Optimal Decentralized Algorithms for Saddle Point Problems over Time-Varying Networks*

Aleksandr Beznosikov[1,2], Alexander Rogozin[1,2], Dmitry Kovalev[3], and Alexander Gasnikov[1,2]

[1] Moscow Institute of Physics and Technology, Dolgoprudny, Russia
[2] Higher School of Economics, Russia
[3] King Abdullah University of Science and Technology, Saudi Arabia

Abstract. Decentralized optimization methods have been in the focus of optimization community due to their scalability, increasing popularity of parallel algorithms and many applications. In this work, we study saddle point problems of sum type, where the summands are held by separate computational entities connected by a network. The network topology may change from time to time, which models real-world network malfunctions. We obtain lower complexity bounds for algorithms in this setup and develop optimal methods which meet the lower bounds.

Keywords: saddle-point problem · distributed optimization · decentralized optimization · time-varying network · lower and upper bounds

1 Introduction

Distributed algorithms are an important part of solving many applied optimization problems [24,14,15]. They help to parallelize the computation process and make it faster. In this paper, we focus on the distributed methods for the saddle point problem:

$$\min_{x\in\mathcal{X}}\max_{y\in\mathcal{Y}} f(x,y) := \frac{1}{M}\sum_{m=1}^{M} f_m(x,y). \tag{1}$$

In this formulation of the problem, the original function f is divided into M parts, each of part f_m is stored on its own local device. Therefore, only the device with the number m knows information about f_m. Accordingly, in order to obtain complete information about the function f, it is necessary to establish a communication process between devices. This process can be organized in two ways: centralized and decentralized. In a centralized approach, communication takes place via a central server, i.e. all devices can send some information about their local f_m function to the central server, the server collects information from the devices and does some additional calculations, and then can send

*The research of A. Beznosikov, A. Rogozin was supported by Russian Science Foundation (project No. 21-71-30005). The work of A. Gasnikov was supported by RFBR 19-31-51001.

new information or request to the devices. Then the process continues. With this approach, one can easily write centralized gradient descent for distributed sum minimization: $\min_x g(x) := \frac{1}{M}\sum_{m=1}^{M} g_m(x)$. All devices compute local gradients in the same current point and then send these gradients to the server, in turn, the server averages the gradients and makes a gradient descent step, thereby obtaining a new current point, which it sends to the devices. Centralized methods for (1) are discussed in detail, for example, in [3]. However, centralized approach has several problems, e.g. synchronization drawback or high requirements to the server. Possible approach to deal with these drawbacks is to use decentralized architecture [1]. In this case, there is no longer any server, and the devices are connected into a certain communication network and workers are able to communicate only with their neighbors and communications are simultaneous. The most popular and frequently used communication methods are the gossip protocol [6,4,17] and accelerated gossip protocol [23,25]. In the gossip protocol, nodes iteratively exchange data with their immediate neighbors using a communication matrix and in this way the information diffuses over the network. The rate of convergence depends on the ratio χ of maximal and minimal non-zero eigenvalues of the communication matrix, which is typically proportional to diameter of the graph squared. Accelerated consensus can be achieved i.e. by Chebyshev acceleration [23] and improves the dependence from χ to $\sqrt{\chi}$, which is optimal. However, the non-accelerated variant is more robust, e.g. it can be applied to time-varying (wireless) communication networks.

1.1 Our contribution

In particular, our contribution can be briefly described as follows

Lower bounds. We present lower bounds for decentralized smooth strongly-convex-strongly-concave and convex-concave saddle-point problems on the time-varying networks. The lower bounds are derived under the assumption that the network is always a connected graph.

Optimal algorithm. The paper constructs an optimal algorithm that meets the lower bounds. The analysis of the algorithm is carried out for smooth strongly-convex-strongly-concave and convex-concave saddle-point problems

See our results in the column "time-varying" of Table 1.

1.2 Related works

Our work is one of the first dedicated to decentralized saddle problems over time-varying networks. Among other works, we can highlight the following paper [2]. This work looks at a more general time-varying setting and suggests a new method. The upper bounds for their method are worse than for our method. We also mention papers on related topics:

Decentralized saddle point problems.

The next work is devoted to centralized and decentralized distributed saddle problems [3]. It carries out lower bounds and optimal algorithms in the case

when the communication network is constant (non-time-varying). See Table 1 for comparison our results for time-varying topology and results from [3] for constant network. Also note the following works devoted to decentralized min-max problems: in deterministic case [12,16,20], in stochastic case [11].

Minimization on time-varying networks.

Decentralized methods are built upon combining iterations of classical first-order methods with communication steps. In the case of time-varying networks, a non-accelerated communication procedure is employed. Paper [17] can be named as an initial work on decentralized sub-gradient methods, and [18] proposed DIG-ing – the first first-order minimization algorithm with linear convergence over time-varying networks. After that, PANDA, which is a dual method capable of working over time-varying graphs, was proposed in [13]. Analysis of DIGing and PANDA assumes that the underlying network is B-connected, that is, the union of B consequent networks is connected, while the network is allowed to be disconnected at some steps. Considering the time-varying graphs which stay connected at each iteration, decentralized Nesterov method [22] has an accelerated rate under the condition that graph changes happen rarely enough, ADOM [9] and ADOM+ [8] are first-order optimization methods which achieve lower complexity bounds [8]. APM-C [21], Acc-GT [10] are accelerated methods over time-varying graphs, as well. The mentioned results are devoted to minimization algorithms and can be generalized to saddle-point problems. In this paper we generalize lower bounds of [8] to min-max problems and obtain an algorithm which reaches them up to a logarithmic factor.

	time-varying network	**constant network [3]**
	`lower`	
`sc`	$\Omega\left(R_0^2 \exp\left(-\frac{\mu K}{256 L \chi}\right)\right)$	$\Omega\left(R_0^2 \exp\left(-\frac{\mu K}{128 L \sqrt{\chi}}\right)\right)$
`c`	$\Omega\left(\frac{LD^2\chi}{K}\right)$	$\Omega\left(\frac{LD^2\sqrt{\chi}}{K}\right)$
	`upper`	
`sc`	$\tilde{\mathcal{O}}\left(R_0^2 \exp\left(-\frac{\mu K}{8 L \chi}\right)\right)$	$\tilde{\mathcal{O}}\left(R_0^2 \exp\left(-\frac{\mu K}{8 L \sqrt{\chi}}\right)\right)$
`c`	$\tilde{\mathcal{O}}\left(\frac{LD^2\chi}{K}\right)$	$\tilde{\mathcal{O}}\left(\frac{LD^2\sqrt{\chi}}{K}\right)$

Table 1. Lower and upper bounds for distributed smooth stochastic strongly-convex–strongly-concave (`sc`) or convex-concave (`c`) saddle-point problems in centralized and decentralized cases. Notation: L – smothness constant of f, μ – strongly-convex-strongly-concave constant, $R_0^2 = \|x_0 - x^*\|_2^2 + \|y_0 - y^*\|_2^2$, D – diameter of optimization set, χ – condition number of communication graph (in time-varying case maximum of all graphs), K – number of communication rounds. In the case of upper bounds in the convex-concave case, the convergence is in terms of the "saddle-point residual", in the rest – in terms of the (squared) distance to the solution.

2 Preliminaries

We use $\langle z, u\rangle := \sum_{i=1}^{d} z_i u_i$ to denote standard inner product of $z, u \in \mathbb{R}^d$. It induces ℓ_2-norm in $\mathbb{R}^d$ in the following way $\|z\| := \sqrt{\langle z, z\rangle}$. We also introduce the following notation $\text{proj}_{\mathcal{Z}}(z) = \min_{u\in\mathcal{Z}} \|u - z\|$ – the Euclidean projection onto $\mathcal{Z}$.

We work with the problem (1), where the sets $\mathcal{X} \subseteq \mathbb{R}^{n_x}$ and $\mathcal{Y} \subseteq \mathbb{R}^{n_y}$ are convex sets. Additionally, we introduce the set $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$, $z = (x, y)$ and the operator F:

$$F_m(z) = F_m(x, y) = \begin{pmatrix} \nabla_x f_m(x, y) \\ -\nabla_y f_m(x, y) \end{pmatrix}. \tag{2}$$

This notation is needed for shortness.

Problem setting. Next, we introduce the following assumptions:

Assumption 1(g). $f(x, y)$ is L - smooth, if for all $z_1, z_2 \in \mathcal{Z}$

$$\|F(z_1) - F(z_2)\| \leq L\|z_1 - z_2\|. \tag{3}$$

Assumption 1(l). For all m, $f_m(x, y)$ is Lipschitz continuous with constant $L_{\max}$, it holds that for all $z_1, z_2 \in \mathcal{Z}$

$$\|F_m(z_1) - F_m(z_2)\| \leq L_{\max}\|z_1 - z_2\|. \tag{4}$$

Assumption 2(s). $f(x, y)$ is strongly-convex-strongly-concave with constant μ, if for all $z_1, z_2 \in \mathcal{Z}$

$$\langle F(z_1) - F(z_2), z_1 - z_2\rangle \geq \mu\|z_1 - z_2\|^2. \tag{5}$$

Assumption 2. $f(x, y)$ is convex-concave, if $f(x, y)$ is strongly-convex-strongly-concave with 0.

Assumption 3. $\mathcal{Z}$ – compact bounded, i.e. for all $z, z' \in \mathcal{Z}$

$$\|z - z'\| \leq D. \tag{6}$$

All assumptions are standard in the literature.

Network setting. In each moment of time (iteration) t, the communication network is modeled as a connected, undirected graph graph $\mathcal{G}(t) \triangleq (\mathcal{V}, \mathcal{E}(t))$, where $\mathcal{V} := \{1, \ldots, M\}$ denotes the vertex set–the set of devices (does not change in time) and $\mathcal{E}(t) := \{(i, j) \,|\, i, j \in \mathcal{V}\}$ represents the set of edges–the communication links at the moment t; $(i, j) \in \mathcal{E}(t)$ iff there exists a communication link between devices i and j in moment t.

As mentioned earlier, the gossip protocol is the most popular communication procedures in decentralized setting. This approach uses a certain matrix W. Local vectors during communications are "weighted" by multiplication of a vector with W. The convergence of decentralized algorithms is determined by the properties of this matrix. Therefore, we introduce it:

Assumption 4. We call a matrix $W(t)$ a gossip matrix at the moment t if it satisfies the following conditions: 1) $W(t)$ is an $M \times M$ symmetric, 2) $W(t)$ is positive semi-definite, 3) the kernel of $W(t)$ is the set of constant vectors, 4) $W(t)$ is defined on the edges of the network at the moment t: $W_{ij}(t) \neq 0$ only if $i = j$ or $(i, j) \in \mathcal{E}(t)$.

Let $\lambda_1(W(t)) \geq \ldots \geq \lambda_M(W(t)) = 0$ the spectrum of $W(t)$, and let define condition number $\chi = \max_t \chi(W(t)) = \frac{\lambda_1(W(t))}{\lambda_{M-1}(W(t))}$. Note that in practice we will use not the matrix $W(t)$, but $\tilde{W}(t) = I - \frac{W(t)}{\lambda_1(W(t))}$. It is these matrices that are used in consensus algorithms [4]. To describe the convergence, we introduce $\rho = \max_t \lambda_2(\tilde{W}(t)) = \max_t \left[1 - \frac{\lambda_{M-1}(W(t))}{\lambda_1(W(t))}\right] = \max_t \left[1 - \frac{1}{\chi(W(t))}\right] = 1 - \frac{1}{\max_t \chi(W(t))} = 1 - \frac{1}{\chi}$.

3 Main part

We divide our contribution into two main parts, first we discuss lower bounds for decentralized saddle point problems over time-varying graphs. In the second part, we present an algorithm that achieves the lower bounds (up to logarithmic factors and numerical constants).

3.1 Lower bounds

Before presenting lower bounds, we must restrict the class of algorithms for which our lower bounds are valid. For this we introduce the following back-box procedure.

Definition 1. *Each device m has its own local memories $\mathcal{M}_m^x$ and $\mathcal{M}_m^y$ for the x- and y-variables, respectively–with initialization $\mathcal{M}_m^x = \mathcal{M}_m^y = \{0\}$. $\mathcal{M}_m^x$ and $\mathcal{M}_m^x$ are updated as follows:*

- ***Local computation:*** *Each device m computes and adds to its $\mathcal{M}_m^x$ and $\mathcal{M}_m^y$ a finite number of points x, y, each satisfying*

$$x \in span\{x' \ , \ \nabla_x f_m(x'', y'')\}, \quad y \in span\{y' \ , \ \nabla_y f_m(x'', y'')\}, \tag{7}$$

for given $x', x'' \in \mathcal{M}_m^x$ and $y', y'' \in \mathcal{M}_m^y$.

- ***Communication:*** *Based upon communication round among neighbouring nodes at the moment t, $\mathcal{M}_m^x$ and $\mathcal{M}_m^y$ are updated according to*

$$\mathcal{M}_m^x := span\left\{\bigcup_{(i,m)\in\mathcal{E}(t)} \mathcal{M}_i^x\right\}, \quad \mathcal{M}_m^y := span\left\{\bigcup_{(i,m)\in\mathcal{E}(t)} \mathcal{M}_i^y\right\}. \tag{8}$$

- ***Output:*** *The final global output at the current moment of time is calculated as:*

$$x \in span\left\{\bigcup_{m=1}^{M} \mathcal{M}_m^x\right\}, \quad y \in span\left\{\bigcup_{m=1}^{M} \mathcal{M}_m^y\right\}.$$

This definition includes all algorithms capable of making local gradient updates, as well as exchanging information with neighbors. Notice that the proposed oracle builds on [23] for minimization problems over networks.

Theorem 1. *For any L, μ and $\chi \leq 1$, there exist a saddle point problem in the form (1) with $\mathcal{Z} = \mathcal{R}^{2d}$ (where d is sufficiently large), and local functions f_m being L-smooth, μ-strongly-convex-strongly-concave, and a gossip matrices $W(t)$ over the connected (at each moment) graph $\mathcal{G}(t)$, satisfying Assumption 4 and with χ, such that any decentralized algorithm satisfying Definition 1 and using the gossip matrices $W(t)$ produces the following estimate on the global output $z = (x, y)$ after K communication rounds:*

$$\|z^K - z^*\|^2 = \Omega\left(\exp\left(-\frac{256\mu}{L-\mu}\cdot\frac{K}{\chi}\right)\|y^*\|^2\right).$$

Corollary 1. *In the setting of Theorem 1, the number of communication rounds required to obtain a ε-solution is lower bounded by*

$$\Omega\left(\chi\frac{L}{\mu}\cdot\log\left(\frac{\|y^*\|^2}{\varepsilon}\right)\right).$$

Additionally, we can get a lower bound for the number of local calculations on each of the devices:

$$\Omega\left(\frac{L}{\mu}\cdot\log\left(\frac{\|y^*\|^2}{\varepsilon}\right)\right).$$

Also we want to find lower bounds for the case of (non strongly) convex-concave problems, one can use regularization and consider the following objective function

$$f(x,y) + \frac{\varepsilon}{4D^2}\cdot\|x - x^0\|^2 - \frac{\varepsilon}{4D^2}\cdot\|y - y^0\|^2,$$

which is strongly-convex-strongly-concave with constant $\mu = \frac{\varepsilon}{4D^2}$, where ε is a precision of the solution and D is the diameter of the sets $\mathcal{X}$ and $\mathcal{Y}$. The resulting new SPP problem is solved to $\varepsilon/2$-precision in order to guarantee an accuracy ε in computing the solution of the original problem. Therefore, we can easily deduce the lower bounds for convex-concave case

$$\Omega\left(\chi\frac{LD^2}{\varepsilon}\right) \text{ communic. rounds} \quad \text{and} \quad \Omega\left(\frac{LD^2}{\varepsilon}\right) \text{ local computations.}$$

See Table 1 to compare with lower bounds for constant networks. The full proof of Theorem 1 one can find in the full version of our paper [for reviewers: here will be the link to the full version in arxiv.org, but we did not publish the paper before review].

Algorithm 1 Gossip Algorithm (`Gossip`)

Parameters: Vectors $z_1, ..., z_M$, communic. rounds H.
Initialization: Construct matrix $\mathbf{z}$ with rows $z_1^T, ..., z_M^T$.
Choose $\mathbf{z}^0 = \mathbf{z}$.
for $h = 0, 1, 2, \ldots, H$ **do**
$\mathbf{z}^{h+1} = \tilde{W}(h) \cdot \mathbf{z}^h$
end for
Output: rows $z_1, ..., z_M$ of $\mathbf{z}^{H+1}$.

3.2 Optimal algorithm

In this part, we present an Algorithm that achieves lower bounds (up to logarithmic terms). Our Algorithm uses an auxiliary procedure for communication. This is a classic procedure - Gossip Algorithm.

The essence of the `Gossip` is very simple. Initially, there are vectors z_1 and z_M, which are stored on their devices. Our goal is to get a vector close to the $\bar{z} = \frac{1}{M}\sum_{m=1}^{M} z_m$ vector on all devices. At each iteration, each device exchange local vectors with its neighbors, and then modify its local vector by averaging local vector and vectors of neighbors with weights from the matrix $W(h)$.

We are now ready to present our main algorithm. It is based on the classical method for smooth saddle point problems - Extra Step Method (Mirror Prox) [19,5]. With the right choice of H, we can achieve averaging of all vectors with good accuracy. In particular, we can assume that $z_1^k \approx \ldots \approx z_M^k$. For more details about the choice of H and a detailed analysis of the algorithm (taking into account that in the general $z_1^k \neq \ldots \neq z_M^k$), see in the full version of the paper [for reviewers: here will be the link to the full version in arxiv.org, but we did not publish the paper before review].

Algorithm 2 Time-Varying Decentralized Extra Step Method (`TVDESM`)

Parameters: Stepsize $\gamma \leq \frac{1}{4L}$, number of `Gossip` steps H.
Initialization: Choose $(x^0, y^0) = z^0 \in \mathcal{Z}$, $z_m^0 = z^0$.
for $k = 0, 1, 2, \ldots,$ **do**
Each machine m compute $\hat{z}_m^{k+1/2} = z_m^k - \gamma \cdot F_m(z_m^k)$
Communication: $\tilde{z}_1^{k+1/2}, \ldots, \tilde{z}_M^{k+1/2}$ =`Gossip`$(\hat{z}_1^{k+1/2}, \ldots, \hat{z}_M^{k+1/2}, H)$
Each machine m compute $z_m^{k+1/2} = \text{proj}_{\mathcal{Z}}(\tilde{z}_m^{k+1/2})$,
Each machine m compute $\hat{z}_m^{k+1} = z_m^k - \gamma \cdot F_m(z_m^{k+1/2})$
Communication: $\tilde{z}_1^{k+1}, \ldots, \tilde{z}_M^{k+1}$ =`Gossip`$(\hat{z}_1^{k+1}, \ldots, \hat{z}_M^{k+1}, H)$
Each machine m compute $z_m^{k+1} = \text{proj}_{\mathcal{Z}}(\tilde{z}_m^{k+1})$
end for

Theorem 2. *Let $\{z_m^k\}_{k\geq 0}^K$ denote the iterates of Algorithm 2 for solving problem (1) after K communication rounds. Let Assumptions 1(g,l) and 4 be satisfied. Then, if $\gamma \leq \frac{1}{4L}$, we have the following estimates in*

- *μ-strongly-convex–strongly-concave case (Assumption 2(s)):*

$$\mathbb{E}[\|\bar{z}^{k+1}-z^*\|^2] = \tilde{\mathcal{O}}\left(\|z^0 - z^*\|^2 \exp\left(-\frac{\mu K}{8L\sqrt{\chi}}\right)\right),$$

- *convex–concave case (Assumption 2 and 3):*

$$\mathbb{E}[gap(\bar{z}_{avg}^{k+1})] = \tilde{\mathcal{O}}\left(\frac{L\Omega_z^2\chi}{K}\right),$$

where $\bar{z}^t = \frac{1}{M}\sum\limits_{m=1}^{M} z_m^t$, $\bar{z}_{avg}^{k+1} = \frac{1}{M(k+1)}\sum\limits_{t=0}^{k}\sum\limits_{m=1}^{M} z_m^{t+1/2}$ *and*

$$gap(z) = \max_{y'\in\mathcal{Y}} f(x, y') - \min_{x'\in\mathcal{X}} f(x', y).$$

Corollary 2. *In the setting of Theorem 2, the number of communication rounds required for Algorithm 2 to obtain a ε-solution is upper bounded by*

$$\tilde{\mathcal{O}}\left(\chi\frac{L}{\mu}\cdot\log\left(\frac{\|z^0 - z^*\|^2}{\varepsilon}\right)\right)$$

in μ-strongly-convex–strongly-concave case and

$$\tilde{\mathcal{O}}\left(\chi\frac{LD^2}{\varepsilon}\right)$$

in convex-concave case. Additionally, one can obtain upper bounds for the number of local calculations on each of the devices:

$$\mathcal{O}\left(\frac{L}{\mu}\cdot\log\left(\frac{\|z^0 - z^*\|^2}{\varepsilon}\right)\right)$$

in μ-strongly-convex–strongly-concave case and

$$\mathcal{O}\left(\frac{LD^2}{\varepsilon}\right)$$

in convex-concave case.

4 Conclusion

In conclusion, we briefly summarize the contributions of this paper and discuss the directions for future work. Our findings consist of two parts: lower bounds and optimal (up to a logarithmic factor) algorithms.

First, we derived the lower bounds for the classes of convex-concave and strongly-convex-strongly-concave min-max problems over time-varying graphs. The graph is assumed to be connected at each communication round. However, we studied only one class of time-varying networks. Other classes are connected

to different assumptions on the network structure. In particular, in B-connected networks [18] the graph can be disconnected at some times, but the union of any B consequent graphs must be connected. Yet another possible assumption is the randomly changing graph with a contraction property of W in expectation [7]. Developing lower bounds for min-max problems for these two classes is an open question in decentralized optimization.

Second, we proposed a near-optimal algorithm with a gossip subroutine resulting in squared logarithmic factor. Developing an algorithm without an additional logarithmic factor would close the gap in theory and result in a more practical algorithm with less parameters to fine-tune. Possible directions for developing such an algorithm are generalizations of dual-based approaches for minimization [9,13] and gradient-tracking [18,13].

Finally, the comparison of our algorithm to existing works requires additional numerical experiments, which is left for future work.

References

1. Bertsekas, D.P., Tsitsiklis, J.N.: Parallel and distributed computation: numerical methods, vol. 23. Prentice hall Englewood Cliffs, NJ (1989)
2. Beznosikov, A., Dvurechensky, P., Koloskova, A., Samokhin, V., Stich, S.U., Gasnikov, A.: Decentralized local stochastic extra-gradient for variational inequalities. arXiv preprint arXiv:2106.08315 (2021)
3. Beznosikov, A., Samokhin, V., Gasnikov, A.: Local sgd for saddle-point problems. arXiv preprint arXiv:2010.13112 (2020)
4. Boyd, S., Ghosh, A., Prabhakar, B., Shah, D.: Randomized gossip algorithms. IEEE transactions on information theory **52**(6), 2508–2530 (2006)
5. Juditsky, A., Nemirovskii, A.S., Tauvel, C.: Solving variational inequalities with stochastic mirror-prox algorithm (2008)
6. Kempe, D., Dobra, A., Gehrke, J.: Gossip-based computation of aggregate information. In: 44th Annual IEEE Symposium on Foundations of Computer Science, 2003. Proceedings. pp. 482–491. IEEE (2003)
7. Koloskova, A., Loizou, N., Boreiri, S., Jaggi, M., Stich, S.U.: A unified theory of decentralized sgd with changing topology and local updates. arXiv preprint arXiv:2003.10422 (2020)
8. Kovalev, D., Gasanov, E., Richtárik, P., Gasnikov, A.: Lower bounds and optimal algorithms for smooth and strongly convex decentralized optimization over time-varying networks. arXiv preprint arXiv:2106.04469 (2021)
9. Kovalev, D., Shulgin, E., Richtárik, P., Rogozin, A., Gasnikov, A.: Adom: Accelerated decentralized optimization method for time-varying networks. arXiv preprint arXiv:2102.09234 (2021)
10. Li, H., Lin, Z.: Accelerated gradient tracking over time-varying graphs for decentralized optimization. arXiv preprint arXiv:2104.02596 (2021)
11. Liu, M., Zhang, W., Mroueh, Y., Cui, X., Ross, J., Yang, T., Das, P.: A decentralized parallel algorithm for training generative adversarial nets. arXiv preprint arXiv:1910.12999 (2019)
12. Liu, W., Mokhtari, A., Ozdaglar, A., Pattathil, S., Shen, Z., Zheng, N.: A decentralized proximal point-type method for saddle point problems. arXiv preprint arXiv:1910.14380 (2019)

13. Maros, M., Jaldén, J.: Panda: A dual linearly converging method for distributed optimization over time-varying undirected graphs. 2018 IEEE Conference on Decision and Control (CDC) pp. 6520–6525 (2018)
14. McDonald, R., Hall, K., Mann, G.: Distributed training strategies for the structured perceptron. In: Human language technologies: The 2010 annual conference of the North American chapter of the association for computational linguistics. pp. 456–464 (2010)
15. McMahan, B., Moore, E., Ramage, D., Hampson, S., y Arcas, B.A.: Communication-efficient learning of deep networks from decentralized data. In: Artificial Intelligence and Statistics. pp. 1273–1282. PMLR (2017)
16. Mukherjee, S., Chakraborty, M.: A decentralized algorithm for large scale min-max problems. In: 2020 59th IEEE Conference on Decision and Control (CDC). pp. 2967–2972 (2020). https://doi.org/10.1109/CDC42340.2020.9304470
17. Nedic, A., Ozdaglar, A.: Distributed subgradient methods for multi-agent optimization. IEEE Transactions on Automatic Control **54**(1), 48–61 (2009)
18. Nedić, A., Olshevsky, A., Shi, W.: Achieving geometric convergence for distributed optimization over time-varying graphs. SIAM Journal on Optimization **27**(4), 2597–2633 (2017)
19. Nemirovski, A.: Prox-method with rate of convergence o(1/t) for variational inequalities with lipschitz continuous monotone operators and smooth convex-concave saddle point problems. SIAM Journal on Optimization **15**, 229–251 (01 2004). https://doi.org/10.1137/S1052623403425629
20. Rogozin, A., Beznosikov, A., Dvinskikh, D., Kovalev, D., Dvurechensky, P., Gasnikov, A.: Decentralized distributed optimization for saddle point problems. arXiv preprint arXiv:2102.07758 (2021)
21. Rogozin, A., Gasnikov, A.: Projected gradient method for decentralized optimization over time-varying networks. arXiv preprint arXiv:1911.08527 (2019)
22. Rogozin, A., Uribe, C.A., Gasnikov, A.V., Malkovsky, N., Nedić, A.: Optimal distributed convex optimization on slowly time-varying graphs. IEEE Transactions on Control of Network Systems **7**(2), 829–841 (2019)
23. Scaman, K., Bach, F., Bubeck, S., Lee, Y.T., Massoulié, L.: Optimal algorithms for smooth and strongly convex distributed optimization in networks. arXiv preprint arXiv:1702.08704 (2017)
24. Shalev-Shwartz, S., Ben-David, S.: Understanding machine learning: From theory to algorithms. Cambridge university press (2014)
25. Ye, H., Luo, L., Zhou, Z., Zhang, T.: Multi-consensus decentralized accelerated gradient descent. arXiv preprint arXiv:2005.00797 (2020)