions $\{f(x,\xi^k)\}_k$ for different k in the memory.

If $\mu = 0$ or it is small enough one may use a regularization technique (see e.g., [Gasnikov, 2017, Shalev-Shwartz et al., 2009]). This allows to reduce the first part of the estimate from $\tilde{O}\left(nM^2R^2/\varepsilon^2\right)$ to $\tilde{O}\left(M^2R^2/\varepsilon^2\right)$. Moreover, we cannot typically find the exact solution of (4) but in the μ-strongly convex (or regularized) case it suffices to solve (4) with accuracy $O(\mu\varepsilon^2/M^2)$ (see [Shalev-Shwartz et al., 2009]).

To solve (4) in offline context we have to store $\{f(x,\xi^k)\}_{k=1}^m$ in the memory. Since m can be large, centralized distributed architecture is often more preferable in this context [Bertsekas and Tsitsiklis, 1989]. In the general case, centralized architecture is based on communication network and it can be obtained by building a spanning tree of a given network [Scaman et al., 2017]. For m-node centralized distributed architecture, the number of gradient oracle calls per each node is defined by (3) with $\sigma^2 = 0$, L and μ corresponding to $\tilde{f}$. The number of communication steps will be d times more, where d is the distance between the origin (root) and farthest node. If we have only $q \ll m$ nodes, then we divide the data $\{f(x,\xi^k)\}_{k=1}^m$ into q blocks with $l = m/q$ terms in each block. If l is too large by itself one can reformulate (4) as follows [McMahan et al., 2016]

$$\min_{x\in Q\subseteq\mathbb{R}^n} \tilde{f}(x) := \frac{1}{q}\sum_{k=1}^{q} \mathbb{E}[f_k(x,\eta^k)], \tag{5}$$

where $f_k(x,\eta^k) = f(x,\xi^{kl+\eta^k})$ and $\eta^k = i$ with probability $1/l$, $i = 1,...,l$. Representation (5) allows to use bound (3) in the stochastic case in a parallel manner at each node. The number of oracle calls per node also corresponds (in general) to (3) and the number of communication steps is also d times more than (3) with $\sigma^2 = 0$.

Unfortunately, centralized architecture has a synchronization drawback and a high requirement for the master node [Scaman et al., 2017]. To address these disadvantages to some extent, a decentralized distributed architecture should be used [Bertsekas and Tsitsiklis, 1989, Kibardin, 1979]. This architecture relies on two basic principles [Nedic, 2020]: every node communicates only with its neighbors, and all communications are performed simultaneously. The main difference here is a simple strategy of communications: each node communicates only with all available direct neighbors. This architecture is more robust. In particular, it can be applied to time-varying (wireless) communication networks [Rogozin et al., 2018].

As problems (4) and (5) have a definite structure of the sum type, they can be solved much faster on one machine. For instance, using some incremental algorithms [Allen-Zhu, 2017, Lan and Zhou, 2017, Lin et al., 2015, Woodworth and Srebro, 2016], one can solve (4) $\sqrt{m}$ times cheaper in terms of the number of oracle calls, but not in terms of the number of iterations (=communication steps). Unfortunately, this result prohibits parallelization. Note, that for this problem in asynchronized mode (at each step only two randomly chosen nodes can

communicate), one can obtain such ($\sim \sqrt{m}$) an acceleration for the star-type communication network [Lan and Zhou, 2018]. Moreover, a 'dual' analogue of this acceleration has recently been proposed for (4) [Hendrikx et al., 2018] and (5) [Hendrikx et al., 2019a, Hendrikx et al., 2019b] with arbitrary communication networks.

Before stating the contribution we introduce the notions of condition number χ for Laplacian communication matrix of some network and the height of spanning tree denoted by d. Note, that $\sqrt{\chi} \geq d$ and typically $\sqrt{\chi} \leq nd$ (see [Nedić et al., 2018]). The last bound corresponds to a star topology [Gasnikov, 2017] (the most simple centralized type architecture). In many interesting cases $\sqrt{\chi} = \tilde{O}(d)$ (see [Nedić et al., 2018, Scaman et al., 2017]).

1.1 Contribution

- We justify the transition from the optimal centralized distributed complexity bounds for problems (4) and (5) in the smooth case to decentralized ones[2] by replacing d with $\sqrt{\chi}$, the average L with the worse one and variance of f with the variance of f_k, that can be m times more.[3] Here and everywhere below we keep μ at the average level (without loss of generality, we can assume that each f_k has the same μ) by using a trick from [Scaman et al., 2017]. The announced results are also not improvable in terms of communication steps (rounds) [Arjevani and Shamir, 2015, Scaman et al., 2017].

  By using different smoothing techniques [Allen-Zhu and Hazan, 2016, Nesterov, 2005, Scaman et al., 2018], we may lead the non-smooth case to the smooth one with $L \sim 1/\varepsilon$. This allows to reduce the complexity estimate (2) by using (3). However, in general, this reduction makes the cost of oracle calls more expensive. Thus, we can only improve the communication steps (rounds) bound that corresponds to (3) (up to a $\sqrt{\chi}$ factor) with $L \sim 1/\varepsilon$ and $\sigma^2 = 0$.[4] Can we preserve the bound (2) for standard conception of oracle calls (primal oracle that gives ∇f_k) per node in decentralized approach by improving the number of communication steps? The answer is positive up to the replacement of the average M to the worst one [Lan et al., 2017, Scaman et al., 2018]. In our paper, we simplify the approaches proposed in these articles to prove this result.

[2] In the deterministic case this was partially done in [Li et al., 2018].

[3] For instance, this takes place in the case when we have independent noise at each f_k. Note also that in this case in decentralized distributed optimization one can improve the variance dependence and eliminate factor m (see [Olshevsky et al., 2019a, Olshevsky et al., 2019b]). But, this is possible due to the worse estimate for the number of communication steps.

[4] Note that such tricks sometimes allow to obtain the optimal (in terms of dependence on ε) communication round estimates [Arjevani and Shamir, 2015, Scaman et al., 2018].

As a motivation for the second main result, we consider the problem of type

$$\min_{x \in Q} f(x) := \frac{1}{m} \sum_{k=1}^{m} f_k(x). \tag{6}$$

where $f_k(x)$ has Fenchel–Legendre representation $f_k(x) = \max_y \{\langle x, y\rangle - \varphi_k(y)\}$ with convex $\varphi_k(y)$. Such type of optimization problems arise, for instance, in different applications of inverse problems, especially for linear problems in Hilbert space where we have to use discretization and sum-type functions naturally arise [Byrne, 2014, Gao and Blumensath, 2017, Gasnikov et al., 2017, Vogel, 2002, Ye et al., 2019]. The case of dual-friendly f_k also arises in the problem of Wasserstein barycenter calculation [Dvinskikh et al., 2019, Dvurechenskii et al., 2018, Uribe et al., 2018]. Suppose that $\nabla\varphi_k(y)$ is available but $f_k(x), \nabla f_k(x)$ are unavailable. Moreover, sometimes $\varphi_k(y) = \mathbb{E}[\varphi_k(y, \xi)]$ and it is worth to use $\nabla\varphi_k(y, \xi)$ instead of $\nabla\varphi_k(y)$ [Dvinskikh et al., 2019, Dvurechenskii et al., 2018]. Considering this example as one of the motivation for using dual oracle instead of primal one, we provide the second main result.

- We develop optimal decentralized distributed algorithms with dual (stochastic) oracle for strongly convex objective in (6). The approach is based on dual reformulation of (6) [Scaman et al., 2017]. An optimal algorithm for non-strongly convex dual function with stochastic oracle was recently proposed in [Dvinskikh et al., 2019]. To propose an optimal method with stochastic dual oracle for strongly convex primal objective, we use recent work [Foster et al., 2019]. We notice a rather unexpected result: we cannot improve (up to a logarithmic factor) the bound for the number of dual stochastic gradient calculations in comparison with non-strongly convex dual objective.

  We also notice that initially we were motivated by the study of the dual oracle not only as an application from [Dvinskikh et al., 2019, Dvurechenskii et al., 2018, Uribe et al., 2018]. We also tried to find a simple explanation for the optimal communication step bounds [Arjevani and Shamir, 2015, Scaman et al., 2018] in non-smooth case. One of the ways to do it is Nesterov's dual smoothing technique [Nesterov, 2005] that builds a bridge to the notion of dual oracle. This plan was partially (in the deterministic case) implemented in [Scaman et al., 2017, Uribe et al., 2020, Uribe et al., 2020]. Here we generalize the results of these works for the stochastic dual oracle.

1.2 Paper organization

The paper is organized as follows. In Section 2, we propose optimal stochastic (parallelized) accelerated gradient methods for stochastic convex optimization problems. In Sections 3 and

4, we apply the results of Section 2 to stochastic convex optimization problems with affine type of constraints (of type $Ax = 0$). We describe the modern stochastic (parallelized) accelerated gradient methods which are optimal both in terms of (stochastic) oracle calls and matrix-vector multiplications Ax. In Sections 3, we are focusing on primal methods, in Section 4, we present dual ones. Section 5, describes the distributed primal and dual formulation of the finite-sum minimization problem, and presents distributed algorithms. In Section 6, we incorporate the proposed distributed decentralized method to get the optimal bounds for the finite-sum minimization problem using primal or dual oracle. Finally, we discuss future work and possible extensions. We notice that all proposed methods are optimal in terms of communication steps and in many cases in terms of (parallel stochastic) primal/dual oracle calls.

2 Stochastic convex optimization

First-order methods for the optimization problem of minimizing a convex function f on a simple convex set Q, e.g.,

$$\min_{x \in Q \subseteq \mathbb{R}^n} f(x), \tag{7}$$

play a fundamental role in modern problems arising in machine learning and statistics. The complexity of these methods is measured by the number of iterations or (and) the number of oracle calls. For a deterministic oracle, this concept can be identified. By the first-order oracle, we mean a black-box model that for a given input $x \in Q$, returns the vector $\nabla f(x)$.

We say that a function f is M-Lipschitz continuous if [5]

$$\forall x \in Q \qquad \|\nabla f(x)\|_2 \leq M.$$

We say that function f is L-smooth or has L-Lipschitz continuous gradient if

$$\forall x, y \in Q \qquad \|\nabla f(y) - \nabla f(x)\|_2 \leq L\|y - x\|_2.$$

We also say that function f is μ-strongly convex if

$$\forall x, y \in Q \qquad f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle + \frac{\mu}{2}\|y - x\|_2^2.$$

Accelerated gradient methods (e.g., Algorithm 1 (`STM`) [Gasnikov and Nesterov, 2018, Nesterov, 2018b, Lan, 2019]) allow to obtain the optimal number of iterations and number

[5] Here and below in such type of assumptions (especially in the case when Q is unbounded) instead of $\forall x \in Q$ we may write $\forall x \in Q : \|x - x^*\|_2 \leq 2R$ [Gasnikov, 2017] (analogously for y).

Algorithm 1 Similar Triangles Method $\texttt{STM}(L, \mu, x^0)$, $Q = \mathbb{R}^n$

Input: $\tilde{x}^0 = z^0 = x^0$, number of iterations N, $\alpha_0 = A_0 = 0$, L, μ

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

2: Set $\alpha_{k+1} = \frac{1+A_k\mu}{2L} + \sqrt{\frac{1+A_k\mu}{4L^2} + \frac{A_k(1+A_k\mu)}{L}}$, $A_{k+1} = A_k + \alpha_{k+1}$

3: $\tilde{x}^{k+1} = (A_k x^k + \alpha_{k+1} z^k)/A_{k+1}$

4: $z^{k+1} = z^k - \frac{\alpha_{k+1}}{1+A_{k+1}\mu}\left(\nabla f(\tilde{x}^{k+1}) + \mu\left(z^k - \tilde{x}^{k+1}\right)\right)$

5: $x^{k+1} = (A_k x^k + \alpha_{k+1} z^{k+1})/A_{k+1}$

6: **end for**

Output: x^N

of gradient oracle calls for problem (7) as described in Table 1, where $R = \|x^0 - x^*\|_2$ is the Euclidean distance from the starting point x^0 to the solution x^* of (7) that corresponds to the minimum of this norm, and ε is the desired precision in function value.

Remark 1 *For a composite optimization problem with composite term $h(x)$, step 4 of Algorithm 1 is replaced by the more general operator [Gasnikov and Nesterov, 2018, Nesterov, 2018b]*

$$z^{k+1} = \arg\min_{z \in Q} \left\{ \sum_{l=0}^{k+1} \alpha_l \left(\langle \nabla f(\tilde{x}^l), z - \tilde{x}^l \rangle + h(z) + \frac{\mu}{2}\|z - \tilde{x}^l\|_2^2 \right) + \frac{1}{2}\|z - \tilde{x}^0\|_2^2 \right\}.$$

If $h(x)$ has L_h-Lipschitz gradient in the ℓ_2-norm then due to Theorem 9 [Stonyakin et al., 2019b] and Theorem 19 [Stonyakin et al., 2019a] it suffices to solve auxiliary problem with accuracy (in terms of function value)

$$O\left(\frac{(\alpha_{k+1}\varepsilon)^2(A_{k+1}\mu + 1)}{(A_{k+1}L_h R)^2}\right) \geq O\left(\frac{\varepsilon^3}{LL_h^2 R^4}\right),$$

where ε is desired accuracy (in function value) for initial problem (7).
If $\mu = 0$ we can also generalize this step for the non-Euclidean case and using restarts [Gasnikov and Nesterov, 2018] generalize such a method on $\mu > 0$. Note, that by using restarts with $\texttt{STM}(L, 0, x^0)$ one can eliminate the gap from $\ln(LR^2/\varepsilon)$ to $\ln(\mu R^2/\varepsilon)$ between lower bounds and the bounds for $\texttt{STM}(L, \mu, x^0)$ without restarts [Gasnikov and Nesterov, 2018]. The same remains true for the stochastic oracle.

Table 1: The optimal number of first-order oracle calls (number of iterations N)

	μ-strongly convex and L-smooth	L-smooth	μ-strongly convex	
#iterations	$\sqrt{\frac{L}{\mu}}\ln\left(\frac{\mu R^2}{\varepsilon}\right)$	$\sqrt{\frac{LR^2}{\varepsilon}}$	$\frac{M^2}{\mu\varepsilon}$	$\frac{M^2R^2}{\varepsilon^2}$
#oracle calls of $\nabla f(x)$	$\sqrt{\frac{L}{\mu}}\ln\left(\frac{\mu R^2}{\varepsilon}\right)$	$\sqrt{\frac{LR^2}{\varepsilon}}$	$\frac{M^2}{\mu\varepsilon}$	$\frac{M^2R^2}{\varepsilon^2}$

Generally, iteration complexity is determined by the complexity of calculating the gradient, which can be computationally expensive. Thus, stochastic approximations of the true gradient can be used instead. In this case, or when the true gradient is unavailable (if e.g., function f is given in the form of expectation $f(x) := \mathbb{E}[f(x,\xi)]$) we denote the inexact (or noise-corrupted) first-order oracle as $\nabla f(x,\xi)$, given by a blackbox model with stochasticity (noise) ξ corrupting the true gradient. Assume that

$$\|\mathbb{E}[\nabla f(x,\xi)] - \nabla f(x)\|_2 \leq \delta = O(\varepsilon/R) \tag{8}$$

and

$$\mathbb{E}\left[\exp\left(\|\nabla f(x,\xi) - \mathbb{E}[\nabla f(x,\xi)]\|_2^2\sigma^{-2}\right)\right] \leq \exp(1),$$

then with probability at least $1-\beta$, we have $f(x^N) - f(x^*) \leq \varepsilon$ after

$$N = \min\left\{O\left(\sqrt{\frac{LR^2}{\varepsilon}}\right), O\left(\sqrt{\frac{L}{\mu}}\ln\left(\frac{LR^2}{\varepsilon}\right)\right)\right\}$$

iterations of `STM` using the approximated gradient (instead of the real one $\nabla f(\tilde{x}^{k+1})$)

$$\nabla^{r_{k+1}} f(\tilde{x}^{k+1}, \{\xi_i^{k+1}\}_{i=1}^{r_{k+1}}) = \frac{1}{r_{k+1}}\sum_{i=1}^{r_{k+1}} \nabla f(\tilde{x}^{k+1}, \xi_i^{k+1}), \tag{9}$$

where $\xi_1^{k+1}, \ldots, \xi_{r_{k+1}}^{k+1}$ are i.i.d from the same distribution as ξ and the batch size is

$$r_{k+1} = O\left(\frac{\sigma^2\alpha_{k+1}\ln(N/\beta)}{(1+A_{k+1}\mu)\varepsilon}\right). \tag{10}$$

Moreover, the total number of oracle calls[6] is (this bound is optimal up to logarithmic factors)

$$\sum_{k=0}^{N} r_k = O(N) + \min\left\{ O\left(\frac{\sigma^2 R^2}{\varepsilon^2}\ln\left(\frac{\sqrt{LR^2/\varepsilon}}{\beta}\right)\right), \right.$$
$$\left. O\left(\frac{\sigma^2}{\mu\varepsilon}\ln\left(\frac{LR^2}{\varepsilon}\right)\ln\left(\frac{\sqrt{L/\mu}}{\beta}\right)\right)\right\}.$$

We refer to such a variant of `STM` as `BSTM(`L`,`μ`,`σ^2`,`x^0`)` (batched `STM(`L`,`μ`,`x^0`)`).

Thus, using minibatches for constructing an approximation of the true gradient allows us to keep the optimal number of iterations for stochastic methods, as presented in Table 1, where we skip high probability logarithmic multipliers. The number of stochastic oracle calls for this case is shown in Table 2.

Remark 2 *We notice that in assumption (8), $R = \|x^0 - x^*\|_2$. Generally, in such type of assumptions, R is the diameter of Q (see [Cohen et al., 2018, d'Aspremont, 2008]) (it is not a compact set when $Q = \mathbb{R}^n$). To obtain such a generalization we have to use the advanced recurrent technique to bound $\|z^k - x^*\|_2$ from [Dvinskikh et al., 2019, Gorbunov et al., 2018] and [Gasnikov, 2017, Chapter 2]. Further we provide the sketches how to get this result (for simplicity $\sigma = 0, \mu = 0$).*

1. For an inexact gradient $\tilde{\nabla} f(x)$ satisfying for all x, y

$$f(x) + \langle \tilde{\nabla} f(x), y - x\rangle - \delta_1\|y - x\|_2 \le f(y)$$
$$\le f(x) + \langle \tilde{\nabla} f(x), y - x\rangle + \frac{L}{2}\|y - x\|_2^2 + \delta_2, \tag{11}$$

`STM` *outputs*[7] x^N *such that [Devolder et al., 2014, Dvinskikh et al., 2020]*

$$f(x^N) - f(x^*) = O\left(\frac{LR^2}{N^2} + \delta_1\tilde{R} + N\delta_2\right),$$

where $\max\left\{\|\tilde{x}^k - x^*\|_2, \|z^k - x^*\|_2, \|x^k - x^*\|_2\right\} \le \tilde{R}$.

2. Since

$$\langle \tilde{\nabla} f(x) - \nabla f(x), y - x\rangle \le \frac{1}{2L}\|\tilde{\nabla} f(x) - \nabla f(x)\|_2^2 + \frac{L}{2}\|y - x\|_2^2,$$

[6] Oracle calls can be easily and fully parallelized (on r_k processors) at each iteration. Note, that for $\nabla^{r_k} f(x, \{\xi_i\}_{i=1}^{r_k})$ we can reduce the variance $\sigma^2 := O(\sigma^2/r_k)$.

[7] Note, that according to [Poljak, 1981] even for the last point of gradient descent on a simple quadratic optimization problem we cannot guarantee convergence without proper stopping rule. With proper stopping rule in (8) it is required (see [Polyak, 1987, Theorem 7, item 6.1.3]) $\delta \sim \varepsilon^2$ that is worse than what we have $\delta \sim \varepsilon$. But we can guarantee standard convergence of noisy gradient descent under (8) in (Cesaro) average [Gasnikov, 2017] (not for the last point). The results below generalize [Gasnikov, 2017] on a proper accelerated method (STM).

one can consider $\delta_2 := \delta^2/(2L)$ *and* $L := 2L$ *in* (11).
3. In the deterministic case, $\tilde{R} = R$ *with the proper stopping rule of the algorithm (see [Gasnikov, 2017, formulas (2.17), (2.18) in Chapter 2] and [Gasnikov and Nesterov, 2018]). In the stochastic case (with high probability)* $\tilde{R} = O(R)$ *(see [Dvinskikh et al., 2019, Gorbunov et al., 2018]) for* `STM` *with the proper batch-size* (10).

In particular, for the case of non-smooth objective, the stochastic oracle does not yield gains compared to its deterministic counterpart.

Table 2: The optimal number of stochastic (unbiased) first-order oracle calls

	μ-strongly convex and L-smooth	L-smooth	μ-strongly convex	
#iterations	$\sqrt{\frac{L}{\mu}} \ln\left(\frac{\mu R^2}{\varepsilon}\right)$	$\sqrt{\frac{LR^2}{\varepsilon}}$	$\frac{M^2}{\mu\varepsilon}$	$\frac{M^2R^2}{\varepsilon^2}$
#oracle calls of $\nabla f(x,\xi)$	$\max\left\{\frac{\sigma^2}{\mu\varepsilon}, \sqrt{\frac{L}{\mu}} \ln\left(\frac{\mu R^2}{\varepsilon}\right)\right\}$	$\max\left\{\frac{\sigma^2R^2}{\varepsilon^2}, \sqrt{\frac{LR^2}{\varepsilon}}\right\}$	$\frac{M^2}{\mu\varepsilon}$	$\frac{M^2R^2}{\varepsilon^2}$

In Table 2, in the appropriate cells, we assumed that the following inequalities hold: $\mathbb{E}\|\nabla f(x,\xi) - \nabla f(x)\|_2^2 \le \sigma^2$ and $\mathbb{E}\|\nabla f(x,\xi)\|_2^2 \le M^2$.

Both in Table 1 and in Table 2, the last two columns can be obtained from the corresponding first columns by choosing $L = M^2/(2\delta)$, where $\delta = \varepsilon/N$ (see [Gasnikov and Nesterov, 2018]). This is the idea of universal accelerated methods [Nesterov, 2015], but with predefined L. Here and in all further tables we skip numerical constants.

3 Primal methods for stochastic convex optimization with affine constraints

To build the complete theory of distributed primal and dual method we need to generalize the result of Tables 1 and 2 for the convex optimization problem[8]

$$\min_{\substack{Ax=0,\\ x\in Q}} f(x), \tag{12}$$

[8]In decentralized optimization A is taken to be $\sqrt{W}$ (square root of the Laplacian matrix of the communication network).

where $A \succeq 0$ and $\mathrm{Ker}A \neq \varnothing$. The purpose of this section is to develop such algorithms for (12) that are optimal in terms of the number of $\nabla f(x)$ calculations and the number of A^TAx calculations. In this section we use Euclidean proximal setup [Ben-Tal and Nemirovski, 2001]. This is the only section where we significantly rely on Euclidean prox-structure.

Denote by $R_y = \|y^*\|_2$ the ℓ_2-norm of the smallest solution y^* of dual (up to a sign) problem (16). Solution y^* is not unique since $\mathrm{Ker}A \neq \varnothing$. From [Lan et al., 2017] we have such a bound

$$R_y^2 \leq \frac{\|\nabla f(x^*)\|_2^2}{\lambda_{\min}^+(A^TA)}. \tag{13}$$

Using the penalty method we rewrite (12) as follows

$$F(x) = f(x) + \frac{R_y^2}{\varepsilon}\|Ax\|_2^2 \to \min_{x \in Q}. \tag{14}$$

Next we use [Gasnikov, 2017, Remark 4.2] and get if the following holds

$$F(x^N) - \min_{x \in Q} F(x) \leq \varepsilon,$$

then

$$f(x^N) - \min_{x \in Q, Ax = 0} f(x) \leq \varepsilon, \quad \|Ax^N\|_2 \leq \frac{(1+\sqrt{5})\varepsilon}{2R_y}.$$

We start with the smooth case and assume that $Q = \mathbb{R}^n$. If f has L-Lipschitz continuous gradient then we can solve (14) by `STM` (or `BSTM` in the stochastic case) considering the second term to be composite [Gasnikov and Nesterov, 2018, Nesterov, 2013]. In this case, we obtain the optimal number of $\nabla f(x)$ (or $\nabla f(x, \xi)$) calculations, see Tables 1, 2. But the total number of A^TAx calculations is

$$\tilde{O}\left(\sqrt{\lambda_{\max}(A^TA)/\lambda_{\min}^+(A^TA)}\right)$$

times more since $\mathrm{Im}A = \mathrm{Im}A^T = (\mathrm{Ker}A)^\perp$ and $Q = \mathbb{R}^n$ and as a consequence of these facts the auxiliary problem can be divided into two subproblems: minimization of quadratic form with matrix of the form $(R_y^2/\varepsilon)A^TA + cI$ (c is some positive constant and I is identity matrix) on $(\mathrm{Ker}A)^\perp$ and minimization of quadratic form with matrix of the form cI on $\mathrm{Ker}A$. Linear terms do not play any role in complexity. The complexity of the auxiliary problem is determined by the worst (reduced on corresponding subspace) conditional number of these two subproblems. Obviously, the first one is worse. The reduced the conditional number is

$$\frac{\lambda_{\max}\left((R_y^2/\varepsilon)A^TA + cI\right)}{\lambda_{\min}^+\left((R_y^2/\varepsilon)A^TA + cI\right)} \leq \frac{\lambda_{\max}(A^TA)}{\lambda_{\min}^+(A^TA)}.$$

This factor arises because of the complexity of the auxiliary problem. We refer to these approaches as `PSTM` and `PBSTM` (Penalty `STM` and `BSTM`). Here and below we skip arguments of the algorithms if they are obvious from the context.

In non-smooth case (f is M-Lipschitz), we use the Sliding algorithm [Lan, 2016], [Lan, 2019] . If $\mu = 0$ according to [Lan, 2016] this algorithm requires (see Tables 1, 2 for comparison)

$$O\left(\sqrt{\frac{\lambda_{\max}(A^TA)R_y^2R_x^2}{\varepsilon^2}}\right)$$

calculations of A^TAx and

$$O\left(\frac{M^2R_x^2}{\varepsilon^2}\right)$$

calculations of $\nabla f(x)$, where $R_x = \|x^0 - x^*\|_2$.

If we have unbiased $\nabla f(x, \xi)$ with σ^2-sub-Gaussian variance [Jin et al., 2019] instead of $\nabla f(x)$, i.e.,

$$\mathbb{E}\left[\exp\left(\|\nabla f(x,\xi) - \nabla f(x)\|_2^2\sigma^{-2}\right)\right] \leq \exp(1),$$

with $\sigma^2 = O(M^2)$ (for compact notation[9]), then the bound for calculations of A^TA does not change and the bound for calculations of $\nabla f(x, \xi)$ is the same as it was for the number of calculations of $\nabla f(x)$ in deterministic case (up to a logarithmic high-probability deviations factor).

By using a restart technique [Uribe et al., 2020] we can generalize this method for μ-strongly convex f:

$$O\left(\sqrt{\frac{\lambda_{\max}(A^TA)R_y^2}{\mu\varepsilon}}\ln\left(\frac{\mu R_x^2}{\varepsilon}\right)\right)$$

calculations of A^TAx and

$$O\left(\frac{M^2}{\mu\varepsilon}\right)$$

calculations of $\nabla f(x)$ ($\nabla f(x, \xi)$).

We call this approach by `R`-Sliding (Restart Sliding).

[9] In general M^2 is replaced by $M^2 + \sigma^2$ in the stochastic case.

4 Dual methods for stochastic convex optimization with affine constraints

Now we assume that we can build a dual problem for

$$\min_{\substack{Ax=0,\\ x\in Q}} f(x), \tag{15}$$

where $\mathrm{Ker} A \neq \varnothing$.

Remark 3 *We notice that turning to a dual problem does not oblige us using dual oracle. Instead, we can use a primal oracle and the Moreau theorem [Rockafellar, 2015] with Fenchel-Legendre representation. This maximization problem can be solved using the first-order oracle for the function f. But such an approach does not allow to obtain the optimal bounds on the number of primal first-order oracle calls. Note that typically in decentralized optimization A in (15) is taken as the square root of the Laplacian matrix W of the communication network [Scaman et al., 2017]. But in the asynchronized case the square root $\sqrt{W}$ replaced by incidence matrix M [Hendrikx et al., 2018] ($W = M^T M$). Then in asynchronized case instead of accelerated methods for (16) one should use an accelerated (block) coordinate descent method [Dvurechensky et al., 2017, Gasnikov, 2017, Hendrikx et al., 2018, Shalev-Shwartz and Zhang, 2014].*

Algorithm 2 `PDSTM`

Input: $\tilde{y}^0 = z^0 = y^0 = 0$, number of iterations N, $\alpha_0 = A_0 = 0$

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

2: Set $\alpha_{k+1} = \frac{1}{2L_\psi} + \sqrt{\frac{1}{4L_\psi^2} + \frac{A_k}{L_\psi}}$, $A_{k+1} = A_k + \alpha_{k+1}$

3: $\tilde{y}^{k+1} = (A_k y^k + \alpha_{k+1} z^k)/A_{k+1}$

4: $z^{k+1} = z^k - \alpha_{k+1} \nabla \psi(\tilde{y}^{k+1}) = z^k - \alpha_{k+1} A x(A^T \tilde{y}^{k+1})$

5: $y^{k+1} = (A_k y^k + \alpha_{k+1} z^{k+1})/A_{k+1}$

6: **end for**

Output: y^N, $x^N = \frac{1}{A_N} \sum_{k=0}^N \alpha_k x(A^T \tilde{y}^k)$

The dual problem (up to a sign) is following

$$\begin{aligned} \psi(y) = \varphi(A^T y) &= \max_{x\in Q} \{\langle y, Ax\rangle - f(x)\} = \langle y, Ax(A^T y)\rangle - f(x(A^T y)) \\ &= \langle A^T y, x(A^T y)\rangle - f(x(A^T y)) \to \min_y . \end{aligned} \tag{16}$$

If f is μ-strongly convex in the ℓ_2-norm, then ψ has $L_\psi = \frac{\lambda_{\max}(A^TA)}{\mu}$–Lipschitz continuous gradient in the ℓ_2-norm[10] [Kakade et al., 2009, Rockafellar, 2015]. In this case we can apply `STM(`L_ψ`,0,0)` to (16). Note that due to Demyanov–Danskin's theorem $\nabla\psi(y) = Ax(A^Ty)$ [Rockafellar, 2015]. Similarly to [Anikin et al., 2017, Chernov et al., 2016] one can prove that

$$f(x^N) - f(x^*) = f(x^N) - f(x(A^Ty^*)) \le f(x^N) + \psi(y^N)$$
$$= O\left(\frac{L_\psi R_y^2}{N^2}\right), \quad \|Ax^N\|_2 = O\left(\frac{L_\psi R_y}{N^2}\right), \tag{17}$$

where $R_y = \|y^*\|_2$ is the radius of solution of (16) which is the smallest in the ℓ_2-norm, see (13). We call this approach by `PDSTM` (Primal-Dual `STM`).

If we have only a stochastic (randomized) unbiased model $\nabla\varphi(\lambda,\xi)|_{\lambda=A^Ty} = x(A^Ty,\xi)$ with σ_φ^2-sub-Gaussian variance, i.e.

$$\mathbb{E}\left[\exp\left(\|\nabla\varphi(\lambda,\xi) - \nabla\varphi(\lambda)\|_2^2\sigma_\varphi^{-2}\right)\right]$$
$$= \mathbb{E}\left[\exp\left(\|x(A^Ty,\xi) - x(A^Ty)\|_2^2\sigma_\varphi^{-2}\right)\right] \le \exp(1),$$

then for `BSTM(`L_ψ`,0,`σ_ψ^2`,0)` where $\sigma_\psi^2 = \lambda_{\max}(A^TA)\sigma_\varphi^2$ with probability $\ge 1-\beta$ (17) holds true [Dvinskikh et al., 2019]. We refer to this algorithm as `SPDSTM` (Stochastic `PDSTM`).

In the case when ψ from (16) is additionally μ_ψ-strongly convex in the ℓ_2-norm in[11] $y^0 + (\mathrm{Ker}A^T)^\perp$ (if f has L-Lipschitz gradient in the ℓ_2-norm and $Q = \mathbb{R}^n$ then $\mu_\psi = \lambda_{\min}^+(A^TA)/L$ [Kakade et al., 2009, Rockafellar, 2015], where $\lambda_{\min}^+(A^TA)$ is the minimal positive eigenvalue of A^TA) we need to use another approach. Because of primal-duality [Nesterov, 2009, Nemirovski et al., 2010] we have to put $\mu_\psi = 0$ in `STM` and in methods based on STM (`STM(`L_ψ`,`μ_ψ`,`y^0`)` is not primal-dual method when $\mu_\psi > 0$). The restart technique (see, e.g. [Gasnikov, 2017]) also does not work here because in (17) we have to use $R_y = \|y^0\|_2 + \|y^0 - y^*\|_2$ in general. That is why we take here $y^0 = 0$. So the main trick here is the following relation [Allen-Zhu, 2018, Anikin et al., 2017, Nesterov, 2012]

$$f(x(A^Ty)) - f(x^*) \le \langle\nabla\psi(y), y\rangle = \langle Ax(A^Ty), y\rangle. \tag{18}$$

From (18), to satisfy

$$f(x^N) - f(x^*) = f(x(A^Ty^N)) - f(x(A^Ty^*)) \le 2\varepsilon, \|Ax^N\|_2 \le \varepsilon/R_y,$$

[10] Here and below we can also consider other norms (see [Uribe et al., 2020] for details).

[11] Since $\mathrm{Im}A = (\mathrm{Ker}A^T)^\perp$ we will have that all the points $\tilde{y}^k, z^k, y^k$, generated by STM and methods based on STM, belong to $y^0 + (\mathrm{Ker}A^T)^\perp$. That is, from the point of view of estimates this means, that we can consider ψ to be μ_ψ-strongly convex everywhere.

it is sufficient to find such y^N ($\|y^N\|_2 \leq 2R_y$) that

$$\|\nabla\psi(y^N)\|_2 \leq \varepsilon/R_y. \tag{19}$$

Recently, there appear accelerated methods with the proper rate of convergence in terms of the norm of the gradient `OGM-G` [Gasnikov, 2017, Kim and Fessler, 2018]:

$$\|\nabla\psi(y^N)\|_2 = O\left(\frac{L_\psi\|y^0 - y^*\|_2}{N^2}\right) = O\left(\frac{L_\psi\|\nabla\psi(y^0)\|_2}{\mu N^2}\right).$$

After $\bar{N} = O\left(\sqrt{\frac{L_\psi}{\mu_\psi}}\right)$ iterations of `OGM-G` we have

$$\|\nabla\psi(y^{\bar{N}})\|_2 \leq \frac{1}{2}\|\nabla\psi(y^0)\|_2.$$

So after $l = \log_2\left(\|\nabla\psi(y^0)\|_2\frac{R_y}{\varepsilon}\right)$ restarts ($y^0 := y^{\bar{N}}$) we have (19). We denote such an approach by `ROGM-G` (Restart `OGM-G`). This approach requires

$$O\left(\sqrt{\frac{L_\psi}{\mu_\psi}}\ln\left(\|\nabla\psi(y^0)\|_2\frac{R_y}{\varepsilon}\right)\right)$$

of $\nabla\psi(y)$ (that is $Ax(A^Ty)$) calculations. The key inequality to prove this fact is

$$\|y^0 - y^*\|_2^2 \leq \frac{1}{\mu_\psi^2}\|\nabla\psi(y^0)\|_2^2.$$

This holds due to

$$\frac{\mu}{2}\|y^0 - y^*\|_2^2 \leq \psi(y^0) - \psi(y^*) \leq \frac{1}{2\mu_\psi}\|\nabla\psi(y^0)\|_2^2.$$

The same result with the replacement

$$\sqrt{\frac{L_\psi}{\mu_\psi}}\ln\left(\|\nabla\psi(y^0)\|_2\frac{R_y}{\varepsilon}\right) \to \sqrt{\frac{L_\psi}{\mu_\psi}}\ln\left(2L_\psi^2\frac{R_y^4}{\varepsilon^2}\right)$$

can be obtained by using `STM(`L_ψ`,`μ_ψ`,0)` with bound $\sqrt{\frac{L_\psi}{\mu_\psi}}\ln\left(\frac{L_\psi R_y^2}{\varepsilon'}\right)$ (see [Nesterov, 2010]) and desired accuracy $\varepsilon' = \frac{\varepsilon^2}{2L_\psi R_y^2}$. This follows from

$$\frac{1}{2L_\psi}\|\nabla\psi(y^N)\|_2^2 \leq \psi(y^N) - \psi(y^*) \leq \varepsilon'.$$

Now we consider `RRMA+AC-SA`^{2} [Foster et al., 2019] (see also [Allen-Zhu, 2018] in the non-accelerate, but composite case). This algorithm converges as follows (for simplicity we skip polylogarithmic factors and high probability terminology)

$$\|\nabla\psi(y^N)\|_2^2 = \tilde{O}\left(\frac{L_\psi^2\|y^0 - y^*\|_2^2}{N^4} + \frac{\sigma_\psi^2}{N}\right) = \tilde{O}\left(\frac{L_\psi^2\|\nabla\psi(y^0)\|_2^2}{\mu_\psi^2 N^4} + \frac{\sigma_\psi^2}{N}\right).$$

We assume that at each iteration, $\nabla\psi(y,\xi)$ with sub-Gaussian variance σ_ψ^2 is available [Jin et al., 2019] (see also above). If we use restarts with the size of each restart $\bar{N} = \tilde{O}\left(\sqrt{\frac{L_\psi}{\mu_\psi}}\right)$ (see above) and use batched gradient (9) with batch size (at k-th restart; $\bar{y}^k$ is the output point from the previous restart)

$$r_{k+1} = \tilde{O}\left(\frac{\sigma_\psi^2}{\bar{N}\|\nabla\psi(\bar{y}^{k+1})\|_2^2}\right).$$

then $\|\nabla\psi(\bar{y}^l)\|_2 \leq \varepsilon/R_y$ after $l = O\left(\log_2\left(\|\nabla\psi(y^0)\|_2 R_y/\varepsilon\right)\right)$ restarts. Therefore, the total number of oracle calls is

$$\tilde{O}\left(\frac{\sigma_\psi^2 R_y^2}{\varepsilon^2}\right).$$

Note that the same bound takes place in the non-strongly convex case ($\mu_\psi = 0$). From [Allen-Zhu, 2018, Jin et al., 2019] it is known that this bound cannot be improved. However, we may expect that this bound can be reduced to $\tilde{O}(\sigma_\psi^2/(\mu_\psi\varepsilon))$ (see Table 2 for stochastic primal oracle). For the stochastic dual oracle, such a reduction is probably impossible. We call this approach by `R-RRMA+AC-SA`^{2} (Restart `RRMA+AC-SA`^{2}).

5 Decentralized distributed optimization

Now we show how to present (6) in a decentralized distributed manner

$$\min_{x\in Q\subseteq\mathbb{R}^n} f(x) := \frac{1}{m}\sum_{k=1}^m f_k(x). \tag{P1}$$

This particular representation of the objective in (P1) allows involving distributed methods which are particularly necessary for large-scale problems handling the large quantities of data and which are based on the idea of agents' cooperative solution of the global problem [Bertsekas and Tsitsiklis, 1989]. For a given multi-agent network system, we privately assign each function f_k to the agent k and suppose that agents can exchange the information with

their neighbors (e.g., send and receive vectors). We define this system through the Laplacian matrix $\bar{W} \in \mathbb{R}^{m\times m}$ of some graph (communication network) $G = (V, E)$ with the set V of m vertices and the set of edges $E = \{(i, j) : i, j \in V\}$ as follows

$$\bar{W}_{ij} = \begin{cases} -1, & \text{if } (i,j) \in E, \\ \deg(i), & \text{if } i = j, \\ 0, & \text{otherwise,} \end{cases}$$

where $\deg(i)$ is the degree of vertex i (i.e., the number of neighboring nodes).

From the definition of the matrix $\bar{W}$ it can be easily seen that $\bar{W}$ establishes the communication of agents and allows only the communication between neighboring nodes. Moreover, due to connectivity of graph G the vector $\mathbf{1}_m = (1, ..., 1)^T \in \mathbb{R}^m$ is the unique (up to a scaling factor) eigenvector of $\bar{W}$ associated with the eigenvalue $\lambda = 0$, which allows us to compactly rewrite the consensus agreement $x_1 = ... = x_m \in \mathbb{R}^n$ as $W\mathbf{x} = 0$, moreover, as $\sqrt{W}\mathbf{x} = 0$ (see [Scaman et al., 2017]), where $W = \bar{W} \otimes I_n$ is the Kronecker product of the Laplacian matrix $\bar{W} \in \mathbb{R}^m$ and the identity (unit) matrix I_n and $\mathbf{x} = [x_1^T, ..., x_m^T]^T \in \mathbb{R}^{mn}$.

To present problem (P1) in a distributed fashion we rewrite it with introducing the artificial consensus equality constraints and then change these constraints to one affine constraint with the communication matrix W as follows

$$\min_{\substack{x_1=\cdots=x_m, \\ x_1,\ldots,x_m\in Q\subseteq\mathbb{R}^n}} F(\mathbf{x}) := \frac{1}{m}\sum_{k=1}^{m} f_k(x_k)$$

or in another form

$$\min_{\substack{\sqrt{W}\mathbf{x}=0, \\ x_1,\ldots,x_m\in Q\subseteq\mathbb{R}^n}} F(\mathbf{x}) := \frac{1}{m}\sum_{k=1}^{m} f_k(x_k), \tag{P2}$$

where all f_k are M-Lipschitz, L-smooth and μ-strongly convex (it is possible that, $L = \infty$ or (and) $\mu = 0$).

We also consider the stochastic version of problem (P2), where $f_k(x_k) = \mathbb{E}[f_k(x_k, \xi_k)]$. We consider the unbiased stochastic primal oracle that returns $\nabla f_k(x_k, \xi_k)$ (where $\xi = \{\xi_k\}_{k=1}^m$ are independent) under the following σ^2-sub-Gaussian variance condition (for all $k = 1, ..., m$)

$$\mathbb{E}\left[\exp\left(\frac{\|\nabla f_k(x_k, \xi_k) - \nabla f_k(x_k)\|_2^2}{\sigma^2}\right)\right] \le \exp(1).$$

Problem (P2) can be considered to be a particular case of problem (12) with the following replacements

- $A = \sqrt{W}$

- $L_F = L/m$
- $\mu_F = \mu/m$
- $\|\nabla F(\mathbf{x})\|_2^2 \leq M_F^2 = M^2/m$
- $\sigma_F^2 = O\left(\sigma^2/m\right)$
- $R_{\mathbf{x}}^2 = \|\mathbf{x}^0 - \mathbf{x}^*\|_2^2 = m\|x^0 - x^*\|_2^2 = mR^2$
- $R_{\mathbf{y}}^2 = \|\mathbf{y}^*\|_2^2 \leq \|\nabla F(\mathbf{x}^*)\|_2^2/\lambda_{\min}^+(W) \leq M^2/(m\lambda_{\min}^+(W))$

The main observation in the primal approach (see Section 3) is as follows [Scaman et al., 2017]:

$$\boxed{A^T A x = W x \text{ (calculated in a decentralized distributed manner)}}$$

If each function f_k is a dual-friendly (dual function is available calculated by the Fenchel–Legendre transform [Uribe et al., 2020]) then we can construct the dual problem to problem (P2) with dual Lagrangian variables $\mathbf{y} = [y_1^T \in \mathbb{R}^n, \cdots, y_m^T \in \mathbb{R}^n]^T \in \mathbb{R}^{mn}$

$$\min_{\mathbf{y} \in \mathbb{R}^{mn}} \Psi(\mathbf{y}) := \frac{1}{m}\Phi(m\sqrt{W}\mathbf{y}) := \frac{1}{m}\sum_{k=1}^{m} \varphi_k(m[\sqrt{W}\mathbf{y}]_k), \tag{D2}$$

where $\varphi_k(\lambda_k) = \max\limits_{x_k \in Q \subseteq \mathbb{R}^n} \{\langle \lambda_k, x_k \rangle - f_k(x_k)\}$ and the vector $[\sqrt{W}\mathbf{x}]_k$ represents the k-th n-dimensional block of $\sqrt{W}\mathbf{x}$.

We also consider the stochastic version of problem (D2), where $\varphi_k(\lambda_k) = \mathbb{E}[\varphi_k(\lambda_k, \xi_k)]$. We consider the unbiased stochastic dual oracle returns $\nabla\varphi_k(\lambda_k, \xi_k)$ (where $\xi = \{\xi_k\}_{k=1}^m$ are independent) under the following σ_φ^2-sub-Gaussian variance condition (for all $k = 1, ..., m$)

$$\mathbb{E}\left[\exp\left(\|\nabla\varphi_k(\lambda_k, \xi_k) - \nabla\varphi_k(\lambda_k)]\|_2^2 \sigma_\varphi^{-2}\right)\right] \leq \exp(1).$$

Problem (D2) can be considered as a particular case of problem (16) with

- $A = \sqrt{W}$
- $\sigma_\Psi^2 = O\left(\lambda_{\max}(W) m \sigma_\varphi^2\right)$

The main observation in the dual approach (see Section 4) is as follows [Scaman et al., 2017]: since $x(A^T y) = \mathbf{x}(\sqrt{W}\mathbf{y})$ we should change the variables as follows

1: $\tilde{\mathbf{y}} := \sqrt{W}\tilde{\mathbf{y}}$
2: $\mathbf{z} := \sqrt{W}\mathbf{z}$
3: $\mathbf{y} := \sqrt{W}\mathbf{y}$

It is obvious that input, output and steps 3–5 of Algorithm 2 are changed such that they can be performed in a decentralized distributed manner. For that we just multiply the corresponding steps by $\sqrt{W}$.

6 Main Results

In this section, we present the rates of convergence for problems (P1) and (D2) (and their stochastic counterparts) in terms of the number of iterations (communication steps) and the number of (parallelized) oracle calls. For the primal problem, we present the results to achieve ε-precision in objective residuals, and for the dual problem we seek to achieve ε-precision in duality gap or primal objective residuals (in smooth strongly convex case). Feasibility constrains are smaller than $\varepsilon/R_{\mathbf{y}}$.

For brevity, we introduce the condition number of the Laplacian matrix W as follows

$$\chi = \frac{\lambda_{\max}(W)}{\lambda^{+}_{\min}(W)}, \tag{20}$$

where $\lambda^{+}_{\min}(W)$ is the minimal positive eigenvalue of W, and $\lambda_{\max}(W)$ is the maximal eigenvalue of W. Now we are ready to present our main results incorporated in multiple tables: Tables 3– 6. These results are obtained by direct substitution of constants from Section 5 to the problems from Sections 3 and 4.

Table 3: The optimal bounds for primal deterministic oracle

	f_k is μ-strongly convex and L-smooth	f_k is L-smooth	f_k is μ-strongly convex	
#communication rounds	$\widetilde{O}\left(\sqrt{\frac{L}{\mu}}\chi\right)$	$\widetilde{O}\left(\sqrt{\frac{LR^2}{\varepsilon}}\chi\right)$	$O\left(\sqrt{\frac{M^2}{\mu\varepsilon}}\chi\right)$	$O\left(\sqrt{\frac{M^2R^2}{\varepsilon^2}}\chi\right)$
#oracle calls of $\nabla f_k(x_k)$ per node k	$\widetilde{O}\left(\sqrt{\frac{L}{\mu}}\right)$	$O\left(\sqrt{\frac{LR^2}{\varepsilon}}\right)$	$O\left(\frac{M^2}{\mu\varepsilon}\right)$	$O\left(\frac{M^2R^2}{\varepsilon^2}\right)$
Algorithm	PSTM, $Q=\mathbb{R}^n$	PSTM, $Q=\mathbb{R}^n$	R-Sliding	Sliding

Note that the bounds on communication steps (rounds) are optimal (up to a logarithmic factor) due to [Arjevani and Shamir, 2015, Scaman et al., 2017, Scaman et al., 2018]. Bounds for the oracle calls per node are probably optimal in the class of methods with optimal number of communication steps (up to a logarithmic factor) in the deterministic case [Allen-Zhu, 2018, Foster et al., 2019, Woodworth et al., 2018] and optimal for the non-smooth stochastic primal oracle and stochastic dual oracle for parallel architecture.[12] For stochastic oracle the

[12] In parallel architecture the bounds on stochastic oracle calls per node of type $\max\{B, D\}$ can be parallel up to B/D processors.

Table 4: The optimal bounds for primal stochastic (unbiased) oracle

	f_k is μ-strongly convex and L-smooth	f_k is L-smooth	f_k is μ-strongly convex	
#communication rounds	$\widetilde{O}\left(\sqrt{\frac{L}{\mu}}\chi\right)$	$\widetilde{O}\left(\sqrt{\frac{LR^2}{\varepsilon}}\chi\right)$	$O\left(\sqrt{\frac{M^2}{\mu\varepsilon}}\chi\right)$	$O\left(\sqrt{\frac{M^2R^2}{\varepsilon^2}}\chi\right)$
#oracle calls of $\nabla f_k(x_k,\xi_k)$ per node k	$\widetilde{O}\left(\max\left\{\frac{\sigma^2}{\mu\varepsilon}, \sqrt{\frac{L}{\mu}}\right\}\right)$	$O\left(\max\left\{\frac{\sigma^2R^2}{\varepsilon^2}, \sqrt{\frac{LR^2}{\varepsilon}}\right\}\right)$	$O\left(\frac{M^2+\sigma^2}{\mu\varepsilon}\right)$	$O\left(\frac{(M^2+\sigma^2)R^2}{\varepsilon^2}\right)$
Algorithm	PBSTM, $Q=\mathbb{R}^n$	PBSTM, $Q=\mathbb{R}^n$	Stochastic R-Sliding	Stochastic Sliding

Table 5: The optimal bounds for dual deterministic oracle

	f_k is μ-strongly convex and L-smooth	f_k is μ-strongly convex
#communication rounds	$\widetilde{O}\left(\sqrt{\frac{L}{\mu}}\chi\right)$	$O\left(\sqrt{\frac{M^2}{\mu\varepsilon}}\chi\right)$
#oracle calls of $\nabla\varphi_k(\lambda_k)$ per node k	$\widetilde{O}\left(\sqrt{\frac{L}{\mu}}\chi\right)$	$O\left(\sqrt{\frac{M^2}{\mu\varepsilon}}\chi\right)$
Algorithm	ROGM-G or STM, $Q=\mathbb{R}^n$	OGM-G or PDSTM

Table 6: The optimal bounds for dual stochastic (unbiased) oracle

	f_k is μ-strongly convex and L-smooth	f_k is μ-strongly convex
#communication rounds	$\widetilde{O}\left(\sqrt{\frac{L}{\mu}}\chi\right)$	$O\left(\sqrt{\frac{M^2}{\mu\varepsilon}}\chi\right)$
#oracle calls of $\nabla\varphi_k(\lambda_k,\xi_k)$ per node k	$\widetilde{O}\left(\max\left\{\frac{M^2\sigma_\varphi^2}{\varepsilon^2}\chi, \sqrt{\frac{L}{\mu}}\chi\right\}\right)$	$O\left(\max\left\{\frac{M^2\sigma_\varphi^2}{\varepsilon^2}\chi, \sqrt{\frac{M^2}{\mu\varepsilon}}\chi\right\}\right)$
Algorithm	R-RRMA+AC-SA^2, $Q=\mathbb{R}^n$	SPDSTM

bounds hold in terms of high probability deviations (we skip the corresponding logarithmic factor).

We emphasize that the difference between centralized (or parallel) estimates and obtained

decentralized ones is not only in the replacement of d by $\tilde{O}(\sqrt{\chi})$ in the smooth cases for the primal oracle and the meaning of L. In the stochastic smooth strongly convex case (one can also consider the convex case) we know that the total number of primal oracle calls [Kulunchakov and Mairal, 2019a, Kulunchakov and Mairal, 2019b, Kulunchakov and Mairal, 2019c, Lan and Zhou, 2018] is

$$\tilde{O}\left(m + \sqrt{m\frac{L}{\mu} + \frac{\sigma^2}{\mu\varepsilon}}\right).$$

This bound is optimal but it uses an incremental oracle and does not imply full parallelization. The best known way to parallelize it is described in [Lan and Zhou, 2018]. For full parallelization one should use a standard accelerated scheme without variance reduction and incremental oracle [Woodworth et al., 2018]. In this case, another bound for the total number of oracle calls occurs, that is

$$\tilde{O}\left(m\sqrt{\frac{L}{\mu} + \frac{\sigma^2}{\mu\varepsilon}}\right).$$

But this bound assumes the natural way of parallelization or centralized distribution of calculations. In the last case for a graph of diameter d with m nodes we have the following number of oracle calls per node

$$\tilde{O}\left(\sqrt{\frac{L}{\mu} + \frac{\sigma^2}{m\mu\varepsilon}}\right) = \tilde{O}\left(\max\left\{\frac{\sigma^2}{m\mu\varepsilon}, \sqrt{\frac{L}{\mu}}\right\}\right)$$

and the following number of communication steps

$$\tilde{O}\left(d\sqrt{\frac{L}{\mu}}\right).$$

For decentralized architecture (see Table 4) the number of oracle calls per node and the number of communication steps are

$$\tilde{O}\left(\max\left\{\frac{\sigma^2}{\mu\varepsilon}, \sqrt{\frac{L}{\mu}}\right\}\right) \text{ and } \tilde{O}\left(\sqrt{\frac{L}{\mu}\chi}\right). \tag{21}$$

respectively. Unfortunately, the factor m is no longer presented in σ^2 in the decentralized case. It is interesting to note, that it is possible to propose such a decentralized distributed algorithm that requires

$$O\left(\frac{\sigma^2}{m\mu\varepsilon}\right)$$

oracle calls per node (stochastic gradients calculations) [Olshevsky et al., 2019a, Olshevsky et al., 2019b]. However, this algorithm is not optimal in terms of communication steps. Moreover, to the best of our knowledge, it is an open question whether is (21) optimal bound in terms of oracle calls (per node) in the class of methods with the optimal number of communication steps.

Note also that the blue bound in Table 6 seems to be rather unexpected at first sight for us. But we hypothesize that this bound is optimal not only in terms of the number of communication steps but also in terms of the number of oracle calls (per node) in the class of methods with the optimal number of communication steps.

The detailed proofs of the statements collected in this paper takes more than 90 pages. These can be found in the arXiv preprint [Gorbunov et al., 2019]:

`https://arxiv.org/pdf/1911.07363.pdf`.

7 Discussion

Below we outline various areas for further work.

- We can expect that the results can be improved by replacing the first-order methods with primal and dual deterministic oracle by tensor methods ($p = 2, 3$) from [Nesterov, 2018a]. However, for the moment we do not know any such results. For the dual approach we also do not know how to use the trick $A = \sqrt{W}$ (see [Scaman et al., 2017]). Here we should take $A = W$, which (with additional increased complexity of auxiliary problem) makes the bounds on communication steps worse. The basic fact in the dual approach is the following. To solve auxiliary problems we have to calculate the values of the form $\nabla_y^p \varphi(Wy)$ on different vectors [Carmon and Duchi, 2016, Nesterov, 2018a, Nesterov, 2018b]. This can be done multiplying W by vectors (communications) and multiplying corresponding (block) diagonal tensor ($\nabla_\lambda^p \varphi(\lambda)|_{\lambda=Wy}$) by vectors (can be distributed among nodes)

- The primal approach in the smooth case can be generalized for the (stochastic inexact) gradient-free oracle. The number of communication steps remains the same. The number of oracle calls becomes $\sim n$ times larger [Gorbunov et al., 2018, Dvurechensky et al., 2017]. In the non-smooth case gradient-free (stochastic) decentralized distributed algorithm was developed in [Beznosikov et al., 2019]

- In [Hendrikx et al., 2019a], [Hendrikx et al., 2019b] asynchronized distributed optimization was considered via dual accelerated (block) coordinate descent algorithms. The primal approach proposed above allows asynchronized generalizations in the smooth

case. For that we should use the (block) coordinate version of `STM` [Dvurechensky et al., 2017] and additional randomization of sum type when $[Wx]_i$ is calculated. This will increase the number of communication steps $\sim \sqrt{n} \div n$ times

- Most of the results of this paper can be generalized to composite problems [Nesterov, 2013]. Perhaps, it is possible to make the next step and try to generalize these results to more general types of models [Stonyakin et al., 2019b, Stonyakin et al., 2019a]

- For smooth convex centralized distributed optimization problems there exists a universal way to accelerate non-accelerated (stochastic, asynchronized etc.) algorithm Catalyst [Lin et al., 2015]. The basic idea is using a non-accelerated centralized distributed algorithm for the inner problem arising at each step of the Catalyst procedure

- Perhaps, it is possible to generalize the primal approach described above on time-varying graphs [Rogozin and Gasnikov, 2019]. Moreover, these generalizations can be done also for the smooth stochastic case

- It seems the result of [Rogozin et al., 2018] can be improved by using mixed communication: many decentralized steps alternate with centralized ones. In this case, one can use non-accelerated distributed decentralized algorithms, which are robust on time-varying graphs [Rogozin et al., 2018], and then accelerate them by using the Catalyst technique [Lin et al., 2015] and centralization. Since the graph is changing we should recalculate spanning tree whenever we apply a centralized step. We expect that this mixed communication will be useful also for tensor schemes in decentralized optimization

- The main scheme in the primal approach is based on the result formulated directly after (14). This result does not depend on convexity of the objective. So it would be interesting to apply this scheme for non-convex distributed optimization problems [Sun and Hong, 2018]

Since the first version of this paper was submitted on arXiv, there appeared alternative explanations (for smooth problems) of the results for the primal deterministic oracle [Fallah et al., 2019, Kovalev et al., 2020, Li and Lin, 2020, Xu et al., 2019, Hendrikx et al., 2020a] and the primal stochastic oracle (strongly convex case) [Fallah et al., 2019]. Moreover, in [Rogozin et al., 2020, Ye et al., 2020] (see also [Rogozin and Gasnikov, 2019] for the non-accelerated case and [Scaman et al., 2018] for lower bounds) for the primal deterministic oracle (strongly convex case) it was shown that $L := \max_k L_k$ used in this paper can be improved to $L := L_f$, where L_f is the Lipschitz gradient constant of (6) (which can be much smaller [Tang et al., 2019]). The same holds true for μ. More interestingly, we expect that combinations of [Rogozin and Gasnikov, 2019, Rogozin et al., 2020, Ye et al., 2020]

allows to develop accelerated decentralized distributed primal algorithms on time-varying graphs (accelerated in L and ε, but not in χ). Moreover, based on [Rogozin and Gasnikov, 2019, Rogozin et al., 2020, Ye et al., 2020] we may expect that the answer to the open problem posed in the Section 6 is negative. The bound can be decreased by a factor m for a stochastic primal oracle. The reasons for that are almost the same that we have for the $\max_k L_k \to L_f$ improvement.

Based on the recent works [Koloskova et al., 2020, Woodworth et al., 2020b, Woodworth et al., 2020a] we describe below a hypothesis about the optimal bounds in a more general situation. We consider the general sum-type problem (for simplicity we consider the case $Q = \mathbb{R}^n$, but it can be naturally generalized)

$$\min_{x \in \mathbb{R}^n} f(x) := \frac{1}{m} \sum_{k=1}^{m} f_k(x) = \frac{1}{m} \sum_{k=1}^{m} \mathbb{E}[f_k(x, \xi)]. \tag{22}$$

We assume that all $f_k(x, \xi)$ in (22) satisfy

$$\|\nabla f_k(y, \xi) - \nabla f_k(x, \xi)\|_2 \leq L \|y - x\|_2.$$

Also we introduce[13]

$$\bar{\zeta}^2 = \frac{1}{m} \sum_{k=1}^{m} \|\nabla f_k(x^*)\|_2^2$$

and

$$\bar{\sigma}^2 = \frac{1}{m} \sum_{k=1}^{m} \mathbb{E}\left[\|\nabla f_k(x^*, \xi) - \nabla f_k(x^*)\|_2^2\right].$$

Assume now that at each iteration t we may call one time an oracle (that returns an independent realization of $\nabla f_k(x^t, \xi^{t,k})$) at each node and make no more than one communication step with (in general, random) communication matrix $\bar{W}_k$. Moreover, we assume that [Koloskova et al., 2020][14]

$$\mathbb{E}_{\bar{W}_l, \ldots, \bar{W}_{l+\tau}} \left[\left\| (I + \bar{W}_{l+\tau}) \cdot \ldots \cdot (I + \bar{W}_l) x - \frac{1}{m} \mathbf{1}_m \mathbf{1}_m^T x \right\|_2^2 \right] \leq$$

[13] Note that nowadays it's quite popular to obtain estimates on rate of convergence that depend on the constants defined at the solution point x^* [Stonyakin et al., 2020].

[14] In [Koloskova et al., 2020] it was also assumed that symmetric matrix $\bar{W}$ determined by doubly stochastic matrix P: $\bar{W} = P - I$, where I – unit matrix. So do we. But below, when we generalize the results of [Koloskova et al., 2020] on a large class of algorithms that assumes more than one communication on one iteration, we may consider $\bar{W}$ to be the same as in section 5 and χ is determined in (20) style.

$$\leq (1-\chi^{-1})\left\|x-\frac{1}{m}\mathbf{1}_m\mathbf{1}_m^T x\right\|_2^2.$$

For the class of algorithms described above the required number of iterations N to achieve an accuracy $\mathbb{E}[f(x^N)]-f(x^*)\leq\varepsilon$ for the best known (for the moment) algorithms is

$$\tilde{O}\left(\left(\frac{LR^2}{\varepsilon}\right)^{\alpha}+\frac{\bar{\sigma}^2R^2}{m\varepsilon^2}+\frac{\sqrt{L}R^2\left(\bar{\zeta}\chi\tau+\bar{\sigma}\cdot(\chi\tau)^{\beta}\right)}{\varepsilon^{3/2}}(1-\chi^{-1})^{I[\tau=1]}+\left(\frac{\chi LR^2}{\varepsilon}\right)^{\alpha}\tau\right),$$

where $\beta\in[0,1]$, $\alpha=1$ or $1/2$, here I is a function such that $I[\text{true}]=1$, $I[\text{false}]=0$, and when f_k is μ-strongly convex

$$\tilde{O}\left(\left(\frac{L}{\mu}\right)^{\alpha}+\frac{\bar{\sigma}^2}{m\mu\varepsilon}+\frac{\sqrt{L}\left(\bar{\zeta}\chi\tau+\bar{\sigma}\cdot(\chi\tau)^{\beta}\right)}{\mu\sqrt{\varepsilon}}(1-\chi^{-1})^{I[\tau=1]}+\left(\frac{\chi L}{\mu}\right)^{\alpha}\tau\right).$$

This bound with $\beta=1,\alpha=1$ (non-accelerated case) was obtained in [Koloskova et al., 2020] for simple (local) decentralized SGD. Roughly speaking, $\beta=0$ corresponds to the lower bound for this algorithm [Koloskova et al., 2020, Yuan and Ma, 2020, Woodworth et al., 2020b, Woodworth et al., 2020a]. In [Karimireddy et al., 2019] by using a variance reduction trick in federated learning architecture ($\bar{W}_{p\tilde{\tau}+q}\equiv 0$ for $q=1,...,\tilde{\tau}-1$ and $\bar{W}_{p\tilde{\tau}}$ is a full communication matrix, hence $\chi=1$, $\tau=\tilde{\tau}$), the SCAFFOLD algorithm was proposed with $\alpha=1$ and (under some additional assumptions) without middle (blue) terms. Also in the federated learning architecture in case $f_1=...=f_m$ it seems that middle (blue) terms and the factor τ in the last terms of the described bounds can be eliminated under some natural additional assumptions [Woodworth et al., 2020b]. Here we assume at least one communication. If there is exactly one communication it will take place at the very end. This result means that in a federated learning setup under $f_1=...=f_m$ the frequency of communications τ does not play any significant role, which was previously mentioned in [Godichon-Baggioni and Saadane, 2020]. In centralized architecture it seems that it is possible to obtain additional acceleration of the algorithms mentioned above ($\alpha=1/2$, but the transition $\chi\to\sqrt{\chi}$ in the last terms requires additional assumptions) by using proper accelerated envelopes like Catalyst [Mathieu et al., 2020, Dvinskikh et al., 2020, Ivanova et al., 2019, Kulunchakov and Mairal, 2019c] with the precision $\varepsilon'=\tilde{O}\left(\varepsilon\sqrt{\max\{\varepsilon/(LR^2),\mu/L\}}\right)$ (in the function) of the solution of auxiliary problem at each outer iteration. It is highly likely that this is true for more general architectures, see examples in [Hendrikx et al., 2020a, Li and Lin, 2020].

If we remove the condition that at each iteration we can only call stochastic oracle one time and make no more than one communication, then the blue term can be eliminated with $\alpha=1/2$. More precisely, the first two terms correspond to the total number of oracle calls

per node and the last term – to the number of communications steps. Up to a factor m in the denominator we have developed these results in the paper, but with $\tau = 1$ and fixed W.[15] We also note that it is an open problem whether it is possible in the general (accelerated) situation (which differs from the [Koloskova et al., 2020]) to determine $\bar{\zeta}^2$ and $\bar{\sigma}^2$ only at point x^*? It seems, that for the current moment of time we have a positive answer only with respect to $\bar{\zeta}^2$.

The following generalizations are related with the case

$$f_k(x) = \frac{1}{l}\sum_{j=1}^{l} f_{kj}(x).$$

In this case with additional assumptions about proximal and dual friendly f_k it is possible to reduce worth case constant L to the ¡¡average¿¿ one [Hendrikx et al., 2020b]. In [Hendrikx et al., 2020b] this looks like a variance reduction acceleration, but the nature of the effect (also explained in [Hendrikx et al., 2020b]) based on coordinate descent acceleration [Nesterov and Stich, 2017] for the dual problem formulation. In [Hendrikx et al., 2020a] was proposed a dual-free generalization of [Hendrikx et al., 2020b] with a bit worse oracle complexity estimate. The main idea is to apply a non-accelerated coordinate descent for the dual problem with Bregman divergence determined by the dual function itself. In [Li et al., 2020] an optimal algorithm both for communications steps and oracle calls per node was developed.

Another way to obtain a better result for the required number of communication steps is possible if $\{f_{kj}\}$ have i.i.d. nature (that is typically for data science applications). In this case f_k is statistically similar to f. Based on this fact in centralized architecture, we may use on a master node statistical preconditioned algorithms that can significantly reduce the required number of communications with slaves [Hendrikx et al., 2020c].

Acknowledgment

The work of D. Dvinskikh in Sections 1 and 5 was supported by RFBR 19-31-51001 and in Section 6 was funded by the Russian Science Foundation (project 18-71-10108). The work of A. Gasnikov was supported by the Ministry of Science and Higher Education of the Russian Federation (Goszadaniye) no. 075-00337-20-03, project No. 0714-2020-0005.

We would like to thank F. Bach, P. Dvurechensky, E. Gorbunov, A. Koloskova, A. Kulunchakov, J. Mairal, A. Nemirovski, A. Olshevsky, S. Parsegov, B. Polyak, N. Srebro A. Taylor and C. Uribe for useful discussions.

[15] For time-varying communication graphs in the eneral case for the moment it seems that we should put the factor χ instead of $\sqrt{\chi}$ in the last term [Rogozin and Gasnikov,].

References

[Allen-Zhu, 2017] Allen-Zhu, Z. (2017). Katyusha: The first direct acceleration of stochastic gradient methods. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2017, pages 1200–1205, New York, NY, USA. ACM. arXiv:1603.05953.

[Allen-Zhu, 2018] Allen-Zhu, Z. (2018). How to make the gradients small stochastically: Even faster convex and nonconvex sgd. In *Advances in Neural Information Processing Systems*, pages 1157–1167.

[Allen-Zhu and Hazan, 2016] Allen-Zhu, Z. and Hazan, E. (2016). Optimal black-box reductions between optimization objectives. In *Advances in Neural Information Processing Systems*, pages 1614–1622.

[Anikin et al., 2017] Anikin, A. S., Gasnikov, A. V., Dvurechensky, P. E., Tyurin, A. I., and Chernov, A. V. (2017). Dual approaches to the minimization of strongly convex functionals with a simple structure under affine constraints. *Computational Mathematics and Mathematical Physics*, 57(8):1262–1276.

[Arjevani and Shamir, 2015] Arjevani, Y. and Shamir, O. (2015). Communication complexity of distributed convex learning and optimization. In *Advances in neural information processing systems*, pages 1756–1764.

[Ben-Tal and Nemirovski, 2001] Ben-Tal, A. and Nemirovski, A. (2001). *Lectures on Modern Convex Optimization.* Society for Industrial and Applied Mathematics.

[Bertsekas and Tsitsiklis, 1989] Bertsekas, D. P. and Tsitsiklis, J. N. (1989). *Parallel and distributed computation: numerical methods*, volume 23. Prentice hall Englewood Cliffs, NJ.

[Beznosikov et al., 2019] Beznosikov, A., Gorbunov, E., and Gasnikov, A. (2019). Derivative-free method for decentralized distributed non-smooth optimization. *arXiv preprint arXiv:1911.10645*.

[Byrne, 2014] Byrne, C. L. (2014). *Iterative optimization in inverse problems*. CRC Press.

[Carmon and Duchi, 2016] Carmon, Y. and Duchi, J. C. (2016). Gradient descent efficiently finds the cubic-regularized non-convex newton step. *arXiv preprint arXiv:1612.00547*.

[Chernov et al., 2016] Chernov, A., Dvurechensky, P., and Gasnikov, A. (2016). Fast primal-dual gradient method for strongly convex minimization problems with linear constraints. In Kochetov, Y., Khachay, M., Beresnev, V., Nurminski, E., and Pardalos, P., editors, *Discrete Optimization and Operations Research: 9th International Conference, DOOR 2016, Vladivostok, Russia, September 19-23, 2016, Proceedings*, pages 391–403. Springer International Publishing.

[Cohen et al., 2018] Cohen, M. B., Diakonikolas, J., and Orecchia, L. (2018). On acceleration with noise-corrupted gradients. *arXiv preprint arXiv:1805.12591*.

[d'Aspremont, 2008] d'Aspremont, A. (2008). Smooth optimization with approximate gradient. *SIAM J. on Optimization*, 19(3):1171–1183.

[Devolder, 2013] Devolder, O. (2013). *Exactness, inexactness and stochasticity in first-order methods for large-scale convex optimization*. PhD thesis, ICTEAM and CORE, Université Catholique de Louvain.

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

[Dvinskikh et al., 2019] Dvinskikh, D., Gorbunov, E., Gasnikov, A., Dvurechensky, P., and Uribe, C. A. (2019). On dual approach for distributed stochastic convex optimization over networks. *arXiv preprint arXiv:1903.09844*.

[Dvinskikh et al., 2020] Dvinskikh, D. M., Turin, A. I., Gasnikov, A. V., and Omelchenko, S. S. (2020). Accelerated and non accelerated stochastic gradient descent in model generality. *Matematicheskie Zametki*, 108(4):515–528.

[Dvurechenskii et al., 2018] Dvurechenskii, P., Dvinskikh, D., Gasnikov, A., Uribe, C., and Nedich, A. (2018). Decentralize and randomize: Faster algorithm for wasserstein barycenters. In *Advances in Neural Information Processing Systems*, pages 10760–10770.

[Dvurechensky and Gasnikov, 2016] Dvurechensky, P. and Gasnikov, A. (2016). Stochastic intermediate gradient method for convex problems with stochastic inexact oracle. *Journal of Optimization Theory and Applications*, 171(1):121–145.

[Dvurechensky et al., 2018] Dvurechensky, P., Gasnikov, A., and Lagunovskaya, A. (2018). Parallel algorithms and probability of large deviation for stochastic convex optimization problems. *Numerical Analysis and Applications*, 11(1):33–37.

[Dvurechensky et al., 2017] Dvurechensky, P., Gasnikov, A., and Tiurin, A. (2017). Randomized similar triangles method: A unifying framework for accelerated randomized optimization methods (coordinate descent, directional search, derivative-free method). *arXiv:1707.08486*.

[Fallah et al., 2019] Fallah, A., Gurbuzbalaban, M., Ozdaglar, A., Simsekli, U., and Zhu, L. (2019). Robust distributed accelerated stochastic gradient methods for multi-agent networks. *arXiv preprint arXiv:1910.08701*.

[Foster et al., 2019] Foster, D., Sekhari, A., Shamir, O., Srebro, N., Sridharan, K., and Woodworth, B. (2019). The complexity of making the gradient small in stochastic convex optimization. *arXiv preprint arXiv:1902.04686*.

[Gao and Blumensath, 2017] Gao, Y. and Blumensath, T. (2017). Distributed computation of linear inverse problems with application to computed tomography. *arXiv preprint arXiv:1709.00953*.

[Gasnikov, 2017] Gasnikov, A. (2017). Universal gradient descent. *arXiv preprint arXiv:1711.00394*.

[Gasnikov et al., 2017] Gasnikov, A., Kabanikhin, S., Mohammed, A., and Shishlenin, M. (2017). Convex optimization in hilbert space with applications to inverse problems. *arXiv preprint arXiv:1703.00267*.

[Gasnikov and Nesterov, 2018] Gasnikov, A. V. and Nesterov, Y. E. (2018). Universal method for stochastic composite optimization problems. *Computational Mathematics and Mathematical Physics*, 58(1):48–64.

[Ghadimi and Lan, 2013] Ghadimi, S. and Lan, G. (2013). Stochastic first- and zeroth-order methods for nonconvex stochastic programming. *SIAM Journal on Optimization*, 23(4):2341–2368. arXiv:1309.5549.

[Godichon-Baggioni and Saadane, 2020] Godichon-Baggioni, A. and Saadane, S. (2020). On the rates of convergence of parallelized averaged stochastic gradient algorithms. *Statistics*, pages 1–18.

[Gorbunov et al., 2019] Gorbunov, E., Dvinskikh, D., and Gasnikov, A. (2019). Optimal decentralized distributed algorithms for stochastic convex optimization. *arXiv preprint arXiv:1911.07363*.

[Gorbunov et al., 2018] Gorbunov, E., Dvurechensky, P., and Gasnikov, A. (2018). An accelerated method for derivative-free smooth stochastic convex optimization. *arXiv preprint arXiv:1802.09022*.

[Guigues et al., 2017] Guigues, V., Juditsky, A., and Nemirovski, A. (2017). Non-asymptotic confidence bounds for the optimal value of a stochastic program. *Optimization Methods and Software*, 32(5):1033–1058.

[Hastie et al., 2001] Hastie, T., Tibshirani, R., and Friedman, J. (2001). *The Elements of Statistical Learning*. Springer Series in Statistics. Springer New York Inc., New York, NY, USA.

[Hendrikx et al., 2018] Hendrikx, H., Bach, F., and Massoulié, L. (2018). Accelerated decentralized optimization with local updates for smooth and strongly convex objectives. *arXiv preprint arXiv:1810.02660*.

[Hendrikx et al., 2019a] Hendrikx, H., Bach, F., and Massoulie, L. (2019a). An accelerated decentralized stochastic proximal algorithm for finite sums. *arXiv preprint arXiv:1905.11394*.

[Hendrikx et al., 2019b] Hendrikx, H., Bach, F., and Massoulié, L. (2019b). Asynchronous accelerated proximal stochastic gradient for strongly convex distributed finite sums. *arXiv preprint arXiv:1901.09865*.

[Hendrikx et al., 2020a] Hendrikx, H., Bach, F., and Massoulié, L. (2020a). Dual-free stochastic decentralized optimization with variance reduction. *arXiv preprint arXiv:2006.14384*.

[Hendrikx et al., 2020b] Hendrikx, H., Bach, F., and Massoulie, L. (2020b). An optimal algorithm for decentralized finite sum optimization. *arXiv preprint arXiv:2005.10675*.

[Hendrikx et al., 2020c] Hendrikx, H., Xiao, L., Bubeck, S., Bach, F., and Massoulie, L. (2020c). Statistically preconditioned accelerated gradient method for distributed optimization. *arXiv preprint arXiv:2002.10726*.

[Ivanova et al., 2019] Ivanova, A., Grishchenko, D., Gasnikov, A., and Shulgin, E. (2019). Adaptive catalyst for smooth convex optimization. *arXiv preprint arXiv:1911.11271*.

[Jin et al., 2019] Jin, C., Netrapalli, P., Ge, R., Kakade, S. M., and Jordan, M. I. (2019). A short note on concentration inequalities for random vectors with subgaussian norm. *arXiv preprint arXiv:1902.03736*.

[Juditsky and Nemirovski, 2012] Juditsky, A. and Nemirovski, A. (2012). First order methods for non-smooth convex large-scale optimization, i: General purpose methods. In Suvrit Sra, Sebastian Nowozin, S. W., editor, *Optimization for Machine Learning*, pages 121–184. Cambridge, MA: MIT Press.

[Kakade et al., 2009] Kakade, S., Shalev-Shwartz, S., and Tewari, A. (2009). On the duality of strong convexity and strong smoothness: Learning applications and matrix regularization. *Unpublished Manuscript, http://ttic. uchicago. edu/shai/papers/KakadeShalevTewari09.pdf*, 2(1).

[Karimireddy et al., 2019] Karimireddy, S. P., Kale, S., Mohri, M., Reddi, S. J., Stich, S. U., and Suresh, A. T. (2019). Scaffold: Stochastic controlled averaging for federated learning. *arXiv preprint arXiv:1910.06378*.

[Kibardin, 1979] Kibardin, V. (1979). Decomposition into functions in the minimization problem. *Avtomatika i Telemekhanika*, (9):66–79.

[Kim and Fessler, 2018] Kim, D. and Fessler, J. A. (2018). Optimizing the efficiency of first-order methods for decreasing the gradient of smooth convex functions. *arXiv preprint arXiv:1803.06600*.

[Koloskova et al., 2020] Koloskova, A., Loizou, N., Boreiri, S., Jaggi, M., and Stich, S. U. (2020). A unified theory of decentralized sgd with changing topology and local updates. *ICML 2020 (accepted), arXiv preprint arXiv:2003.10422*.

[Kovalev et al., 2020] Kovalev, D., Salim, A., and Richtarik, P. (2020). Optimal and practical algorithms for smooth and strongly convex decentralized optimization. *arXiv preprint arXiv:2006.11773*.

[Kulunchakov and Mairal, 2019a] Kulunchakov, A. and Mairal, J. (2019a). Estimate sequences for stochastic composite optimization: Variance reduction, acceleration, and robustness to noise. *arXiv preprint arXiv:1901.08788*.

[Kulunchakov and Mairal, 2019b] Kulunchakov, A. and Mairal, J. (2019b). Estimate sequences for variance-reduced stochastic composite optimization. *arXiv preprint arXiv:1905.02374*.

[Kulunchakov and Mairal, 2019c] Kulunchakov, A. and Mairal, J. (2019c). A generic acceleration framework for stochastic composite optimization. *arXiv preprint arXiv:1906.01164*.

[Lan, 2016] Lan, G. (2016). Gradient sliding for composite optimization. *Mathematical Programming*, 159(1):201–235.

[Lan, 2019] Lan, G. (2019). Lectures on optimization methods for machine learning. *e-print*.

[Lan et al., 2017] Lan, G., Lee, S., and Zhou, Y. (2017). Communication-efficient algorithms for decentralized and stochastic optimization. *Mathematical Programming*, pages 1–48.

[Lan and Zhou, 2017] Lan, G. and Zhou, Y. (2017). An optimal randomized incremental gradient method. *Mathematical Programming*.

[Lan and Zhou, 2018] Lan, G. and Zhou, Y. (2018). Random gradient extrapolation for distributed and stochastic optimization. *SIAM Journal on Optimization*, 28(4):2753–2782.

[Li et al., 2018] Li, H., Fang, C., Yin, W., and Lin, Z. (2018). A sharp convergence rate analysis for distributed accelerated gradient methods. *arXiv preprint arXiv:1810.01053*.

[Li and Lin, 2020] Li, H. and Lin, Z. (2020). Revisiting extra for smooth distributed optimization. *arXiv preprint arXiv:2002.10110*.

[Li et al., 2020] Li, H., Lin, Z., and Fang, Y. (2020). Optimal accelerated variance reduced extra and diging for strongly convex and smooth decentralized optimization. *arXiv preprint arXiv:2009.04373*.

[Lin et al., 2015] Lin, H., Mairal, J., and Harchaoui, Z. (2015). A universal catalyst for first-order optimization. In *Proceedings of the 28th International Conference on Neural Information Processing Systems*, NIPS'15, pages 3384–3392, Cambridge, MA, USA. MIT Press.

[Mathieu et al., 2020] Mathieu, B., Adrien, T., and Francis, B. (2020). Principled analyses and design of first-order methods with proximal inexact proximal operator. *arXiv preprint arXiv:2006.06041*.

[McMahan et al., 2016] McMahan, H. B., Moore, E., Ramage, D., Hampson, S., et al. (2016). Communication-efficient learning of deep networks from decentralized data. *arXiv preprint arXiv:1602.05629*.

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

[Nedić et al., 2018] Nedić, A., Olshevsky, A., and Uribe, C. A. (2018). Graph-theoretic analysis of belief system dynamics under logic constraints. *arXiv preprint arXiv:1810.02456*.

[Nemirovski et al., 2010] Nemirovski, A., Onn, S., and Rothblum, U. G. (2010). Accuracy certificates for computational problems with convex structure. *Mathematics of Operations Research*, 35(1):52–78.

[Nesterov, 2005] Nesterov, Y. (2005). Smooth minimization of non-smooth functions. *Mathematical Programming*, 103(1):127–152.

[Nesterov, 2009] Nesterov, Y. (2009). Primal-dual subgradient methods for convex problems. *Mathematical programming*, 120(1):221–259.

[Nesterov, 2010] Nesterov, Y. (2010). *Introduction to Convex Optimization*. Moscow, MCCME.

[Nesterov, 2012] Nesterov, Y. (2012). How to make the gradients small. *Optima*, 88:10–11.

[Nesterov, 2013] Nesterov, Y. (2013). Gradient methods for minimizing composite functions. *Mathematical Programming*, 140(1):125–161. First appeared in 2007 as CORE discussion paper 2007/76.

[Nesterov, 2015] Nesterov, Y. (2015). Universal gradient methods for convex optimization problems. *Mathematical Programming*, 152(1):381–404.

[Nesterov, 2018a] Nesterov, Y. (2018a). Implementable tensor methods in unconstrained convex optimization. Technical report, CORE UCL. CORE Discussion Paper 2018/05.

[Nesterov, 2018b] Nesterov, Y. (2018b). *Lectures on convex optimization*, volume 137. Springer.

[Nesterov and Stich, 2017] Nesterov, Y. and Stich, S. U. (2017). Efficiency of the accelerated coordinate descent method on structured optimization problems. *SIAM Journal on Optimization*, 27(1):110–123. First presented in May 2015 http://www.mathnet.ru:8080/PresentFiles/11909/7_nesterov.pdf.

[Olshevsky et al., 2019a] Olshevsky, A., Paschalidis, I. C., and Pu, S. (2019a). Asymptotic network independence in distributed optimization for machine learning. *arXiv preprint arXiv:1906.12345*.

[Olshevsky et al., 2019b] Olshevsky, A., Paschalidis, I. C., and Pu, S. (2019b). A non-asymptotic analysis of network independence for distributed stochastic gradient descent. *arXiv preprint arXiv:1906.02702*.

[Poljak, 1981] Poljak, B. (1981). Iterative algorithms for singular minimization problems. In *Nonlinear Programming 4*, pages 147–166. Elsevier.

[Polyak, 1987] Polyak, B. (1987). *Introduction to Optimization*. New York, Optimization Software.

[Rockafellar, 2015] Rockafellar, R. T. (2015). *Convex analysis*. Princeton university press.

[Rogozin and Gasnikov,] Rogozin, A. and Gasnikov, A. Penalty-based method for decentralized optimization over time-varying graphs. In *Optimization and Applications: 11th International Conference, OPTIMA 2020, Moscow, Russia, September 28–October 2, 2020, Proceedings*, pages 239–256. Springer Nature.

[Rogozin and Gasnikov, 2019] Rogozin, A. and Gasnikov, A. (2019). Projected gradient method for decentralized optimization over time-varying networks. *arXiv preprint arXiv:1911.08527*.

[Rogozin et al., 2020] Rogozin, A., Lukoshkin, V., Gasnikov, A., Kovalev, D., and Shulgin, E. (2020). Towards accelerated rates for distributed optimization over time-varying networks. *arXiv preprint arXiv:2009.11069*.

[Rogozin et al., 2018] Rogozin, A., Uribe, C. A., Gasnikov, A., Malkovsky, N., and Nedić, A. (2018). Optimal distributed optimization on slowly time-varying graphs. *arXiv preprint arXiv:1805.06045*.

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

[Scaman et al., 2018] Scaman, K., Bach, F., Bubeck, S., Massoulié, L., and Lee, Y. T. (2018). Optimal algorithms for non-smooth distributed optimization in networks. In *Advances in Neural Information Processing Systems*, pages 2745–2754.

[Shalev-Shwartz and Ben-David, 2014] Shalev-Shwartz, S. and Ben-David, S. (2014). *Understanding machine learning: From theory to algorithms*. Cambridge university press.

[Shalev-Shwartz et al., 2009] Shalev-Shwartz, S., Shamir, O., Srebro, N., and Sridharan, K. (2009). Stochastic convex optimization. In *COLT*.

[Shalev-Shwartz and Zhang, 2014] Shalev-Shwartz, S. and Zhang, T. (2014). Accelerated proximal stochastic dual coordinate ascent for regularized loss minimization. In Xing, E. P. and Jebara, T., editors, *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 64–72, Bejing, China. PMLR. First appeared in arXiv:1309.2375.

[Shapiro et al., 2009] Shapiro, A., Dentcheva, D., and Ruszczynski, A. (2009). *Lectures on Stochastic Programming*. Society for Industrial and Applied Mathematics.

[Spokoiny et al., 2012] Spokoiny, V. et al. (2012). Parametric estimation. finite sample theory. *The Annals of Statistics*, 40(6):2877–2909.

[Stonyakin et al., 2019a] Stonyakin, F., Gasnikov, A., Tyurin, A., Pasechnyuk, D., Agafonov, A., Dvurechensky, P., Dvinskikh, D., and Piskunova, V. (2019a). Inexact model: A framework for optimization and variational inequalities. *arXiv preprint arXiv:1902.00990*.

[Stonyakin et al., 2019b] Stonyakin, F. S., Dvinskikh, D., Dvurechensky, P., Kroshnin, A., Kuznetsova, O., Agafonov, A., Gasnikov, A., Tyurin, A., Uribe, C. A., Pasechnyuk, D., et al. (2019b). Gradient methods for problems with inexact model of the objective. In *International Conference on Mathematical Optimization Theory and Operations Research*, pages 97–114. Springer.

[Stonyakin et al., 2020] Stonyakin, F. S., Stepanov, A. N., Gasnikov, A. V., and Titov, A. A. (2020). Mirror descent for constrained optimization problems with large subgradient values of functional constraints. *Computer Research and Modelling*, 12(2):301–317.

[Sun and Hong, 2018] Sun, H. and Hong, M. (2018). Distributed non-convex first-order optimization and information processing: Lower complexity bounds and rate optimal algorithms. In *2018 52nd Asilomar Conference on Signals, Systems, and Computers*, pages 38–42. IEEE.

[Tang et al., 2019] Tang, J., Egiazarian, K., Golbabaee, M., and Davies, M. (2019). The practicality of stochastic optimization in imaging inverse problems. *arXiv preprint arXiv:1910.10100*.

[Uribe et al., 2018] Uribe, C. A., Dvinskikh, D., Dvurechensky, P., Gasnikov, A., and Nedić, A. (2018). Distributed computation of Wasserstein barycenters over networks. In *2018 IEEE 57th Annual Conference on Decision and Control (CDC)*. Accepted, arXiv:1803.02933.

[Uribe et al., 2020] Uribe, C. A., Lee, S., Gasnikov, A., and Nedić, A. (2020). A dual approach for optimal algorithms in distributed optimization over networks. *Optimization Methods and Software*, pages 1–40.

[Vogel, 2002] Vogel, C. R. (2002). *Computational methods for inverse problems*, volume 23. Siam.

[Woodworth et al., 2020a] Woodworth, B., Patel, K. K., and Srebro, N. (2020a). Minibatch vs local sgd for heterogeneous distributed learning. *arXiv preprint arXiv:2006.04735*.

[Woodworth et al., 2020b] Woodworth, B., Patel, K. K., Stich, S. U., Dai, Z., Bullins, B., McMahan, H. B., Shamir, O., and Srebro, N. (2020b). Is local sgd better than minibatch sgd? *arXiv preprint arXiv:2002.07839*.

[Woodworth and Srebro, 2016] Woodworth, B. E. and Srebro, N. (2016). Tight complexity bounds for optimizing composite objectives. In *Advances in neural information processing systems*, pages 3639–3647.

[Woodworth et al., 2018] Woodworth, B. E., Wang, J., Smith, A., McMahan, B., and Srebro, N. (2018). Graph oracle models, lower bounds, and gaps for parallel stochastic optimization. In *Advances in Neural Information Processing Systems*, pages 8505–8515.

[Xu et al., 2019] Xu, J., Tian, Y., Sun, Y., and Scutari, G. (2019). Accelerated primal-dual algorithms for distributed smooth convex optimization over networks. *arXiv preprint arXiv:1910.10666*.

[Ye et al., 2020] Ye, H., Luo, L., Zhou, Z., and Zhang, T. (2020). Multi-consensus decentralized accelerated gradient descent. *arXiv preprint arXiv:2005.00797*.

[Ye et al., 2019] Ye, N., Roosta-Khorasani, F., and Cui, T. (2019). Optimization methods for inverse problems. In *2017 MATRIX Annals*, pages 121–140. Springer.

[Yuan and Ma, 2020] Yuan, H. and Ma, T. (2020). Federated accelerated stochastic gradient descent. *arXiv preprint arXiv:2006.08950*.