ADOM: Accelerated Decentralized Optimization Method for Time-Varying Networks

King Abdullah University of Science and Technology

Dmitry Kovalev[1] Egor Shulgin[1] Peter Richtárik[1] Alexander Rogozin[2] Alexander Gasnikov[2]

[1]King Abdullah University of Science and Technology (KAUST) [2]Moscow Institute of Physics and Technology (MIPT)

MIPT MOSCOW INSTITUTE OF PHYSICS AND TECHNOLOGY

Time-Varying Decentralized Minimization

SETUP: $\mathcal{G}^k := (\mathcal{V}, \mathcal{E}^k)$ – undirected connected networks, where

- $\mathcal{V} := \{1, \dots, n\}$ is a set of computing nodes,
- $\mathcal{E}^k \subset \mathcal{V} \times \mathcal{V}$ is a sequence of communication links.

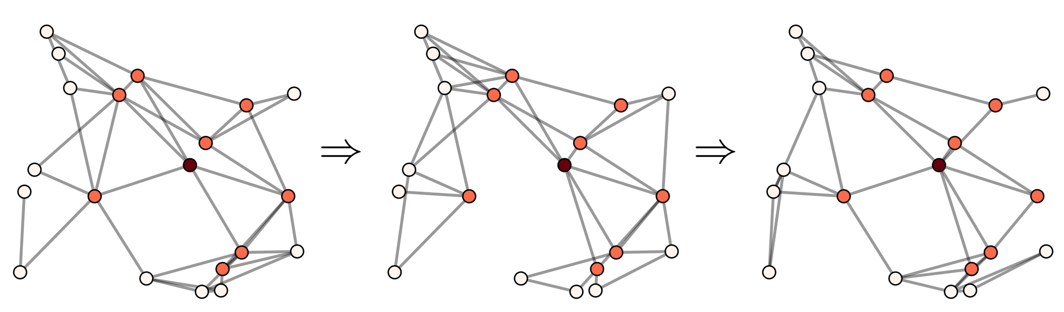


Figure 1: A sample time-varying network with $n = 20$ nodes.

Each node $i \in \mathcal{V}$ owns function $f_i \colon \mathbb{R}^d \to \mathbb{R}$, which is L-smooth and μ-strongly convex.

GOAL: Find solution of the minimization problem

$$\min_{x \in \mathbb{R}^d} \sum_{i \in \mathcal{V}} f_i(x). \tag{1}$$

Each node $i \in \mathcal{V}$ is allowed to calculate $\nabla f_i(x)$ and communicate $\mathcal{O}(1)$ vectors of size d with neighbors along the links $e \in \mathcal{E}^k$.

Problem Reformulation

Consider function $F \colon (\mathbb{R}^d)^{\mathcal{V}} \to \mathbb{R}$ defined by

$$F(x) := \sum_{i \in \mathcal{V}} f_i(x_i), \quad \text{where} \quad x = (x_1, \dots, x_n) \in (\mathbb{R}^d)^{\mathcal{V}}.$$

Consider also a sequence of $nd \times nd$ matrices

$$\mathbf{W}(k) := \hat{\mathbf{W}}(k) \otimes \mathbf{I},$$

where $\mathbf{I}$ is $d \times d$ identity matrix and $\hat{\mathbf{W}}(k)$ is an $n \times n$ matrix which satisfies the following properties:

1) $\hat{\mathbf{W}}(k)$ is symmetric positive semi-definite,
2) $\hat{\mathbf{W}}_{ij}(k) \neq 0$ if and only if $i = j$ or $(i,j) \in \mathcal{E}^k$,
3) $\ker \hat{\mathbf{W}}(k) = \operatorname{span}(\{(1, \dots, 1) \in \mathbb{R}^n\})$.

We are going to call $\mathbf{W}(k)$ a **gossip matrix.** Note that decentralized communication at time step k can be represented as multiplication of $\mathbf{W}(k)$ by vector $x = (x_1, \dots, x_n) \in (\mathbb{R}^d)^{\mathcal{V}}$:

$$y = (y_1, \dots, y_n) = \mathbf{W}(k)x \Rightarrow y_i \in \operatorname{span}(\{x_j : j \text{ is neighbor of } i\}).$$

Problem (1) can be reformulated as a **lifted problem with consensus constraints:**

$$\min_{x \in \mathcal{L}} F(x), \tag{1a}$$

where $\mathcal{L} := \{(x_1, \dots, x_n) \in (\mathbb{R}^d)^{\mathcal{V}} : x_1 = \dots = x_n\}$.
By $x^* := (\hat{x}, \dots, \hat{x}) \in (\mathbb{R}^d)^{\mathcal{V}}$ we denote the solution to Problem (1a), where $\hat{x} \in \mathbb{R}^d$ is the solution to Problem (1).

Dual Problem

Problem (1a) has an equivalent *dual formulation* of the form

$$\min_{z \in \mathcal{L}^\perp} F^*(z), \tag{2}$$

where F^* is the Fenchel transform of F and $\mathcal{L}^\perp \subset (\mathbb{R}^d)^{\mathcal{V}}$ is the orthogonal complement to the space $\mathcal{L}$, given as follows:

$$\mathcal{L}^\perp = \left\{ (z_1, \dots, z_n) \in (\mathbb{R}^d)^{\mathcal{V}} : \sum_{i=1}^n z_i = 0 \right\}.$$

Function $F^*(z)$ is $\frac{1}{\mu}$-smooth and $\frac{1}{L}$-strongly convex. Hence, problem (2) also has a unique solution, which we denote as $z^* \in \mathcal{L}^\perp$.

Communication as a Compression Operator

Let $\mathcal{Q}$ be a linear space. A mapping $\mathcal{C} \colon \mathcal{Q} \to \mathcal{Q}$ is called a *compression operator* if there exists $\delta \in (0, 1]$ such that

$$\|\mathcal{C}(z) - z\|^2 \le (1 - \delta)\|z\|^2 \quad \text{for all} \quad z \in \mathcal{Q}.$$

The following lemma shows that matrix-vector multiplication by gossip matrix $\mathbf{W}(k)$ is a contractive compression operator acting on the subspace $\mathcal{L}^\perp$.

Lemma (Main Idea)

Let $\sigma \in (0, 1/\lambda_{\max})$, $k \in \{0, 1, 2 \dots\}$. Then the following inequality holds for all $z \in \mathcal{L}^\perp$:

$$\|\sigma \mathbf{W}(k) z - z\|^2 \le (1 - \sigma\lambda_{\min}^+)\|z\|^2.$$

References:

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

[2] Marie Maros and Joakim Jaldén.
Panda: A dual linearly converging method for distributed optimization over time-varying undirected graphs.
In *2018 IEEE Conference on Decision and Control (CDC)*, pages 6520–6525. IEEE, 2018.

[3] Guannan Qu and Na Li.
Accelerated distributed nesterov gradient descent.
IEEE Transactions on Automatic Control, 2019.

[4] Huan Li, Cong Fang, Wotao Yin, and Zhouchen Lin.
A sharp convergence rate analysis for distributed accelerated gradient methods.
arXiv preprint arXiv:1810.01053, 2018.

[5] Haishan Ye, Luo Luo, Ziang Zhou, and Tong Zhang.
Multi-consensus decentralized accelerated gradient descent.
arXiv preprint arXiv:2005.00797, 2020.

[6] Alexander Rogozin, Cesar Uribe, Alexander Gasnikov, Nikolai Malkovskii, and Angelia Nedich.
Optimal distributed convex optimization on slowly time-varying graphs.
IEEE Transactions on Control of Network Systems, 2019.

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

Accelerated Algorithm with Guarantees

Our algorithm uses the dual oracle, and is based on a careful generalization of the **Projected Nesterov Gradient Descent**.

Algorithm 1 ADOM

1: **input:** $z^0 \in \mathcal{L}^\perp, m^0 \in (\mathbb{R}^d)^{\mathcal{V}}, \alpha, \eta, \theta, \sigma > 0, \tau \in (0,1)$
2: **set** $z_f^0 = z^0$
3: **for** $k = 0, 1, 2, \dots$ **do**
4: $\quad z_g^k = \tau z^k + (1 - \tau) z_f^k$
5: $\quad \Delta^k = \sigma \mathbf{W}(k)(m^k - \eta \nabla F^*(z_g^k))$
6: $\quad m^{k+1} = m^k - \eta \nabla F^*(z_g^k) - \Delta^k$
7: $\quad z^{k+1} = z^k + \eta\alpha(z_g^k - z^k) + \Delta^k$
8: $\quad z_f^{k+1} = z_g^k - \theta \mathbf{W}(k) \nabla F^*(z_g^k)$
9: **end for**

Method combines ideas of biased compression with error-feedback mechanism and acceleration.

Convergence of ADOM

Set parameters $\alpha, \eta, \theta, \sigma, \tau$ of Algorithm 1 to $\alpha = \frac{1}{2L}$, $\eta = \frac{2\lambda_{\min}^+\sqrt{\mu L}}{7\lambda_{\max}}$, $\theta = \frac{\mu}{\lambda_{\max}}$, $\sigma = \frac{1}{\lambda_{\max}}$, and $\tau = \frac{\lambda_{\min}^+}{7\lambda_{\max}}\sqrt{\frac{\mu}{L}}$. Then there exists $C > 0$, such that

$$\|\nabla F^*(z_g^k) - x^*\|^2 \le C\left(1 - \frac{\lambda_{\min}^+}{7\lambda_{\max}}\sqrt{\frac{\mu}{L}}\right)^k,$$

where $\lambda_{\min}^+$ and $\lambda_{\max}$ refer to bounds for the largest and to the smallest positive eigenvalue respectively

$$\lambda_{\min}^+ \le \lambda_{\min}^+(\hat{\mathbf{W}}(k)) \le \lambda_{\max}(\hat{\mathbf{W}}(k)) \le \lambda_{\max}$$

Comparison with Previous Methods

Table 1: A review of decentralized optimization algorithms capable of working in the time-varying network regime, with guarantees. Complexity terms highlighted in red represent the best known dependencies. Our method is the only algorithm with best known dependencies in all terms ($\kappa := L/\mu$, $\chi := \lambda_{\max}/\lambda_{\min}^+$).

Algorithm	Communication complexity
DIGing [1]	$\mathcal{O}\left(n^{1/2}\chi^2\kappa^{3/2}\log\frac{1}{\epsilon}\right)$
PANDA [2]	$\mathcal{O}\left(\chi^2\kappa^{3/2}\log\frac{1}{\epsilon}\right)$
Acc-DNGD [3]	$\mathcal{O}\left(\chi^{3/2}\kappa^{5/7}\log\frac{1}{\epsilon}\right)$
APM [4]	$\mathcal{O}\left(\chi\kappa^{1/2}\log^2\frac{1}{\epsilon}\right)$
Mudag [5]	$\mathcal{O}\left(\chi\kappa^{1/2}\log(\kappa)\log\frac{1}{\epsilon}\right)$
ADOM (Algorithm 1)	$\mathcal{O}\left(\chi\kappa^{1/2}\log\frac{1}{\epsilon}\right)$

ADOM achieves the new state-of-the-art rate for decentralized optimization over time-varying networks.

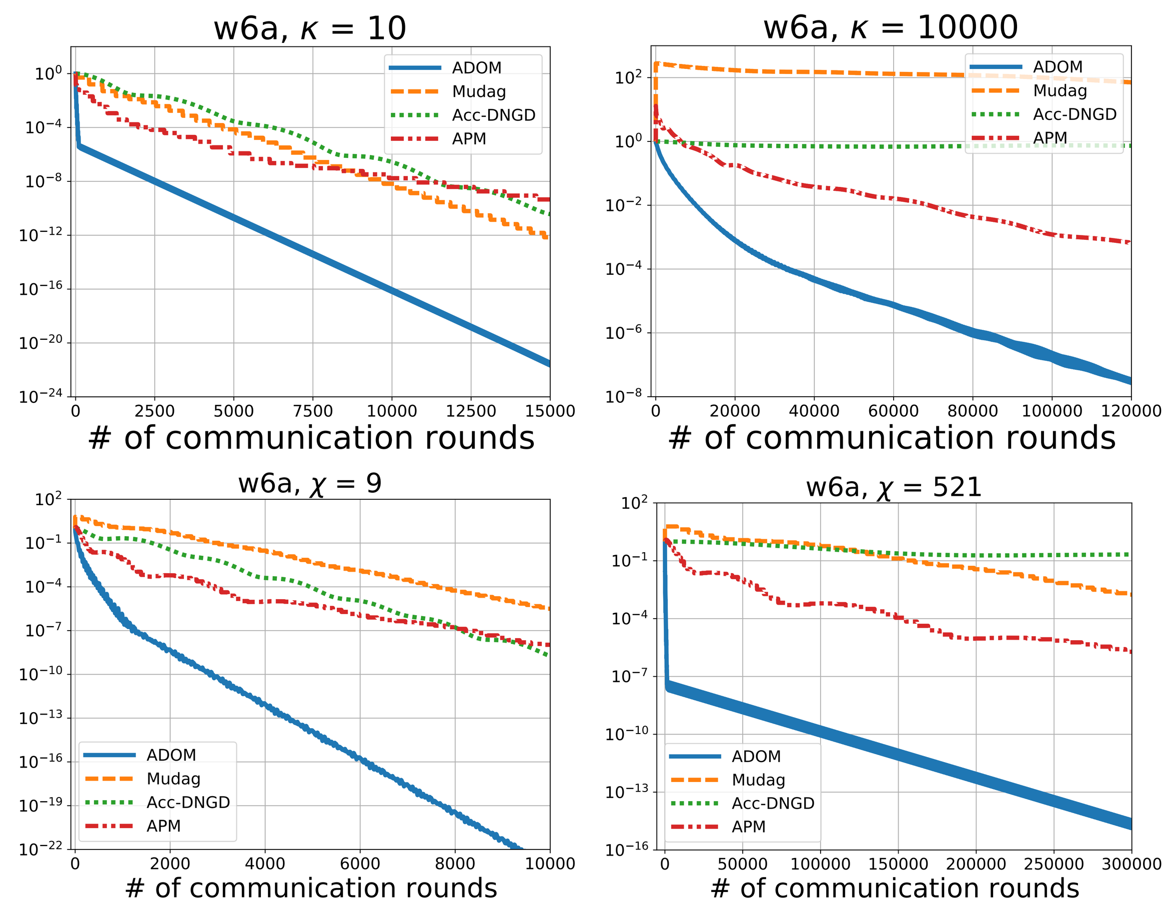
Numerical Experiments

We compare with the best previous methods on the logistic regression problem with ℓ_2 regularization:

$$f_i(x) = \frac{1}{m}\sum_{j=1}^m \log(1 + \exp(-b_{ij}a_{ij}^\top x)) + \frac{r}{2}\|x\|^2.$$

To simulate a time-varying network, we use geometric random graphs and choose matrix $\mathbf{W}(k)$ as the Laplacian. ADOM needs dual gradients $\nabla F^*(z_g^k)$, which are calculated inexactly using $T (\le 3$ sufficient in our case) iterations of gradient method for problem:

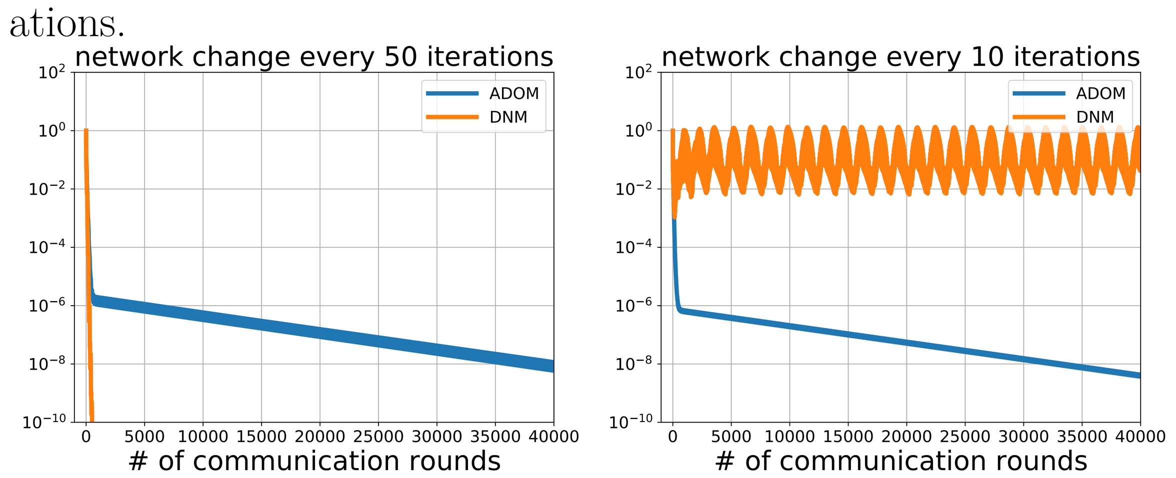
$$\nabla F^*(z_g^k) = \arg\min_{x \in (\mathbb{R}^d)^{\mathcal{V}}} F(x) - \langle x, z_g^k \rangle.$$



Comparison of ADOM, Mudag, Acc-DNGD and APM on *w6a* ($n = 17188, d = 300$) LIBSVM dataset. **First row:** $\kappa \in \{10, 10^4\}$ and networks with $\chi \approx 30$. **Second row:** $\kappa = 100$ and networks with $\chi \in \{9, 521\}$.

ADOM converges linearly and outperforms all known algorithms for every set of parameters.

Next we compare against the Distributed Nesterov Method (DNM) [6], which has an $\sqrt{\kappa}$ dependence. We use synthetic data and switch between two geometric graphs ($\chi \approx 400$) every t iterations.



Comparison of ADOM and DNM [6] on a problem with $\kappa = 30$ and number of features $d = 40$.

ADOM always converges, unlike DNM.

More experimental results (including real networks) in the paper [7].