Decentralized and Parallelized 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 in the class of methods with optimal number of communications steps takes place only up to a logarithmic factor and the notion of smoothness (the worst case vs the average one). We also show that using mini-batching technique all proposed methods with stochastic oracle can be additionally parallelized on each node.

I. INTRODUCTION

Consider stochastic convex optimization problem

\[ f(x) = \mathbb{E}[f(x, \xi)] \to \min_{x \in \mathbb{R}^n} f(x). \]  

Such kind of problems arise in many applications of data science [56], [59] and mathematical statistics [60]. To solve this problem with average precision \( \varepsilon \) in function value one can use stochastic gradient (mirror) descent [30] with

\[ \min \left\{ O \left( \frac{M^2 R^2}{\varepsilon^2} \right), O \left( \frac{M^2}{\mu \varepsilon} \right) \right\} \]  

number of calculations of unbiased stochastic subgradients \( \nabla f(x, \xi) \) \( \mathbb{E}[\|\nabla f(x, \xi)\|_2^2] \leq M^2 \). Here \( R = \|x^0 - x^*\|_2 \) is the Euclidean distance from the starting point \( x^0 \) to the solution \( x^* \) and \( \mu \) is the constant of strong convexity of \( f \) in (1). Unfortunately, generally in this case we can parallelize calculations at most of \( \tilde{O}(1) \) processors [17]. 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 we can reduce (2) to

\[ \min \left\{ O \left( \sqrt{\frac{L R^2}{\varepsilon}} \right) + O \left( \frac{\sigma^2 R^2}{\varepsilon^2} \right), O \left( \sqrt{\frac{L}{\mu \ln \left( \frac{\mu R^2}{\varepsilon} \right)} \right) + O \left( \frac{\sigma^2}{\mu \varepsilon} \right) \right\} \]  

by using batch parallelization [11], [15], [21], [22]. Note that we can parallelize calculations at no more than

\[ O \left( \frac{\sigma^2 R^2}{\varepsilon} \right) \]  

processors (depends on where is a minimum in (3)), that is much better. Since the result cannot be improved [68], it is the best possible way (in general) to solve (1) by using parallel architecture in online context.

For many reasons, in some situations it can impossible to organize model-based request for calculation of stochastic gradient \( \nabla f(x(k), \xi(k)) \) in online regime. Typically, in machine learning applications [25], [56] instead of online access to \( \{ \nabla f(x(k), \xi(k)) \}_{k=1}^{m} \) we have offline access. This means that the set of functions \( \{ f(x, \xi(k)) \}_{k=1}^{m} \) are stored somewhere 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, from [24], [57], [59] it is known that with high probability the exact solution of problem

\[ \hat{f}(x) = \frac{1}{m} \sum_{k=1}^{m} f(x, \xi(k)) \to \min_{x \in \mathbb{R}^n} f(x). \]  

is an \( \varepsilon \)-solution (in function) of problem (1) if

\[ m = \min \left\{ \tilde{O} \left( \frac{n M^2 R^2}{\varepsilon^2} \right), \tilde{O} \left( \frac{M^2}{\mu \varepsilon} \right) \right\}. \]
Moreover, we cannot typically find the exact solution of (4) but in $\mu$-strongly convex (or regularized) case it is sufficient to solve (4) with accuracy $O(\mu^{-2}M^{-2})$ [57].

To solve (4) in offline context we have to store $\{f(x, \xi^k)\}_{k=1}^m$ somehow in the memory. As we have mentioned above, $m$ can be large. That is why centralized distributed architecture is often more preferable in this context [6]. In general case centralized architecture is based on communication network. We build spanning tree for this network with the origin (root) to be a master-node [54]. Denote by $d$ the distance between the origin and farthest leaf. Estimate (4) with $\sigma=0$ and $L$ corresponds to $\bar{f}$ (see (4)) can be carry out for the problem (5) if we have $m$-node network and we are interested in oracle complexity per node. The number of communications steps will be $d$ times larger. But $m$ can be too large! If we have only $q \ll m$ nodes, then we split the data $\{f(x, \xi^k)\}_{k=1}^m$ on $q$ blocks for $l=m/q$ terms in each block. If $l$ is too large by itself on can reformulate (4) in the following way [42]

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

(5)

where $f_k(x, \eta^k) = f(x, \xi^{kl+i} + \eta^k)$ and $\eta^k = i$ with probability $1/l$, $i = 1, \ldots, l$. Representation (5) allows to use bound (3) in stochastic case in a parallel manner at each node. The main conclusions here remain the same as before. The number of oracle calls per node corresponds to (3) and the number of communications steps is also $d$ times larger than (3) with $\sigma=0$. And these bounds seems to be not improvable [5], [55], [68].

Unfortunately, centralized architecture has synchronization drawback and high requirement to master node [54]. To eliminate this drawbacks to some extent one should use decentralized distributed architecture [6], [33] which relies on two basic principles: every node communicates only with all its neighbours, and all communications are performed simultaneously. The main difference here is simple strategy of communications: each node communicates only with all available direct neighbours. This architecture is more robust. In particular it can be applied to time-varying (wireless) communication networks [53].

One of the purposes of this paper is a justification of a transition from optimal centralized distributed complexity bounds for (4) and (5) in smooth case to decentralized ones by replacing $d$ (the height of spanning tree) to $\sqrt{\Delta}$ (square root of condition number of the Laplacian communication matrix describing the network), by replacing the average $L$ to the worse one and by replacing variance of $f$ by the variance of $f_k$, that can be $\sqrt{m}$ times larger.

By using different smoothing techniques [3], [44], [55], we may reduce non-smooth case to smooth one with $L \sim 1/\varepsilon$. This allows to reduce the complexity estimate (2) by using (3). However, in general this reduction complicates the complexity of oracle calls. Thus, we can only improve the communication steps (rounds) bound that corresponds to (3) (up to a $\sqrt{\Delta}$ factor) with $L \sim 1/\varepsilon$ and $\sigma^2 = 0$. Could we conserve the bound (3) for standard (old one) conception of oracle calls per node in decentralized approach by improving the number of communications steps? Up to the replacement of the average $M$ to the worst one the answer is positive [37], [55]. Below in the paper, we simplify the approaches proposed in these papers to prove this result.

In different applications (e.g. Wasserstein barycenter calculation problem [13], [14], [64]) we have to solve

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

(6)

where $f_k(x)$ has Fenchel–Legendre representation $f_k(x) = \max_y \{ \langle x, y \rangle - \varphi_k(y) \}$ with convex $\varphi_k(y)$. 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 $\nabla \varphi_k(y)$ [13], [14]. In these cases we will use dual (stochastic) oracle.

The other purpose of this paper is to develop the optimal decentralized distributed algorithms with dual (stochastic) oracle for strongly convex objective in (6).

3Problems (4), (5) have specific sum-type structure. Roughly speaking, this structure allows to solve these problems much faster on one machine. For example, by using some incremental algorithms [1], [38], [41], [67] one can solve (4) $\sqrt{m}$ times cheaper in terms of number of oracle calls, but not in terms of the number of iterations = communications steps. Unfortunately, this results does not assume any parallelization as a consequence there appear troubles with decentralized generalizations. Note, that for this problem in asynchronized mode (only two nodes, choose at random, can communicate at each step) one can obtain such ($\sim \sqrt{m}$) an acceleration for star type communications network [39]. Moreover, ‘dual’ analogue of this acceleration have been recently proposed for (4) [26] and (5) [27], [28] with arbitrary communications networks.

4In deterministic case this was partially done in [40]. Note, that the announced results are also not improvable in terms of communications steps (rounds) [5], [54].

5Note, that $\sqrt{\Delta} \geq d$ and typically $\sqrt{\Delta} \leq nd$ [43]. The last bound corresponds to stars topology [20] (the most simple centralized type architecture). In many interesting cases $\sqrt{\Delta} = O(d)$ [43], [54].

6For constant $\mu$ here everywhere below we may use a trick from [54] to save it at an average level.

7For example, this will happen in case when we have independent noise at each $f_k$. Note also, that in this case in decentralized distributed optimization one can improve the dependence on the variance and win $\sqrt{m}$ factor [50], [51]. But, this is possible due to the worse estimate for the number of communications steps.

8Note that such tricks sometimes allow to obtain optimal (in terms of dependence of $c$) communications rounds estimates [5], [55].
The approach is based on dual reformulation of \([6]\) [54]. The optimal algorithm for non-strictly convex dual function with stochastic oracle was recently proposed in [13]. To propose an optimal method with stochastic dual oracle in strongly convex case we use recent work [19]. Note, that rather unexpected result here is that we cannot improve (up to a logarithmic factor) the bound for the dual stochastic gradient calculations in comparison with not strongly convex dual objective.

We also notice that initially we were motivated to investigate dual oracle not only by the applications from [13], [14], [64]. We try to find a simple explanation for optimal communications steps bounds [5], [55] in non-smooth case. One of the way to do it is Nesterov’s dual smoothing technique [44] that builds a bridge to the notion of dual oracle. This plan was partially (in deterministic case) implemented in [54], [65], [66]. Here we generalize the results of these works for stochastic dual oracle.

Paper organization

The paper is organised as follows. In Section II for stochastic convex optimization problems we propose optimal stochastic (parallelized) accelerated gradient methods. In Sections III and IV we apply the results of Section II to stochastic convex optimization problems with affine type 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 \(Ax\) (correspond to communications). In Sections III and IV we are focusing on primal methods, in Section VI we focusing on dual ones. Section VI is responsible for the distributed primal and dual formulation of finite-sum minimization problem and for the representing the algorithms in distributed fashion. In Section VII we incorporate proposed distributed decentralized method on purpose getting optimal bounds for finite-sum minimization problem using primal or dual oracle. Finally, we discuss the future work and possible extensions. We notice that all proposed methods are also optimal in terms of communications steps and at the same time optimal in the class of methods with optimal number of communications steps (up to poly logarithmic multipliers and interpretations of smoothness constants) in terms of (parallelized stochastic) primal/dual oracle calls.

II. Stochastic Convex Optimization

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

\[
\min_{x \in Q \subseteq \mathbb{R}^n} f(x),
\]

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 deterministic oracle this concepts can be identified. By the first-order oracle, we will mean a black-box model that for the given input \(x \in Q\), returns the vector \(\nabla f(x)\).

Algorithm 1 Similar Triangles Methods STM\((L,\mu, x_0)\), the case when \(Q = \mathbb{R}^n\)

Input: \(\tilde{x}^0 = x^0 = x^0\), number of iterations \(N\), \(\alpha_0 = A_0 = 0\)
1: for \(k = 0, \ldots, N\) do
2: \(\bar{x}^{k+1} = \frac{1}{2L} \bar{A}^2_k + \frac{1}{2A_k} \bar{A}_k^2 + \frac{A_k(1 + A_k \mu)}{L} \bar{A}_{k+1} = \bar{A}_k + \alpha_{k+1}\)
3: \(x^{k+1} = (A_k x^k + \alpha_{k+1} z^k) / A_{k+1}\)
4: \(z^{k+1} = x^k - \alpha_{k+1} (\nabla f(x^{k+1}) - \mu z^{k+1})\)
5: \(x^{k+1} = (A_k x^k + \alpha_{k+1} z^{k+1}) / A_{k+1}\)
Output: \(x^N\)

Accelerated gradient methods (e.g. Algorithm I) STM (21), [49], see also [36]) allow to obtain the optimal number of iterations and number of oracle calls for problem (7), as described in Table I where \(R = \|x^0 - x^*\|_2\) is the

\[
\min_{x \in Q \subseteq \mathbb{R}^n} \sum_{i=0}^{k+1} \alpha_i \left\{ \left(\nabla f(x^i), z - x^i\right) + h(z) + \frac{\mu}{2} \|z - x^i\|^2_2 \right\} + \frac{1}{2} \|z - x^0\|^2_2.
\]

If \(h(x)\) has \(L_h\)-Lipschitz gradient in 2-norm then due to Theorem 9 [61] and Theorem 19 [62] it is sufficient to solve auxiliary problem with accuracy (in terms of function value)

\[
O \left( \frac{(\alpha_{k+1} + 1)^2 (A_{k+1} + 1)}{(A_{k+1} L_h R^2)^2} \right) \geq O \left( \frac{\epsilon^3}{L L_h R^4} \right),
\]

where \(\epsilon\) is a desired accuracy (in function value) for initial problem (7).

If \(\mu = 0\) one can also generalize this step for non-Euclidean case. Then using restarts [21] one can generalize such a method on \(\mu > 0\). Note, that by using restarts with STM\((L,\mu, x^0)\) one can eliminate the gap \(\ln(L R^2/\epsilon) / \ln(\mu R^2/\epsilon)\) between lower bounds and the bounds of STM\((L,\mu, x^0)\) without restarts [21]. The same is true for the stochastic case, see below.

Here and below (see, e.g. Table II) the last two columns can be obtained from the corresponding first columns by choosing \(L = M^2/(2\delta)\), where \(\delta = \epsilon/N\) [21]. This is the idea of universal accelerated methods [47], but with predefined \(L\). Note also, that in Tables II we skip numerical constants.
distance between the solution $x^*$ and starting point $x^0$ (if $x^*$ is not unique we may take here such $x^*$ that is the closest one to $x^0$), and $\varepsilon$ is the desired precision in function value.

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

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

We also say that function $f$ is $\mu$-strongly convex if

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

**TABLE I:** Optimal number of first-order oracle calls (number of iterations $N$)

|                          | $\mu$-strongly convex, $L$-smooth | $L$-smooth | $\mu$-strongly convex, $\|\nabla f(x)\|_2 \leq M$ | $\|\nabla f(x)\|_2 \leq M$ |
|--------------------------|-----------------------------------|------------|-----------------------------------------------|-----------------------------------------------|
| # of iterations          | $\sqrt{\frac{\mu}{L}} \ln \left( \frac{2LR^2}{\varepsilon} \right)$ | $\sqrt{\frac{L^2}{\varepsilon}}$ | $\frac{M^2}{\mu\varepsilon}$ | $\frac{M^2R^2}{\varepsilon^2}$ |
| # of $\nabla f(x)$ oracle calls | $\sqrt{\frac{\mu}{L}} \ln \left( \frac{2LR^2}{\varepsilon} \right)$ | $\sqrt{\frac{L^2}{\varepsilon}}$ | $\frac{M^2}{\mu\varepsilon}$ | $\frac{M^2R^2}{\varepsilon^2}$ |

Generally, iteration complexity is given by the complexity of gradient calculations, which can be hard to compute. 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) $\xi$ corrupting the true gradient. Assume that

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

and

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

Then with probability $\geq 1 - \beta$

$$
f(x^N) - f(x^*) \leq \varepsilon
$$

after

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

iterations of STM, that used approximated gradient (instead of real one $\nabla f(x^k)$)

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

where $\xi_i$ are i.i.d from the same distribution as $\xi$ and batch size

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

11Here and below in such type of assumption (especially in the case when $Q$ is unbounded) instead of $\forall x \in Q$ we may write $\forall x \in Q : \|x - x^*\|_2 \leq 2R - [20]$ (analogously for $y$).

12We notice that unlike [9], [10] where $R$ is a diameter of $Q$, we outperform the bound by $R = \|x^0 - x^*\|_2$ due to typical case when $Q = \mathbb{R}^n$ and it is not a compact set. To obtain such a generalization we have to use advanced recurrent technique to bound $\|x^k - x^*\|_2$ from [13], [23] and chapter 2 [20]. This result can be obtained by the following scheme (for simplicity $\sigma = 0, \mu = 0$).

1. Prove that using inexact gradient $\tilde{\nabla} f(x)$, satisfies for all $x, y$

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

for STM we’ll have [12]

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

2. Due to the

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

show that one can consider $\delta_2 = \delta_2^2/(2L)$ and $L := 2L$ in [13].

3. Show that $\delta_1 = R\delta$ under the assumption $\|x^k - x^*\|_2 \leq R$.

4. In deterministic case show that $R = R$ (see chapter 2 [20], [21]). In stochastic case $R = O(R)$ [13], [23] for STM with the proper batch-size [10].
Moreover the total number of oracle calls\footnote{Oracle calls can be easily and fully parallelized (on \(r_k\) processors) at each iteration. Note, that for \(\nabla f(x, \xi)\) calculations we can reduce variance \(\sigma^2 := O(\sigma^2/r_k)\).} is (this bound is optimal up to a red factors)
\[
\sum_{k=0}^{N} r_k = O(N) + \min \left\{ O \left( \frac{\sigma^2 R^2}{\varepsilon^2} \ln \left( \frac{\sqrt{L R^2 / \varepsilon}}{\beta} \right) \right), O \left( \frac{\sigma^2}{\mu \varepsilon} \ln \left( \frac{LR^2}{\varepsilon} \right) \ln \left( \frac{\sqrt{L / \mu}}{\beta} \right) \right) \right\}.
\]

Such a variant of STM we will further call \(\text{BSTM}(L,\mu,\sigma^2, x^0)\) (batched STM(\(L,\mu,x^0\)).

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 I, where we skip high probability logarithmic multipliers. The number of stochastic oracle calls for this case is shown in Table II. In particular, for the case of non-smooth objective, the stochastic oracle does not yield gains compared to its deterministic counterpart.

**TABLE II: Optimal number of stochastic (unbiased) first-order oracle calls**

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

III. 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 I and II for convex optimization problem\footnote{This is because \(\text{Im} A = \text{Im} A^T = (\text{Ker} A)^\perp\) and \(Q = \mathbb{R}^n\). As a consequence of these facts auxiliary problem is split on two sub problems: 1) minimization on \((\text{Ker} A)^\perp\) of quadratic form with matrix of form \((R^2_{c}/\varepsilon)A^T A + c I\), where \(c\) is some positive constant and \(I\) is identity matrix; 2) minimization on \(\text{Ker} A\) of quadratic form with matrix of form \(c I\). Linear terms don’t play any role in complexity. Complexity of auxiliary problem determines by the worst (reduced on corresponding sub space) conditional number of these two sub problems. It’s obvious that the first one is worse: reduced conditional number \(\lambda_{\text{max}} \left( (R^2_{c}/\varepsilon)A^T A + c I \right) / \lambda_{\text{min}}^+ \left( (R^2_{c}/\varepsilon)A^T A + c I \right) \leq \lambda_{\text{max}}(A^T A) / \lambda_{\text{min}}^+(A^T A)\).}.

\[
f(x) \rightarrow \min_{A x = 0, x \in Q} f(x),
\]

where \(A > 0\) and \(\text{Ker} A \neq \emptyset\). The purpose of this section is to develop such algorithms for (11) that are optimal in terms of the number of \(\nabla f(x)\) calculation and the number of \(A^T A\) calculation. In this section we use Euclidean proximal setup. This is the only section where we significantly use Euclidean prox-structure.

Denote by \(R_y = \|y^*\|_2\) 2-norm of the smallest solution \(y^*\) of dual (up to a sign) problem (15). Solution \(y^*\) is not unique since \(\text{Ker} A \neq \emptyset\). From [37] we have such a bound

\[
R_y^2 \leq \frac{\|\nabla f(x^*)\|^2}{\lambda_{\text{min}}^+(A^T A)}.
\]

The main trick of this section is to use special penalty method to solve (11)

\[
F(x) = f(x) + \frac{R^2_y}{\varepsilon} \|A x\|_2^2 \rightarrow \min_{x \in Q} F(x).
\]

From the remark 4.2 of [20] the following holds: if

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

then

\[
f(x^N) - \min_{x \in Q} f(x) \leq \varepsilon, \|A x^N\|_2 \leq \varepsilon / R_y.
\]
in \( O \left( \sqrt{\frac{\lambda_{\text{max}}(A^T A)}{\lambda_{\text{min}}(A^T A)}} \right) \) times larger. This factor arises because of the complexity of auxiliary problem. These approaches we will call 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 Sliding algorithm [35], [36]. If \( \mu = 0 \) according to [35] this algorithm requires (see Tables 4, 11 for comparison)

\[
O \left( \frac{\lambda_{\text{max}}(A^T A) R_y^2}{\mu \varepsilon^2} \right) A^T A \text{-calculations}
\]

and

\[
O \left( \frac{M^2 R_y^2}{\varepsilon^2} \right) \nabla f(x) \text{-calculations,}
\]

where \( R_y = \|y^* - y\|_2 \).

If instead of \( \nabla f(x) \) we have unbiased \( \nabla f(x, \xi) \) with \( \sigma^2 \)-subgaussian variance [29], i.e.

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

with \( \sigma^2 = O(M^2) \) (for compactness of designation), then the bound on \( A^T A \text{-calculations} \) does not change and the bound on \( \nabla f(x, \xi) \text{-calculations} \) will be the same as it was for \( \nabla f(x) \text{-calculations} \) in deterministic case (up to a logarithmic high-probability deviations factor).

By using restart technique [65] we can generalize this methods for \( \mu \)-strongly convex \( f \):

\[
O \left( \frac{\lambda_{\text{max}}(A^T A) R_y^2}{\mu \varepsilon^2} \right) \ln \left( \frac{\mu R_y^2}{\varepsilon^2} \right) A^T A \text{-calculations}
\]

and

\[
O \left( \frac{M^2}{\mu \varepsilon^2} \right) \nabla f(x) \nabla f(x, \xi) \text{-calculations.}
\]

We will call these approach R-Sliding (Restart Sliding).

IV. DUAL METHODS FOR STOCHASTIC CONVEX OPTIMIZATION WITH AFFINE CONSTRAINTS

Now we assume that we can build a dual problem for\(^{17}\)

\[
f(x) \rightarrow \min_{Ax=0, x \in Q},
\]

where \( \ker A \neq \emptyset \). The dual problem (up to a sign) is as follows

\[
\psi(y) = \varphi(A^T y) = \max_{x \in Q} \{ y, Ax \} = (y, Ax) = \langle y, Ax(A^T y) \rangle = \langle A^T y, x(A^T y) \rangle = \langle A^T y, x(A^T y) \rangle - f(x(A^T y)) \rightarrow \min_y.
\]

If \( f \) is \( \mu \)-strongly convex in 2-norm, then \( \psi \) has \( L_y = \frac{\lambda_{\text{max}}(A^T A)}{\mu} \)-Lipschitz continuous gradient in 2-norm [32], [52].

In this case we can apply STM(\( L_y, \), 0) to (15). Note that due to Demyanov–Danskin’s theorem \( \nabla \psi(y) = Ax(A^T y) \) [52].

Similarly to [4], [8] one can prove that

\[
f(x^N) - f(x^*) = f(x^N) - f(x(A^T y)) \leq f(x^N) + \psi(y^N) = O \left( \frac{L_y R_y^2}{N^2} \right), \|Ax^N\|_2 = O \left( \frac{L_y R_y}{N^2} \right),
\]

where \( R_y = \|y^*\|_2 \) is the radius of solution of (15) which is the smallest in 2-norm, see (12). We will call this approach PDSTM (Primal-Dual STM).

If we have only stochastic (randomized) unbiased model \( \nabla \varphi(\lambda, \xi)_{\lambda = A^T y} = x(A^T y, \xi) \) with \( \sigma^2 \)-subgaussian variance, i.e.

\[
\mathbb{E} \left[ \exp \left( \frac{\|\nabla \varphi(\lambda, \xi) - \nabla \varphi(\lambda)\|_2^2}{\sigma^2} \right) \right] = \mathbb{E} \left[ \exp \left( \frac{\|x(A^T y, \xi) - x(A^T y)\|_2^2}{\sigma^2} \right) \right] \leq \exp(1),
\]

\(^{17}\)We notice that turning to dual problem does not oblige us using dual oracle. Instead we can use primal oracle and Moreau theorem [52] with Fenchel-Legendre representation. This maximization problem can be solved using first-order oracle for function \( f \). But such an approach doesn’t allow to obtain optimal bounds on number of primal first-order oracle calls.

\(^{18}\)In general: \( M^2 \rightarrow M^2 + \sigma^2 \).
from (17) we have that for \(y^k\) to be true it is sufficient to find such \(\hat{y}^k\) with desired accuracy \(\bar{Q}\) 2-norm and converges as follows (for simplicity we skip poly-logarithmic factors and high probability terminology)

\[
\hat{y}^{k+1} = \frac{y^k + \alpha_k z^k}{\bar{Q}^2 + \alpha_k + 1} / \bar{Q}^2 + \alpha_k + 1
\]

Output: \(y^N, x^N = \frac{1}{\bar{Q}^2} \sum_{k=0}^N \alpha_k x(A^T \hat{y}^k)\).

then for BSTM(\(L, \sigma^2, 0\)) where \(\sigma^2 = \lambda_{\max}(A^T A) \sigma^2 \) with probability \(\geq 1 - \beta\) \((16)\) holds true \([13]\). We will call this approach SPDSTM (Stochastic PDSTM).

In case when \(\psi\) in \((15)\) is additionally \(\mu_\psi\)-strongly convex in 2-norm \(\psi^0 + (\text{Ker}A^T)^\perp\) (if \(f\) has \(L\)-Lipschitz gradient in 2-norm and \(Q = \mathbb{R}^n\) then \(\mu_\psi = \lambda_{\min}(A^T A) / L\) \([32, 34]\), where \(\lambda_{\min}(A^T A)\) is the minimal positive eigenvalue of \(A^T A\) we need to use another approach. Because of primal-duality we have to put in STM and its derivatives \(\mu_\psi = 0\) (STM(\(L, 0, \psi^0\)) isn’t primal-dual method when \(\mu_\psi > 0\)). Restarts technique (see, e.g. \([20]\)) also does not work here, because in \((16)\) we have to use in general \(R_y = \|y^0\|^2 + \|y^0 - y^*\|^2\). That is why we take here \(y^0 = 0\). So the main trick here is the following relation \([2, 4, 45]\)

\[
f(x(A^T y)) - f(x^*) \leq \langle \nabla \psi(y), y \rangle = \langle A x(A^T y), y \rangle.
\]

From \((17)\) we have that for

\[
f(x^N) - f(x^*) = f(x(A^T y^N)) - f(x(A^T y^*)) \leq 2\varepsilon, \|A x^N\|^2 \leq \varepsilon / R_y,
\]

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

\[
\|\nabla \psi(y^N)\|^2 \leq \varepsilon / R_y.
\]

Recently, there appear accelerated method with the proper rate of convergence in terms of the norm of the gradient OGM-G \([20, 34]\):

\[
\|\nabla \psi(y^N)\|^2 = O\left(\frac{L\|y^0 - y^*\|^2}{N^2}\right).
\]

After \(N = O(\sqrt{L\mu_\psi})\) iterations of OGM-G we will have

\[
\|\nabla \psi(y^N)\|^2 \leq \frac{1}{2} \|\nabla \psi(y^0)\|^2.
\]

So after \(l = \log_2\left(\|\nabla \psi(y^0)\|^2 / 2R_y / \varepsilon\right)\) restarts \((y^0 := y^N)\) we will have \((13)\). Such an approach we will denote ROGM-G (Restart OGM-G). This approach requires

\[
O\left(\sqrt{\frac{L\mu_\psi}{\ln(\|\nabla \psi(y^0)\|^2 / R_y / \varepsilon)} N^2}\right)
\]

of \(\nabla \psi(y)\) (that is \(Ax(A^T y)\) calculations.

The same result (with replacement \(\ln(\|\nabla \psi(y^0)\|^2 / 2R_y / \varepsilon) \rightarrow \ln\left(2L_\psi R_\psi^2 / \varepsilon^2\right)\) can be obtained by using STM(\(L_\psi, \mu_\psi, 0\)) with desired accuracy \(\varepsilon := \varepsilon^2 / (2L_\psi R_\psi^2)\). This follows from

\[
\frac{1}{2L_\psi} \|\nabla \psi(y^N)\|^2 \leq \psi(y^N) - \psi(y^*).
\]

Now following by \([19]\) (see also \([2]\) in non accelerate, but composite case) consider RRMA+AC-SA\(^2\). This algorithm converges as follows (for simplicity we skip poly-logarithmic factors and high probability terminology)

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

19Since \(\text{Im} A = (\text{Ker}A^T)^\perp\) we will have that all the points \(\hat{y}^k, z^k, y^k\), generated by STM and its derivatives, belong to \(y^0 + (\text{Ker}A^T)^\perp\). That is, from the point of view of estimates this means, that we can consider \(\psi\) to be \(\mu_\psi\)-strongly convex everywhere.

20The key inequality to prove this fact is:

\[
\|y^0 - y^*\|^2 \leq \frac{1}{\mu_\psi^2} \|\nabla \psi(y^0)\|^2.
\]
At each iteration there available $\nabla \psi(y, \xi)$ with subgaussian variance $\sigma_\psi^2$ [29] (see also above). If we use restarts with size of each restart $N = \tilde{O}(\sqrt{L_\psi/\mu_\psi})$ (see above) and use batched gradient [9] with batch size (at $k$-th restart; $\hat{y}^k$ is the output point from the previous restart)

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

then $||\nabla \psi(\hat{y}^l)||_2 \leq \varepsilon/R_y$ 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

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

Note that the same bound take place in non strongly convex case ($\mu_\psi = 0$). From [2], [29] it’s known that this bound can not be improved. But from the Table II (for stochastic primal oracle) we may expect that this bound can be reduced to $\tilde{O}(\sigma_\psi^2/\mu_\psi \varepsilon)$. It seems that for stochastic dual oracle such a reduction is impossible. We will call this approach R-RRMA+AC-SA$^2$ (Restart RRMA+AC-SA$^2$).

V. Decentralized Distributed Optimization

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

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

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 [6]. 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 $\tilde{W} \in \mathbb{R}^{m \times m}$ 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

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

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

From the definition of matrix $\tilde{W}$ it can be easily seen that $\tilde{W}$ establishes the communication of agents and allows only the communication between neighboring nodes. Moreover, due to connectivity of graph $G$ the vector $1_m = (1, ..., 1)^T \in \mathbb{R}^m$ is the unique (up to a scaling factor) eigenvector of $\tilde{W}$ associated with the eigenvalue $\lambda = 0$, that allows us to compactly rewrite the consensus agreement $x_1 = ... = x_m \in \mathbb{R}^n$ as $\tilde{W}x = 0$, moreover, as $\sqrt{\tilde{W}}x = 0$ [54] (to be explained the purpose precisely soon), where $\tilde{W} = \tilde{W} \otimes I_n$ is the Kronecker product of the Laplacian matrix $\tilde{W} \in \mathbb{R}^m$ and the identity matrix $I_n$ and $x = [x_1^T, ..., x_m^T]^T \in \mathbb{R}^{mn}$.

To present the 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 communication matrix $\tilde{W}$ as following

$$F(x) = \frac{1}{m} \sum_{k=1}^{m} f_k(x_k) \rightarrow \min_{x_1, ..., x_m \in \mathbb{R}^n} \sum_{k=1}^{m} f_k(x_k). \quad (P2)$$

or in another form

$$F(x) = \frac{1}{m} \sum_{k=1}^{m} f_k(x_k) \rightarrow \min_{x \in \mathbb{R}^n} \sum_{k=1}^{m} f_k(x_k), \quad (P2)$$

where

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

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 returns $\nabla f_k(x_k, \xi_k)$ (where $\xi = \{\xi_k\}_{k=1}^{m}$ are independent) under the following $\sigma^2$-subgaussian variance condition (for all $k = 1, ..., m$)

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

Problem (P2) can be considered to be a particular case of problem (11) with (for $\sigma^2$, see [29], [31])
\[
A = \sqrt{W}, \; L_F = \frac{L}{m}, \; \mu_F = \frac{\mu}{m}, \; \|\nabla F(x)\|^2 \leq M_F^2 = \frac{M^2}{m}, \; \sigma_F^2 = O\left(\frac{x^2}{m}\right),
\]

\[
R_k^2 = \|x^0 - x^*\|^2 = m\|x^0 - x^*\|^2 = mR^2, \; R_k^2 = \|y^*\|^2 \leq \frac{\|\nabla F(x^*)\|^2}{\lambda_{\min}(W)} \leq \frac{M^2}{m\lambda_{\min}(W)}.
\]

The main observation in primal approach (see Section III) is as follows [54]:

\[
A^T Ax = \sqrt{W}^T \sqrt{W} x = Wx - \text{calculated in a decentralized distributed manner!}
\]

If each function \(f_k\) is dual-friendly then we can construct dual problem to problem (P2) with dual Lagrangian variables \(y = [y_1^T, \ldots, y_m^T]^{T} \in \mathbb{R}^{mn}\)

\[
\Psi(y) = \frac{1}{m} \Phi(m \sqrt{W} y) = \frac{1}{m} \sum_{k=1}^{m} \varphi_k(m \sqrt{W} y_k) \rightarrow \min_{y \in \mathbb{R}^{mn}}, \quad (D2)
\]

where \(\varphi_k(\lambda_k) = \max_{x_k \in Q_k \subseteq \mathbb{R}^n} \{(\lambda_k, x_k) - f_k(x_k)\}\) and vector \([\sqrt{W}x]_k\) represents \(k\)-th \(n\)-dimensional block of \(\sqrt{W}x\).

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 \(\sigma^2\)-subgaussian variance condition (for all \(k = 1, \ldots, m\))

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

Problem (D2) can be considered to be a particular case of problem (15) with

\[
A = \sqrt{W}, \; \sigma^2 = O\left(\frac{\lambda_{\max}(W)m\sigma^2}{\lambda_{\min}(W)}\right).
\]

The main observation in dual approach (see Section IV) is as follows [54]:

\[
\text{Since } x(A^T y) = x(\sqrt{W} y) \text{ we should change the variables: } \hat{y} := \sqrt{W} y, \; z := \sqrt{W} z, \; y := \sqrt{W} y.
\]

It is obvious that Input, Output and lines 3–5 of Algorithm2 have changed such that they can be fulfilled in a decentralized distributed manner. For that we have just multiply corresponding lines on \(\sqrt{W}\).

VI. MAIN RESULTS

In this section we present the rates of convergence for problems (P1) and (D2) (and their stochastic counterparts) in terms of number of iterations (communication steps) and the number of (parallelized) oracle calls. For primal problem we present the results to achieve \(\varepsilon\) precision for the functional optimality gap, and for dual problem we seek to achieve \(\varepsilon\) precision for the duality gap or functional optimality gap (in smooth strongly convex case) and \(\varepsilon/R_g\) for the feasibility gap.

For brevity, we introduce the condition number of the Laplacian matrix \(W\) as is follows

\[
\chi = \frac{\lambda_{\max}(W)}{\lambda_{\min}(W)}
\]

Now we are ready to present our main results incorporated in multiple tables. This results are obtained by direct substitution of constants marked by the boxes to the bounds in Sections III, IV, V Note that the bounds on communications steps (rounds) are optimal (up to a logarithmic factor) due to the [5], [54], [55]. As for the the oracle calls per node, this bounds also seem to be optimal in the class of methods with optimal number of communications steps (up to a logarithmic factor) in deterministic case [2], [68], [19] and optimal in the class of methods with optimal number of communications steps in stochastic case in parallel architecture. For stochastic oracle the bounds holds true in terms of high probability deviations – we skip corresponding logarithmic factors.

Note that the red bound in Table VI seems to be rather unexpected at first sight for us.

VII. DISCUSSION

Below we provide different directions of further work.

- If instead of the first-order methods with primal and dual deterministic oracle one will use tensor methods (\(p = 2, 3\)) from [48] then the results can be improved. But for the dual approach we do not know how to use the trick \(A = \sqrt{W}\) [54]. Here we should take \(A \neq W\), that (with additional increased complexity of auxiliary problem) makes the bounds on communications steps worse. The basic fact in dual approach is as follows: to solve auxiliary problems we have to calculate the values of the form \(\nabla^2 \varphi(W y)\) on different vectors [7], [48], [49]. One can show that this can be done by

\footnote{In parallel architecture the bounds on stochastic oracle calls per node of type max\(\{B, D\}\) can be pararlelized up to \(B/D\) processors.}
successive multiplications of $W$ on vectors (communications) and corresponding (block) diagonal tensor (correspond to $\nabla_x \varphi(\lambda)|_{\lambda=Wx}$) on vectors (can be distributed among nodes);

- Primal approach in smooth case can be generalized for (stochastic inexact) gradient-free oracle. The number of communications steps remains the same. The number of oracle calls becomes $\sim n$ times larger [16], [18]. To the best of our knowledge, it is an open questions to generalize Lan’s sliding technique for gradient-free oracle. If one can do it, then it should be apply for gradient-free (stochastic inexact) decentralized distributed optimization;

- In [27], [28] asyncronized distributed optimization considered via dual accelerated (block) coordinate descent algorithms. The proposed above primal approach allows asyncronized generalizations in smooth case. For that we should use (block) coordinate version of STM [18] and additional randomization of sum type when calculate $[Wx]_i$. This will increase the number of communications steps in $\sim \sqrt{n} \div n$ times;

- Most of the results of this paper can be generalized to composite problems [46]. Perhaps, it is possible to next step and try to generalize these results to more general types of models [61], [62];

- For smooth convex centralized distributed optimization problems there exists a universal way to accelerate non-accelerate (stochastic, asyncronized etc.) algorithms – Catalyst [41]. The basic idea: to use not accelerated centralized distributed algorithm for inner problem arises at each step of Catalyst procedure;

- In the work [53] there proposed a way to generalize dual approach on time-varying graphs. Perhaps, it is also possible to generalize primal approach described above on time-varying graphs. Moreover, these generalizations can be done also for smooth stochastic case.

The main scheme in primal approach is based on the result formulated directly after (13). This result does not depend on convexity of target function. So it would be interesting to apply this scheme for non-convex distributed optimization problems [63].

We would like to thank F. Bach, S. Parsegov, A. Nemirovski, A. Olshevsky, B. Polyak, N. Srebro and A. Taylor for useful discussions.

This work is based on three talks of the first author:

- Winter school New frontiers in high-dimensional probability and statistics II, February, 2019, HSE, Moscow;
- WIAS seminars, March, 2019, Berlin;
- Workshop on Optimization and Statistical Learning, March, 2019, Les Houches.

The work of D. Dvinskikh in parts I, V, VI was funded by Russian Science Foundation (project 18-71-10108). The work of A. Gasnikov was supported by RFBR 18-31-20005 mol-a-ved and RFBR 18-29-03071 mk.

| TABLE III: Optimal bounds for primal deterministic oracle |
|--------------------------------------------------------|
| # of communication rounds | $f_k$ is $\mu$-strongly convex, and $L$-smooth | $f_k$ is $L$-smooth | $f_k$ is $\mu$-strongly convex |
|----------------------------|-----------------------------------------------|-------------------|------------------------------|
| $\tilde{O} \left( \sqrt{\frac{L^2}{\mu} \frac{1}{n}} \right)$ | $\tilde{O} \left( \sqrt{\frac{L}{c} \frac{1}{n}} \right)$ | $O \left( \frac{\sqrt{2\mu^2 L^2}}{\mu c} \right)$ | $O \left( \frac{\sqrt{2\mu^2 L^2}}{\mu^2 c^2} \right)$ |

| # of $\nabla f_k(x_k)$ oracle calls per node $k$ | $\tilde{O} \left( \sqrt{\frac{L}{n}} \right)$ | $O \left( \sqrt{\frac{L^2}{c} \frac{1}{n}} \right)$ | $O \left( \frac{\mu^2}{\mu^2 c^2} \right)$ | $O \left( \frac{\mu^2 L^2}{\mu c^2} \right)$ |

| Algorithm | PSTM, $Q = \mathbb{R}^n$ | PSTM, $Q = \mathbb{R}^n$ | R-Sliding | Sliding |

| TABLE IV: Optimal bounds for primal stochastic (unbiased) oracle |
|---------------------------------------------------------------|
| # of communication rounds | $f_k$ is $\mu$-strongly convex, and $L$-smooth | $f_k$ is $L$-smooth | $f_k$ is $\mu$-strongly convex, |
|----------------------------|-----------------------------------------------|-------------------|------------------------------|
| $\tilde{O} \left( \sqrt{\frac{L^2}{\mu} \frac{1}{n}} \right)$ | $\tilde{O} \left( \sqrt{\frac{L}{c} \frac{1}{n}} \right)$ | $O \left( \frac{\sqrt{2\mu^2 L^2}}{\mu c} \right)$ | $O \left( \frac{\sqrt{2\mu^2 L^2}}{\mu^2 c^2} \right)$ |

| # of $\nabla f_k(x_k, \xi_k)$ oracle calls per node $k$ | $\tilde{O} \left( \max \left\{ \frac{\mu^2}{\mu c}, \sqrt{\frac{L}{\mu}} \right\} \right)$ | $O \left( \max \left\{ \frac{\mu^2}{\mu c}, \sqrt{\frac{L^2}{c}} \right\} \right)$ | $O \left( \frac{\mu^2 + \mu^2}{\mu c^2} \right)$ | $O \left( \frac{\mu^2 + \mu^2}{\mu c^2} \right)$ |

| Algorithm | PBSTM, $Q = \mathbb{R}^n$ | PBSTM, $Q = \mathbb{R}^n$ | Stochastic R-Sliding | Stochastic Sliding |
TABLE V: Optimal bounds for dual deterministic oracle

| Algorithm          | $f_k$ is $\mu$-strongly convex, and $L$-smooth | $f_k$ is $\mu$-strongly convex |
|--------------------|---------------------------------------------|--------------------------------|
| ROGM-G or STM, $Q = \mathbb{R}^n$ | $O\left(\sqrt{\frac{\mu}{N}}\right)$ | $O\left(\sqrt{\frac{\mu}{N}}\right)$ |

TABLE VI: Optimal bounds for dual stochastic (unbiased) oracle

| Algorithm          | $f_k$ is $\mu$-strongly convex, and $L$-smooth | $f_k$ is $\mu$-strongly convex |
|--------------------|---------------------------------------------|--------------------------------|
| R-RRMA+AC-SA$^2$, $Q = \mathbb{R}^n$ | $O\left(\sqrt{\frac{\mu}{N}}\right)$ | $O\left(\sqrt{\frac{\mu}{N}}\right)$ |

REFERENCES

[1] Z. Allen-Zhu. 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, 2017. ACM. arXiv:1603.05953.

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

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

[4] A. S. Ankin, A. V. Gasnikov, P. E. Dvurechensky, A. I. Tyurin, and A. V. Chernov. 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, Aug 2017.

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

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

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

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

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

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

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

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

[13] D. Dvinskikh, E. Gorbunov, A. Gasnikov, and C. A. Uribe. On dual approach for distributed stochastic convex optimization over networks. arXiv preprint arXiv:1903.09844, 2019.

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

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

[16] P. Dvurechenskyy, A. Gasnikov, and E. Gorbunov. An accelerated method for derivative-free smooth stochastic convex optimization. arXiv:1802.09022, 2018.

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

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

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

[20] A. Gasnikov. Universal gradient descent. arXiv preprint arXiv:1711.00394, 2017.

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

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

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

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