SOLVING NONSMOOTH DECENTRALIZED OPTIMIZATION WITH AFFINE CONSTRAINTS USING GRADIENT SLIDING

Jiawei Chen
Skolkovo Institute of Science and Technology
Moscow Institute of Physics and Technology
Moscow, Russia
J.Chen@skoltech.ru

Zhenzhen Song
Skolkovo Institute of Science and Technology
Moscow, Russia

Alexander Rogozin[*]
Moscow Institute of Physics and Technology
Skolkovo Institute of Science and Technology
Moscow, Russia
aleksandr.rogozin@phystech.edu

Nhat Trung Nguyen, Demyan Yarmoshik, Irina Podlipnova & Alexander Gasnikov
Moscow Institute of Physics and Technology
Moscow, Russia
gasnikov@yandex.ru

ABSTRACT

This paper explores decentralized nonsmooth convex optimization with affine constraints. We extend existing research by incorporating a nonsmooth stochastic oracle, solved by the well know gradient sliding method. Our result show sliding algorithm achieves sub-optimal solution for these optimization problems under certain conditions, addressing limitations of prior methods. This work enhances the theoretical understanding of distributed optimization and offers practical solutions for applications in sensor networks and machine learning.

1 INTRODUCTION

The study of solving the convex optimization problems in a distributed setting has a long history in the optimization community (Tsitsiklis, 1984; Bertsekas & Tsitsiklis, 2015). Building on the seminal work (Tsitsiklis, 1984), in recent years, there has been a flurry of research around the problem of solving convex optimization problem in the framework of multiagent systems (Nedic & Ozdaglar, 2009; Yuan & Ho, 2014). In particular, the global objective function of the problem is a sum of functions that are distributed over a network, which consists of multiple interacting nodes. Such problem arises in a variety of real applications ranging from sensor networks to machine learning (Duchi et al., 2011; Johansson et al., 2008).

Decentralized convex optimization without affine constraints has been extensively studied. It is well-established that the performance of optimization algorithms applied to strongly-convex smooth objectives is bounded below by a multiple of the graph condition number and the objective condition number, up to a logarithmic factor (Scaman et al., 2017).Moreover constrained distributed optimization has attracted significant attention from researchers. An early application of first-order methods to constrained decentralized optimization is illustrated by the projected subgradient algorithm, as discussed in (Nedic et al., 2010), which also analyzed time-varying networks. A comprehensive review of the main problem classes in distributed constrained optimization, along with algorithms

[*] Corresponding author

suitable for various levels of decentralization, is presented in (Necoara et al., 2011). We will breifly them below:

Necoara & Nedelcu (2014) propose distributed dual gradient algorithms for linearly constrained separable convex problems, it means each agent in the network has their own variable. Moreover, they supposed affine constraints are network compatable (constraint matrix can have a non-zero element on position (i, j) only if there is an edge in communication graph between agents i and j. In our study, we don't have such conditions, the variable is shared among agents , which is clearly a special case of (Necoara & Nedelcu, 2014) , and we have various network structure.

In Necoara et al. (2011) the authors present several formulations of distributed optimization problems, covering scenarios with various types of interconnections between constraints and objectives, including cases where the overall objective (cost) does not equal to the sum of individual cost functions for each agent. However, their algorithms for problems with coupled affine constraints need to solve a "master problem" at a centralized node during each iteration, thereby compromising the decentralization ability.

Rogozin et al. (2022) solve a decentralized convex optimization with affine constraints, and get linear convergence rate. We take this as the base of our problem, and extend it to nonsmooth stochastic oracle.

1.1 Notations and definitions

We use $\langle x, y\rangle \overset{\text{def}}{=} \sum_{i=1}^{n} x_i y_i$ to denote standard inner product of $x, y \in \mathbb{R}^n$ where x_i corresponds to the i-th component of x in the standard basis in $\mathbb{R}^n$. The dual norm $\|\cdot\|_*$ for the norm $\|\cdot\|$ is defined in the following way: $\|y\|_* \overset{\text{def}}{=} \max\{\langle x, y\rangle \mid \|x\| \le 1\}$. To denote maximal and minimal positive eigenvalues of positive semidefinite matrix $A \in \mathbb{R}^{n\times n}$ we use $\lambda_{\max}(A)$ and $\lambda_{\min}^{+}(A)$ respectively and we use $\chi(A) \overset{\text{def}}{=} \lambda_{\max}(A)/\lambda_{\min}^{+}(A)$ to denote condition number of A. Operator $\mathbb{E}[\cdot]$ denotes full mathematical expectation. To define the Kronecker product of two matrices. $A \in \mathbb{R}^{m\times m}$ and $B \in \mathbb{R}^{n\times n}$ we use $A \otimes B \in \mathbb{R}^{nm\times nm}$. The identity matrix of the size $n \times n$ is denoted in our paper by I_n. The diameter of set $\mathcal{X}$ is denoted by $D_{\mathcal{X}} = \max\{\|x - y\|| \quad \forall x, y \in \mathcal{X}\}$. We use col$(\cdot)$ to represent the column vector.

1.2 Problem formulation

$$\min_{x\in\mathcal{X}\subset\mathbb{R}^d} \frac{1}{m}\sum_{i=1}^{m} f_i(x) \quad \text{s.t.} \quad Bx = 0. \tag{1}$$

Assumption 1 *Assume that we have access to the stochastic oracle of f_i. For a given point x, it output $f_i^{'}(x, \xi)$ such that*

$$\mathbb{E}[f_i^{'}(x, \xi)] = f_i'(x) \in \partial f_i(x),$$
$$\mathbb{E}[\|f_i^{'}(x, \xi) - f_i'(x)\|^2] \le \sigma^2.$$

Assumption 2 *f_i is a convex and nonsmooth function satisfying*

$$f_i(x) \le f_i(y) + \langle f_i'(x), x - y\rangle + M\|x - y\|^2, \forall x, y \in \mathcal{X}. \tag{2}$$

We assume that the problem is distributed over a connected network consisting of m agents. Each agent locally store f_i. Agents are connected through a communication network. Agents are only allowed to exchange information with their neighborhoods. Further, we rely on the notion of gossip matrix. W is a gossip matrix of an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, $\|\mathcal{V}\| = m$, if it satisfies following properties:

Assumption 3 *propertoes of W:*

1. *W is a symmetric positive semi-definite matrix.*
2. *(Network compatibility) For all $i, j = 1, \ldots, m$ it holds $[W]_{ij} = 0$ if $(i, j) \notin \mathcal{E}$ and $i \ne j$.*

3. *(Kernel property) For any $v = [v_1, \ldots, v_m]^\top \in \mathbb{R}^m$, $Wv = 0$ if and only if $v_1 = \ldots = v_m$.*

An classical example of a matrix that satisfies Assumption 3 is the graph Laplacian matrix $W \in \mathbb{R}^{m\times m}$:

$$[W]_{ij} = \begin{cases} -1, & \text{if } (i,j) \in E, \\ \deg(i), & \text{if } i = j, \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

where $\deg(i)$ is the degree of the i-th node, i.e., the number of neighbors of the i-th agent. Since we consider only connected networks, the matrix W has a unique eigenvector $\mathbf{1}_m \overset{\text{def}}{=} (1, \ldots, 1)^\top \in \mathbb{R}^m$ corresponding to the eigenvalue 0. It implies that for all vectors $a = (a_1, \ldots, a_m)^\top \in \mathbb{R}^m$ the following equivalence holds:

$$a_1 = \ldots = a_m \iff Wa = 0. \tag{4}$$

Now let us think about a_i as a number that the i-th node stores. Then, we can express in short matrix form. To generalize it for the case when a_j are vectors from $\mathbb{R}^d$, we should consider the matrix $\mathbf{W} \overset{\text{def}}{=} W \otimes I_d$, where $\otimes$ represents the Kronecker product. Indeed, if we consider vectors $x_1, \ldots, x_m \in \mathbb{R}^d$ and $\mathbf{x} = (x_1^\top, \ldots, x_m^\top)^\top \in \mathbb{R}^{md}$, then

$$x_1 = \ldots = x_m \iff W\mathbf{x} = 0. \tag{5}$$

2 Results and algorithm

We use the gradient sliding algorithm from (Lan, 2016), which is designed to solve composite convex optimization problems of the form:

$$\min_{x\in\mathcal{Q}} \{\phi(x) = g(x) + f(x)\}, \tag{6}$$

where $f(x)$ is a nonsmooth convex function and $g(x)$ is a smooth convex function. Initially, we demonstrate the transformation of problem (7) to fit the gradient sliding algorithm, ensuring that it can achieve an ε-suboptimal solution. Subsequently, we reformulate problem (1) to problem (7).

2.1 Convex optimization with two affine constraints

First we introduce a minimization problem with two affine constraints :

$$\min_{\mathbf{x}\in\mathcal{Q}} \mathbf{F}(\mathbf{x}) \tag{7}$$

$$s.t. \quad \mathbf{B}\mathbf{x} = 0, \mathbf{C}\mathbf{x} = 0, \tag{8}$$

where $F(\mathbf{x}) = \frac{1}{m}\sum_{i=1}^m f_i(x_i)$ and $\mathbf{x} = \text{col}(x_1, \ldots, x_m)$, and also introduce $\mathbf{B} = B \otimes \mathbf{I}_m, \mathbf{C} = C \otimes \mathbf{I}_d$, $B \in \mathbb{R}^{p\times d}$, $C \in \mathbb{R}^{m\times m}$.

By choosing a positive scalar γ, we can use the trick in Rogozin et al. (2022) to build $\mathbf{A}^\top = [\mathbf{B}^\top \quad \gamma\mathbf{C}]$ and the dual problem of problem (7) is:

$$\min_{\mathbf{y}} \quad \Phi(\mathbf{y}), \text{ where} \tag{9}$$

$$\Phi(\mathbf{y}) = \max_{\mathbf{x}\in\mathcal{Q}} \langle \mathbf{y}, \mathbf{A}\mathbf{x}\rangle - \mathbf{F}(\mathbf{x}) = \langle \mathbf{A}^\top\mathbf{y}, \mathbf{x}(\mathbf{A}^\top\mathbf{y})\rangle - F(\mathbf{x}(\mathbf{A}^T\mathbf{y})), \tag{10}$$

where $\mathbf{x}(\mathbf{y}) = \arg\min_{\mathbf{x}\in\mathcal{Q}}\{\langle \mathbf{y}, \mathbf{x}\rangle - \mathbf{F}(\mathbf{x})\}$. We use $\mathbf{y}^\star$ to denote the solution of (9) with the smallest l_2-norm $R_{\mathbf{y}} = \|\mathbf{y}^\star\|_2$. And $R_{\mathbf{y}}$ can be bounded as follows:

$$R_{\mathbf{y}}^2 \leq \frac{\|\nabla\mathbf{F}(\mathbf{x}^\star)\|^2}{\lambda_{\min}^+(\mathbf{A}^\top\mathbf{A})}. \tag{11}$$

Then we can introduce

$$\tilde{\mathbf{F}}(\mathbf{x}) = \mathbf{F}(\mathbf{x}) + \frac{R_{\mathbf{y}}^2}{\varepsilon}\|\mathbf{A}\mathbf{x}\|^2, \tag{12}$$

where $\varepsilon > 0$ is the desired accuracy of the solution in terms of $\mathbf{F}(\mathbf{x})$ that we want to achieve.

Gorbunov et al. (2019) proved that if we have $\hat{\mathbf{x}}$ such that $\tilde{\mathbf{F}}(\hat{\mathbf{x}}) - \min_{\mathbf{x}\in\mathcal{Q}} \tilde{\mathbf{F}}(\mathbf{x}) \leq \varepsilon$, then we have

$$\mathbf{F}(\hat{\mathbf{x}}) - \min_{\mathbf{x}\in\mathcal{Q}} \mathbf{F}(\mathbf{x}) \leq \varepsilon, \quad \|\mathbf{A}\hat{\mathbf{x}}\|_2 \leq \frac{2\varepsilon}{R_{\mathbf{y}}}. \tag{13}$$

Then this result can be generalized to the stochastic case: if we have $\hat{\mathbf{x}}$ such that $\mathbb{E}[\tilde{\mathbf{F}}(\hat{\mathbf{x}})] - \min_{\mathbf{x}\in\mathcal{Q}} \tilde{\mathbf{F}}(\mathbf{x}) \leq \varepsilon$, then we have

$$\mathbb{E}[\mathbf{F}(\hat{\mathbf{x}})] - \min_{\mathbf{x}\in\mathcal{Q}} \mathbf{F}(\mathbf{x}) \leq \varepsilon, \quad \sqrt{\mathbb{E}\|\mathbf{A}\hat{\mathbf{x}}\|_2^2} \leq \frac{2\varepsilon}{R_{\mathbf{y}}}. \tag{14}$$

Next, we consider solving this problem with the gradient sliding algorithm 1:

$$\min_{\mathbf{x}\in\mathcal{Q}} \tilde{\mathbf{F}}(\mathbf{x}) = f(\mathbf{x}) + g(\mathbf{x}), \tag{15}$$

$$\text{where } f(\mathbf{x}) = \frac{1}{m}\sum_{i=1}^{m} f_i(x_i), \quad g(\mathbf{x}) = \frac{R_{\mathbf{y}}^2}{\varepsilon}\|\mathbf{A}\mathbf{x}\|^2. \tag{16}$$

Algorithm 1 Sliding Algorithm

```
**Input:** Initial point $x_0 \in \mathcal{X}$ and iteration limit $N$.
Let $\beta_k \in \mathcal{R}_+$, $\gamma_k \in \mathcal{R}_+$, and $T_k \in \mathbb{N}$, $k = 1, 2, \ldots$, be given and set $\overline{x}_0 = x_0$.
**for** $k = 1, 2, \ldots, N$ **do**
    Set $\underline{x}_k = (1-\gamma_k)\overline{x}_{k-1} + \gamma_k x_{k-1}$, and let $h_k(\cdot) \equiv l_g(\underline{x}_k, \cdot)$, where $l_g(x,y) = g(x) + \langle \nabla g(x), y - x\rangle$.
    Set $(x_k, \tilde{x}_k) = \mathbb{PS}(h_k, x_{k-1}, \beta_k, T_k)$.
    Set $\overline{x}_k = (1-\gamma_k)\overline{x}_{k-1} + \gamma_k \tilde{x}_k$.
**end for**
**Output:** $\overline{x}_N$.

**procedure** $(x^+, \tilde{x}^+) = \mathbb{PS}(h, x, \beta, T)$
    Let the parameters $p_t \in \mathcal{R}_+$ and $\theta_t \in [0,1]$, $t = 1, \ldots$, be given. Set $u_0 = \tilde{u}_0 = x$.
    **for** $t = 1, 2, \ldots, T$ **do**
        Set $u_t = \arg\min_{u\in\mathcal{X}} \left\{ h(u) + l_f(u_{t-1}, u) + \frac{\beta}{2}\|u - x\|_2^2 + \frac{\beta p_t}{2}\|u - u_{t-1}\|_2^2 \right\}$.
        Set $\tilde{u}_t = (1-\theta_t)\tilde{u}_{t-1} + \theta_t u_t$.
        where $l_f(x,y) = f(x) + \langle f'(x,\xi), y - x\rangle$.
    **end for**
    Set $x^+ = u_T$ and $\tilde{x}^+ = \tilde{u}_T$.
**end procedure**
```

We reuse the parameters in Lan (2016), then we can have this theorem:

Theorem 1 *Assume that $\{p_t\}$ and $\{\theta_t\}$ in the PS procedure of algorithm 1 are set to*

$$p_t = \frac{t}{2} \quad \textit{and} \quad \theta_t = \frac{2(t+1)}{t(t+3)}, \quad \forall t \geq 1.$$

If N is fixed positive number, and $\{\beta_k\}$, $\{\gamma_k\}$, and $\{T_k\}$ are set to

$$\beta_k = \frac{2L}{k}, \quad \gamma_k = \frac{2}{k+1}, \quad \textit{and} \quad T_k = \left\lceil \frac{(\hat{M}^2 + \sigma^2)Nk^2}{\tilde{D}L^2} \right\rceil.$$

Then it can achieve (14) with probability at least $1 - \beta$, $\beta \in (0,1)$ requiring

$$O\left(\sqrt{\frac{\lambda_{\max}(\mathbf{A}^\top\mathbf{A})R_{\mathbf{y}}^2 D_{\mathcal{Q}}^2}{\varepsilon^2}}\right) \textit{ calculations of } \mathbf{A}^\top\mathbf{A}\mathbf{x}, \tag{17}$$

and

$$\widetilde{O}\left(\frac{(M^2 + \sigma^2)D_{\mathcal{Q}}^2}{\varepsilon^2}\right) \textit{ calculations of } \mathbf{F}'(\mathbf{x}, \xi), \tag{18}$$

when f is a μ-strongly convex function, then use restart technique with algorithm 1, it can achieve (14) with probability at least $1-\beta$, $\beta \in (0,1)$ requiring

$$\widetilde{O}\left(\sqrt{\frac{\lambda_{\max}(\mathbf{A}^\top\mathbf{A})R_{\mathbf{y}}^2}{\mu\varepsilon}}\right) \text{ calculations of } \mathbf{A}^\top\mathbf{A}\mathbf{x}, \tag{19}$$

and

$$\widetilde{O}\left(\frac{M^2+\sigma^2}{\mu\varepsilon}\right) \text{ calculations of } \mathbf{F}'(\mathbf{x},\xi), \tag{20}$$

where $\mathbf{F}'(\mathbf{x},\xi) = \text{col}(f_1^{'}(x,\xi)\dots f_m^{'}(x,\xi))$, $\tilde{D} = \frac{3}{4}D_{\mathcal{Q}}$ and L is the smoothness constant of g, $\|\mathbf{F}'(\mathbf{x},\xi)\|_2 \le \hat{M}$.

In this section, we first reformulate a convex optimization problem with two affine constraints into a minimization problem by introducing penalty terms. These penalty terms arise from the combination of the affine constraints and the variable $\mathbf{x}$. We then apply the gradient sliding algorithm to this problem, yielding the convergence rate discussed above.

2.2 DECENTRALIZED GRADIENT SLIDING

We can rewrite problem (1) into this form:

$$\min_{\substack{x_1=\dots=x_m\\ x_1,x_2,\cdots,x_m\in\mathcal{X}}} f(\mathbf{x}) = \frac{1}{m}\sum_{i=1}^m f_i(x_i) \tag{21}$$

$$s.t. \quad \mathbf{B}\mathbf{x} = 0, \tag{22}$$

where $\mathbf{x} = \text{col}(x_1,\dots,x_m)$. As we mentioned before, each agent i store individual objective function f_i in this network, and when $x_1 = x_2 = \dots = x_m$, $\mathbf{W}\mathbf{x}$ will be equal to $\mathbf{0}$ in our setting. Therefore, we introduce $\mathbf{W}\mathbf{x}$ in (24) as a penalty term to ensure that each x_i converges to the optimal solution, then we can reformulate problem (21) into this form:

$$\min_{\substack{\mathbf{W}\mathbf{x}=0,\\ \mathbf{B}\mathbf{x}=0,\\ \mathbf{x}\in\mathcal{X}^m\subset\mathbb{R}^{md}}} f(\mathbf{x}) = \frac{1}{m}\sum_{i=1}^m f_i(x_i) \tag{23}$$

Then we can directly apply result in section 2.1,

$$\min_{\mathbf{x}\in\mathcal{X}^m} \tilde{\mathbf{F}}(\mathbf{x}) = \mathbf{F}(\mathbf{x}) + \frac{R_{\mathbf{y}}^2}{\varepsilon}\|\mathbf{A}\mathbf{x}\|^2, \tag{24}$$

where $F(\mathbf{x}) = \frac{1}{m}\sum_{i=1}^m f_i(x_i)$, and $\mathbf{A}^\top = [\mathbf{B}^\top \quad \gamma\mathbf{W}]$.

From assumption 2, we can know that $\|f_i'(x_i)\|_2 \le M$, for all $x_i \in \mathcal{X}$, all f_i are convex functions, then set $\mathbf{x}_0^\top = (x_0^\top,\dots,x_0^\top)^\top$ and $\mathbf{x}_*^\top = (x_*^\top,\dots,x_*^\top)^\top$ is the optimality point for (24), from Gorbunov et al. (2019) we can get

$$D_{\mathcal{X}^m}^2 = mD_{\mathcal{X}}^2, \quad \|\nabla f(\mathbf{x})\|_2 \le \frac{M}{\sqrt{m}}, \quad R_{\mathbf{y}}^2 \overset{\text{def}}{=} \|\mathbf{y}_*\|_2^2 \le \frac{M^2}{m\lambda_{\min}(\mathbf{A}^\top\mathbf{A})}. \tag{25}$$

We now need to apply the results obtained in Section 2.1 to the problem (24). To produce a point $\hat{\mathbf{x}}$ that satisfies (14), where $\hat{\mathbf{x}} = \hat{\mathbf{x}}$, $\mathbf{A}^\top := [\mathbf{B}^\top \quad \gamma\mathbf{W}]$, $\mathcal{Q} := \mathcal{X}^m$, and $R_{\mathbf{y}} := R_{\mathbf{y}}$, Algorithm 1 applied to the penalized problem (24) , achieving (14) with probability at least $1-\beta$, $\beta \in (0,1)$, it requires communication complexity and computation complexity same as in Theorem 1.

By accurately choosing factor γ, we can control the condition number $\chi(\mathbf{A}^\top\mathbf{A})$. The minimal value of $\chi(\mathbf{A}^\top\mathbf{A})$ is attained at $\gamma^2 = \frac{\lambda_{\min}^+(B^\top B)}{(\lambda_{\min}^+(W))^2}$ and equals $\chi(\mathbf{A}^\top\mathbf{A}) = \chi(B^\top B) + \chi^2(W)$ in (Rogozin et al., 2022). Then achieving (14) with probability at least $1-\beta$, $\beta \in (0,1)$, algorithm 1 requires

$$\widetilde{O}\left(\frac{(M^2+\sigma^2)D_{\mathcal{X}}^2}{\varepsilon^2}\right) \text{ calculations of } f'(x,\xi) \text{ per node.} \tag{26}$$

and

$$O\left(\sqrt{\frac{\chi^2(W)MD_{\mathcal{X}}^2}{\varepsilon^2}}\right) \text{ communications,} \tag{27}$$

and

$$O\left(\sqrt{\frac{\chi(B^\top B)MD_{\mathcal{X}}^2}{\varepsilon^2}}\right) \text{ multiplications by } B^\top B \text{ per node.} \tag{28}$$

When f_i is a μ-strongly convex function, then it can achieve 14 with probability at least $1-\beta$, $\beta \in (0,1)$ requiring

$$\widetilde{O}\left(\frac{M^2+\sigma^2}{\mu\varepsilon}\right) \text{ calculations of } f'(x,\xi) \text{ per node,} \tag{29}$$

and

$$\widetilde{O}\left(\sqrt{\frac{\chi^2(W)M}{\mu\varepsilon}}\right) \text{ communications,} \tag{30}$$

and

$$\widetilde{O}\left(\sqrt{\frac{\chi(B^\top B)M}{\mu\varepsilon}}\right) \text{ multiplications by } B^\top B \text{ per node.} \tag{31}$$

In this section, we reformulate problem (1) into a convex optimization problem with two affine constraints. Next, we apply the method from Section 2.1, which allows us to obtain the convergence rate for problem (1).

3 Conclusion

In this paper, we have addressed the problem of decentralized convex optimization with affine constraints. We introduce a novel approach that extends the gradient sliding method to incorporate a nonsmooth stochastic oracle, resulting in a decentralized algorithm that achieves linear convergence for such optimization problems. This work overcomes the limitations of previous methods by providing a practical solution that advances the theoretical understanding of distributed optimization. However, our approach relies on the gradient sliding algorithm, which requires parameter estimation before implementation, slightly weakening its theoretical performance. In our experiments, we showed that the effect of choice for different parameters R and T, as in Figure 1.

Future work will focus on extending the algorithm to handle biased stochastic oracle and non-convex objectives, as well as exploring adaptive strategies to dynamically adjust the parameters of the algorithm based on the network topology and the structure of the optimization problem.

4 Numerical Experiments

We conducted numerical experiments on the following optimization problem:

$$\min_x f(x) := \frac{1}{n}\sum_{i=1}^n f_i(x) \quad \text{subject to } Bx = 0,$$

where

$$f_i(x) = \sqrt{\frac{1}{m}\|C_i x - d_i\|^2},$$

with $C_i \in \mathbb{R}^{m\times d}$, $d_i \in \mathbb{R}^{m\times 1}$, and $x \in \mathbb{R}^d$.

The algorithm was tested on four nodes within a single machine. For simplicity, local variables were stored in a single long vector. The dimension of each local variable was set to 5, and the number of samples was 1,000. The algorithm was run with a batch size of 100 over 2,000 iterations. Instead of estimating the parameters, we experimented with various values of the inner loop T and the penalty term coefficient $R = \frac{R_y}{\epsilon}$. We found that $T = 3$ provided good performance, leading us to compare different values of R. Additionally, four distinct network topologies (complete graph, path graph, cycle graph, and star graph) were used in this experiment. The script and data that support the findings of this study are available from the corresponding author upon reasonable request.

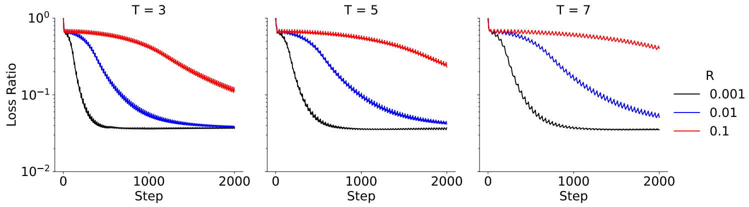

Figure 1: **Loss ratio for different choices of T on a complete graph.** The choice of T affects the convergence rate, but the parameter R has a more significant impact. A smaller R leads to faster convergence, though it results in less consensus among the local variables x_i.

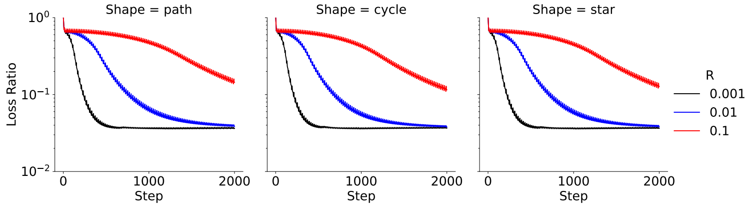

Figure 2: **Loss ratio for different choice of network structure.**

Acknowledgments

The research was supported by Russian Science Foundation (project No. 23-11-00229).

References

Dimitri Bertsekas and John Tsitsiklis. *Parallel and distributed computation: numerical methods*. Athena Scientific, 2015.

John C Duchi, Alekh Agarwal, and Martin J Wainwright. Dual averaging for distributed optimization: Convergence analysis and network scaling. *IEEE Transactions on Automatic control*, 57(3): 592–606, 2011.

Eduard Gorbunov, Darina Dvinskikh, and Alexander Gasnikov. Optimal decentralized distributed algorithms for stochastic convex optimization. *arXiv preprint arXiv:1911.07363*, 2019.

Bjorn Johansson, Tamás Keviczky, Mikael Johansson, and Karl Henrik Johansson. Subgradient methods and consensus algorithms for solving convex optimization problems. In *2008 47th IEEE Conference on Decision and Control*, pp. 4185–4190. IEEE, 2008.

Guanghui Lan. Gradient sliding for composite optimization. *Mathematical Programming*, 159: 201–235, 2016.

Ion Necoara and Valentin Nedelcu. Distributed dual gradient methods and error bound conditions. *arXiv preprint arXiv:1401.4398*, 2014.

Ion Necoara, Valentin Nedelcu, and Ioan Dumitrache. Parallel and distributed optimization methods for estimation and control in networks. *Journal of Process Control*, 21(5):756–766, 2011.

Angelia Nedic and Asuman Ozdaglar. Distributed subgradient methods for multi-agent optimization. *IEEE Transactions on Automatic Control*, 54(1):48–61, 2009.

Angelia Nedic, Asuman Ozdaglar, and Pablo A Parrilo. Constrained consensus and optimization in multi-agent networks. *IEEE Transactions on Automatic Control*, 55(4):922–938, 2010.

Alexander Rogozin, Demyan Yarmoshik, Ksenia Kopylova, and Alexander Gasnikov. Decentralized strongly-convex optimization with affine constraints: Primal and dual approaches. In *International Conference on Optimization and Applications*, pp. 93–105. Springer, 2022.

Kevin Scaman, Francis Bach, Sébastien Bubeck, Yin Tat Lee, and Laurent Massoulié. Optimal algorithms for smooth and strongly convex distributed optimization in networks. In *international conference on machine learning*, pp. 3027–3036. PMLR, 2017.

John N Tsitsiklis. *Problems in decentralized decision making and computation*. PhD thesis, Massachusetts Institute of Technology, 1984.

Deming Yuan and Daniel WC Ho. Randomized gradient-free method for multiagent optimization over time-varying networks. *IEEE Transactions on Neural Networks and Learning Systems*, 26 (6):1342–1347, 2014.