COMPUTER RESEARCH AND MODELING
2024 VOL. 16 NO. 7 P. 1813–1827
DOI: 10.20537/2076-7633-2024-16-7-1813-1827

Ки&М

SPECIAL ISSUE

UDC: 004

Communication-efficient solution of distributed variational inequalities using biased compression, data similarity and local updates

R. E. Voronov[1,a], E. M. Maslennikov[2], A. N. Beznosikov[1,2,3]

[1]Innopolis University,
1 Universitetskaya st., Innopolis, Russia
[2]Moscow Institute of Physics and Technology,
1A Kerchenskaia st., Moscow, 117303, Russia
[1]Ivannikov Institute for System Programming of the Russian Academy of Sciences,
25 A. Solzhenitsyna st., Moscow, 109004, Russia

E-mail: [a] porludom@mail.ru

Received 29.10.2024, after completion — 12.11.2024
Accepted for publication 25.11.2024

Variational inequalities constitute a broad class of problems with applications in a number of fields, including game theory, economics, and machine learning. Today's practical applications of VIs are becoming increasingly computationally demanding. It is therefore necessary to employ distributed computations to solve such problems in a reasonable time. In this context, workers have to exchange data with each other, which creates a communication bottleneck. There are three main techniques to reduce the cost and the number of communications: the similarity of local operators, the compression of messages and the use of local steps on devices. There is an algorithm that uses all of these techniques to solve the VI problem and outperforms all previous methods in terms of communication complexity. However, this algorithm is limited to unbiased compression. Meanwhile, biased (contractive) compression leads to better results in practice, but it requires additional modifications within an algorithm and more effort to prove the convergence. In this work, we develop a new algorithm that solves distributed VI problems using data similarity, contractive compression and local steps on devices, derive the theoretical convergence of such an algorithm, and perform some experiments to show the applicability of the method.

Keywords: variational inequalities, biased compression, data similarity, local updates

Citation: *Computer Research and Modeling*, 2024, vol. 16, no. 7, pp. 1813–1827.

This research was funded by the Russian Science Foundation (project No. 23-11-00229).

КОМПЬЮТЕРНЫЕ ИССЛЕДОВАНИЯ
И МОДЕЛИРОВАНИЕ 2024 Т. 16 № 7 С. 1813–1827
DOI: 10.20537/2076-7633-2024-16-7-1813-1827

Ки&М

СПЕЦИАЛЬНЫЙ ВЫПУСК

УДК: 004

Решение распределенных вариационных неравенств с использованием смещенной компрессии, похожести данных и локальных обновлений

Р. Е. Воронов[1,a], Е. М. Масленников[2], А. Н. Безносиков[1,2,3]

[1]Университет Иннополис,
Россия, г. Иннополис, ул. Университетская, д. 1
[2]Московский физико-технический институт,
Россия, 117303, г. Москва, ул. Керченская, д. 1А
[3]Институт системного программирования им. В. П. Иванникова,
Россия, 109004, г. Москва, ул. А. Солженицына, д. 25

E-mail: [a] porludom@mail.ru

Получено 29.10.2024, после доработки — 12.11.2024
Принято к публикации 25.11.2024

Вариационные неравенства представляют собой широкий класс задач, имеющих применение во множестве областей, включая теорию игр, экономику и машинное обучение. Однако, методы решения современных вариационных неравенств становятся все более вычислительно требовательными. Поэтому растет необходимость использовать распределенных подходов для решения таких задач за разумное время. В распределенной постановке вычислительным устройствам необходимо обмениваться данными друг с другом, что является узким местом. Существует три основных приема снижения стоимости и количества обменов данными: использование похожести локальных операторов, сжатие сообщений и применение локальных шагов на устройствах. Известен алгоритм, который использует эти три техники одновременно для решения распределенных вариационных неравенств и превосходит все остальные методы с точки зрения коммуникационных затрат. Однако этот метод работает только с так называемыми несмещенными операторами сжатия. Между тем использование смещенных операторов приводит к лучшим результатам на практике, но требует дополнительных модификаций алгоритма и больших усилий при доказательстве сходимости. В этой работе представляется новый алгоритм, который решает распределенные вариационные неравенства, используя похожесть локальных операторов, смещенное сжатие и локальные обновления на устройствах; выводится теоретическая сходимость такого алгоритма и проводятся эксперименты.

Ключевые слова: вариационные неравенства, смещенное сжатие, похожесть данных, локальные обновления

Исследование выполнено за счет гранта Российского научного фонда (проект № 23-11-00229).

Introduction

Variational inequalities

Definition of VIs. Variational Inequalities (VIs) are an important area of research with many applications. We consider the following definition of the VI problem:

$$\text{Find } z^* \text{ such that } F(z^*) = 0, \ \forall z \in \mathbb{R}^d, \tag{1}$$

where $F\colon \mathbb{R}^d \to \mathbb{R}^d$ is an operator. To illustrate the extensiveness of VIs, we provide some examples:

• **Minimization.** We consider the following minimization problem:

$$\min_{z\in\mathbb{R}^d} f(z). \tag{2}$$

If we take $F(z) = \nabla f(z)$ in (1), then we try to find any stationary point of the function f. If $f(z)$ is convex, then the solution of (1) is the solution of (2).

When it comes to machine learning, the most widely used example is the empirical risk minimization problem [Shalev-Shwartz, Ben-David, 2014]:

$$\min_{w\in\mathbb{R}^d} \frac{1}{N}\sum_{i=1}^{N} l(g(w, a_i), b_i) + \frac{\lambda}{2}\|w\|^2, \tag{3}$$

where $g(w, a_i)$ is the output of the machine learning model (e. g. linear regression model or neural network) with parameters w on the input a_i, b_i is the label for the object a_i, l is the loss function, and N is the number of samples.

It is important to note that algorithms designed for minimization problems are not suitable for solving VI problems [Goodfellow, 2016]. The reason is that in some situations these algorithms do not give optimal convergence guarantees or even diverge. This is because VI problems are more general and more complicated. One of the examples of such complex problems that we cannot represent as minimization problem is provided below.

• **Saddle point problem.** The problem is to find a saddle point:

$$\min_{x\in\mathbb{R}^{d_x}} \max_{y\in\mathbb{R}^{d_y}} f(x, y). \tag{4}$$

We can formulate this problem as the VI problem with $z = [x, y]$ and $F(z) = F(x, y) = [\nabla_x f(x, y), -\nabla_y f(x, y)]$, $x \in \mathbb{R}^{d_x}$, $y \in \mathbb{R}^{d_y}$. For the convex-concave function f, the solution of (4) and the solution of (1) are equivalent. Saddle point problems have important applications in economics and game theory [Von Neumann, Morgenstern, 1944; Facchinei, Pang, 2003]. They are also used in machine learning: adversarial settings [Madry et al., 2018], GANs training [Goodfellow et al., 2014; Gidel et al., 2018; Chavdarova et al., 2019; Daskalakis et al., 2017], and reinforcement learning [Jin, Sidford, 2020].

As a motivating example, one can extend the empirical risk minimization problem (3). Many machine learning models, including neural networks [Goodfellow, Shlens, Szegedy, 2014], are vulnerable to adversarial examples, where nearly indistinguishable inputs force the model to produce very different outputs. To improve the model's resistance to adversarial attacks, one can formulate the following saddle point problem [Madry et al., 2018]:

$$\min_{w\in\mathbb{R}^{d_w}} \max_{\|r_i\|\leqslant R} \frac{1}{N}\sum_{i=1}^{N} l(f(w, a_i + r_i), b_i) + \frac{\lambda}{2}\|w\|^2 - \frac{\beta}{2}\|r\|^2, \tag{5}$$

where r_i is the adversarial noise applied to the input a_i.

Distributed setting

Distributed setting. Today's industrial tasks require a lot of computation and memory. For example, modern supervised machine learning models require a lot of training data to solve complex real-world problems. Furthermore, these models have many parameters, which makes them computationally expensive. Therefore, we consider a centralized distributed setting, in which the operator F is spread across n devices. The first device is chosen as the server. All other devices can only communicate with this selected server. Each device has its own local $F_m(z)\colon \mathbb{R}^d \to \mathbb{R}^d$, $m \in \overline{1, n}$. Then, we can express $F(z)$ from (1) in the following manner:

$$F(z) = \frac{1}{n}\sum_{m=1}^{n} F_m(z). \tag{6}$$

In the empirical risk minimization problem (3) or its adversarial modification (5), we can distribute training samples across n devices to train a machine learning model. Then, each $F_m(z)$ is the gradient of the loss on the training data on the mth device.

Since the devices have to communicate with the server, this leads to the communication bottleneck [Konečnỳ, 2016]. We cannot afford to send a lot of data, nor can we afford to send anything very often. Therefore, it is important now to solve the problem with the least communication complexity.

Compression. The first way to reduce the communication complexity is to compress the information being sent. As a result, devices communicate frequently but send less information. The first theoretical works (e. g. [Alistarh et al., 2017]) on gradient compression explored the following unbiased compression operators $Q\colon \mathbb{R}^d \to \mathbb{R}^d$:

$$\mathbb{E}[Q(z)] = z, \quad \mathbb{E}\left[\|Q(z)\|^2\right] \leqslant c\|z\|^2, \tag{7}$$

where $c \geqslant 1$. Various methods that use unbiased compression can be found in [Mishchenko et al., 2019; Horváth et al., 2019; Gorbunov et al., 2022]. In our work, we consider more general biased compressors $C\colon \mathbb{R}^d \to \mathbb{R}^d$:

$$\mathbb{E}\left[\|C(z) - z\|^2\right] \leqslant \left(1 - \frac{1}{\omega}\right)\|z\|^2, \tag{8}$$

where $\omega \geqslant 1$. Moreover, we introduce $\phi \in (0, 1]$ which indicates the compression ratio. To illustrate, the Top K operator, which removes all but the largest absolute K components, has $\phi = \frac{K}{d}$. As demonstrated in [Vogels, Karimireddy, Jaggi, 2019], algorithms that use biased compression tend to outperform those that utilize unbiased compression in practice, despite the theoretical advantage of the latter. However, if one replaces the unbiased compression with a biased one in simple distributed algorithms such as gradient descent without any additional modifications, then it may stop converging [Beznosikov et al., 2023b]. The concept of error feedback was introduced in [Seide et al., 2014] to address this problem. Modern results and techniques on biased compression are presented in [Stich, Cordonnier, Jaggi, 2018; Karimireddy et al., 2019; Vogels, Karimireddy, Jaggi, 2019; Stich, Karimireddy, 2020; Richtárik, Sokolov, Fatkhullin, 2021].

Local updates. The orthogonal to compression way to save communications is to perform some updates of variables locally. Consequently, there is a reduction in the amount of communication rounds between devices, but the information transmitted is full. The concept of local method comes from [Mangasarian, 1995; Zinkevich et al., 2010]. One of the earliest methods for solving distributed minimization problems is LocalSGD, the analysis of which is presented in [Khaled, Mishchenko, Richtárik, 2020]. To solve some problems of the basic local method, the authors of [Karimireddy et al., 2020] proposed SCAFFOLD which employs variance reduction via regularization. Further

modifications of LocalSGD can be found in [Li et al., 2020; Mishchenko et al., 2022]. It is important to note that local steps do not significantly reduce the total communication complexity [Woodworth, Patel, Srebro, 2020; Woodworth et al., 2021]. It was observed that data similarity is necessary to fully utilize the power of local computations.

Data similarity. There are different ways to define the similarity in distributed learning, e. g. the similarity of gradients of local functions. However, we consider the similarity of Hessians of $\{f_m\}_{m=1}^n$:

$$\left\|\nabla^2 f_i(z) - \nabla^2 f_j(z)\right\| \leqslant \delta, \quad \forall i,\ j \in \overline{1,\ n}. \tag{9}$$

In the empirical risk minimization problems (3) or its adversarial modification (5), when training data are i.i.d. across devices, $\{f_m\}_{m=1}^n$ are statistically similar to each other. This is reflected in $\delta \sim \widetilde{O}\left(\frac{1}{\sqrt{b}}\right)$ or even $\delta \sim \widetilde{O}\left(\frac{1}{b}\right)$, where b is the size of the local samples [Hendrikx et al., 2020]. The work [Shamir, Srebro, Zhang, 2014] was the first to use the Hessian similarity of local functions in their method DANE for minimization problems. The optimal algorithms for quadratic and general cases that employ similarity were introduced in [Yuan, Li, 2019] and [Kovalev et al., 2022], respectively. A survey of another algorithm that deals with data similarity for minimization problem can be found in [Hendrikx et al., 2020].

Combinations. In the context of minimization problems, [Basu et al., 2019; Nadiradze et al., 2019; Condat, Agarsky, Richtárik, 2022; Malinovsky, Yi, Richtárik, 2022] proposed different algorithms that use compression and local steps. In [Szlendak, Tyurin, Richtárik, 2021], the authors proposed the algorithm that combines compression and data similarity. However, there are no algorithms for minimization problems that combine compression, local steps and data similarity.

In the preceding section, we provided a concise overview of methods for distributed minimization probems. In the next section, we discuss more closed results for the case of distributed VI problems.

Related works

There are many different algorithms that solve distributed VI problems. We provide an overview of some of these algorithms below. For a comprehensive summary of these algorithms, please refer to Table 1.

Compression. In [Beznosikov et al., 2021b], the authors presented MASHA that employs either biased or unbiased compression to solve distributed VI problems.

Local methods. The first methods for solving VI problems utilizing local updates were adaptations of techniques originally developed for minimization problems. [Deng, Mahdavi, 2021; Hou et al., 2021] adapted LocalSGD for VI problems, [Hou et al., 2021] adapted SCAFFOLD for saddle point problems, and [Beznosikov et al., 2021a; Beznosikov et al., 2023a] added local steps to Extra Gradient method.

Local steps and data similarity. Both Extra Gradient Sliding [Kovalev et al., 2022] and SMMDS [Beznosikov et al., 2021c] utilize local updates and data similarity for distributed VI problems.

Compression and data similarity. The authors of [Beznosikov, Gasnikov, 2022] proposed the algorithm that uses compression and data similarity. As previously stated, one cannot expect optimal results with data similarity without incorporating local updates into the algorithm, and this work is no exception.

Starting point of our work. Finally, [Beznosikov, Takác, Gasnikov, 2024] presented Three Pillars Algorithm, which employs similarity, local updates and compression. Moreover, they derived the lower bounds on communication complexity. Three Pillars Algorithm achieves these lower bounds. However, the method only employs unbiased compression. In our work, we add support for biased compression to this algorithm.

Table 1. Comparison of different algorithms for solving distributed VI problems. Communication complexity is the number of transmitted information by 1 device to achieve precision ϵ. Notation: L — Lipschitz constant of F, μ — strong monotonicity constant of F, δ — similarity parameter, n — number of devices, ϵ — precision of solution, ϕ — compression ratio, ω — parameter that controls variance of biased compressor

Algorithm	Local steps	Similarity	Compression	Comm-n complexity
Extra Gradient				$O\left(\frac{L}{\mu}\log\frac{1}{\epsilon}\right)$
MASHA [Beznosikov et al., 2021b]			Any	$O\left(\frac{L}{\sqrt{n}\mu}\log\frac{1}{\epsilon}\right)$
OMASHA [Beznosikov, Gasnikov, 2022]		✓	Unbiased	$O\left(\left[\frac{L}{n\mu}+\frac{\delta}{\sqrt{n}\mu}\right]\log\frac{1}{\epsilon}\right)$
SCAFFOLD-S [Hou et al., 2021]	✓			$O\left(\frac{L^2}{\mu^2}\log\frac{1}{\epsilon}\right)$
SCAFFOLD-Catalyst-S [Hou et al., 2021]	✓			$O\left(\frac{L}{\mu}\log^2\frac{1}{\epsilon}\right)$
SMMDS [Beznosikov et al., 2021c]	✓	✓		$O\left(\left[1+\frac{\delta}{\mu}\right]\log\frac{1}{\epsilon}\right)$
Extra gradient sliding [Kovalev et al., 2022]	✓	✓		$O\left(\left[1+\frac{\delta}{\mu}\right]\log\frac{1}{\epsilon}\right)$
TPA [Beznosikov, Takác, Gasnikov, 2024]	✓	✓	Unbiased	$O\left(\left[1+\frac{\delta}{\sqrt{n}\mu}\right]\log\frac{1}{\epsilon}\right)$
This paper	✓	✓	Any	$O\left(\left[1+\frac{\sqrt{\phi}\omega\delta}{\mu}\right]\log\frac{1}{\epsilon}\right)$

Our contribution

New algorithm. We present a new algorithm (see Algorithm 1) that effectively solves the distributed VI problem using three approaches: biased compression, data similarity and local updates. The algorithm is developed by incorporating an error feedback technique into Three Pillars Algorithm.

Theoretical convergence. We derive the theoretical convergence of the new algorithm for VIs with a Lipschitz strongly monotone operator under the similarity (Assumption 3). Depending on the compressor, the communication complexity is equal to or worse than the guarantees for Three Pillars Algorithm [Beznosikov, Takác, Gasnikov, 2024]. In the case of identical compression operators, we replicate the guarantees for SMMDS [Beznosikov et al., 2021c], Extra Gradient Sliding [Kovalev et al., 2022], and outperform the guarantees for OMASHA [Beznosikov, Gasnikov, 2022].

Experiments. We conduct some experiments to check the theoretical results. The results demonstrate that, despite the theoretical equivalence or inferiority of our algorithm in comparison to Three Pillars Algorithm, the new method is better in practice. This is a usual case when algorithms with biased compression are compared to those with unbiased compression in practice [Beznosikov et al., 2023b]. Moreover, our algorithm outperforms all other algorithms presented in Table 1.

Assumptions

The goal is to solve the distributed VI problem (1). In order to derive the theoretical convergence, it is necessary to consider this VI problem under the following assumptions.

Assumption 1 (Lipschitzness). *Each operator F_m is L-Lipschitz continuous, i. e. for all $u, v \in \mathbb{R}^d$ we have*

$$\|F_m(u) - F_m(v)\| \leqslant L\|u - v\|.$$

For the problems (2), (4), L-Lipschitzness of the operator means L-smoothness of functions $f(z)$, $f(x, y)$, respectively.

Assumption 2 (strong monotonicity). *The operator F is μ-strongly monotone on $\mathbb{R}^d$, i. e. for all $u, v \in \mathbb{R}^d$ we have*

$$\langle F(u) - F(v), u - v\rangle \geqslant \mu\|u - v\|^2.$$

Strong monotonicity in (2), (4) means strong convexity of $f(z)$, and strong convexity, strong concavity of $f(x, y)$, respectively.

Assumption 3 (δ-relatedness in mean). *For any j operators $\{F_i - F_j\}_{i=1}^n$ is δ-Lipschitz continuous in mean, i. e. for all $u, v \in \mathbb{R}^d$ we have*

$$\frac{1}{n}\sum_{i=1}^{n}\|F_i(u) - F_j(u) - F_i(v) + F_j(v)\|^2 \leqslant \delta^2\|u - v\|^2.$$

This is the generalized data similarity assumption from (9). For example, in the case of solving the saddle point problem (2), this assumption means that $\left\|\nabla^2_{xx} f_i(x, y) - \nabla^2_{xx} f_j(x, y)\right\| \leqslant \delta$, $\left\|\nabla^2_{xy} f_i(x, y) - \nabla^2_{xy} f_j(x, y)\right\| \leqslant \delta$, $\left\|\nabla^2_{yy} f_i(x, y) - \nabla^2_{yy} f_j(x, y)\right\| \leqslant \delta$. In the case of minimization, it is practically equivalent to (9).

We also assume that forwarding from devices to the server takes considerably longer than sending from the server to the devices [Kairouz et al., 2021]. Therefore, from the perspective of communication complexity, we are only interested in sending from devices to the server.

Biased Three Pillars algorithm

Description of the algorithm

FBF algorithm with variance reduction. Let us build the algorithm from scratch. We start with the Forward-Backward-Forward algorithm with variance reduction (VR) from [Alacaoglu, Malitsky, 2021]. The FBF algorithm is the optimal proximal method for solving the VI problems. Moreover, it requires a smaller number of proximal/resolvent operators than Extra Gradient. Variance reduction allows us to use the stochastic realizations of operators effectively [Alacaoglu, Malitsky, 2021] and consists of two main parts: updates of reference point and negative momentum. In the following snippet, $p \in (0, 1]$ is the probability of updating the reference point, γ is the learning rate, and $\tau \in (0, 1)$ is the negative momentum parameter. Here is the one iteration of the FBF algorithm with variance reduction [Alacaoglu, Malitsky, 2021]:

$$z^{k+1/2} = J\left(z^k + \tau\left(m^k - z^k\right) - \gamma F\left(m^k\right)\right), \tag{10}$$

$$z^{k+1} = z^{k+1/2} - \gamma\left(F_{\xi_k}\left(z^{k+1/2}\right) - F_{\xi_k}\left(m^k\right)\right), \tag{11}$$

$$m^{k+1} = \begin{cases} z^{k+1}, & \text{with probability } p, \\ m^k, & \text{with probability } 1 - p. \end{cases} \tag{12}$$

The following subsection will address the question of how this algorithm can be used to solve our distributed problem.

Adaptation to our setting. In a distributed setting, starting with initialization, all devices know the latest reference point and the value of the operator at this point. To derive the resolvent operator from (10), we change the representation of the operator (6) in (1) a little bit by extracting the regularization:

$$F(z) = \frac{1}{n}\sum_{m=1}^{n} F_m(z) = q(z) + r(z) = \left[\frac{1}{n}\sum_{m=1}^{n} F_m(z) - F_1(z)\right] + [F_1(z)].$$

Algorithm 1. Three Pillars algorithms with biased compression

Parameters: stepsizes γ and η, momentum τ, probability $p \in (0, 1]$, number of local steps H, number of iterations K

Initialization: Choose $z^0 = m^0 = \left(x^0, y^0\right) \in \mathbb{R}^{d_x+d_y}$, local errors $e_m^0 = 0$

Server broadcasts $z^0 = m^0$ to devices;

Devices in parallel compute $F_m\left(m^0\right)$ and send them to the server;

Server broadcasts $F\left(m^0\right) = \frac{1}{n}\sum_{m=0}^{n-1} F_m\left(m^0\right)$ to devices;

for $k = 0, 1, \ldots, K-1$ **do**

  Server takes $u_0^k = z^k$;

  // Here we solve local problem using Extra Gradient. This is the first step of FBF algorithm with VR.

  for $t = 0, 1, \ldots, H-1$ **do**

    Server computes

    $$u_{t+1/2}^k = u_t^k - \eta\left(F_1\left(u_t^k\right)\right) + \frac{1}{\gamma}\left(u_t^k - z^k - \tau\left(m^k - z^k\right) + \gamma\left(F\left(m^k\right) - F_1\left(m^k\right)\right)\right);$$

    Server updates

    $$u_{t+1}^k = u_t^k - \eta\left(F_1\left(u_{t+1/2}^k\right)\right) + \frac{1}{\gamma}\left(u_{t+1/2}^k - z^k - \tau\left(m^k - z^k\right) + \gamma\left(F\left(m^k\right) - F_1\left(m^k\right)\right)\right);$$

  end

  Server broadcasts u_H^k and $F_1\left(u_H^k\right)$;

  // The parallel work of devices.

  for $m = 1, ..., n$ **do**

    Device m computes message

    $$s_m^k = Q_m\left(F_m\left(m^k\right) - F_1\left(m^k\right) - \left(F_m\left(u_H^k\right) - F_1\left(u_H^k\right)\right) + e_m^k\right);$$

    Device m updates $e_m^{(k+1)} = e_m^k + F_m\left(m^k\right) - F_1\left(m^k\right) - \left(F_m\left(u_H^k\right) - F_1\left(u_H^k\right)\right) - s_m^k$;

    // The data that is lost during this compression process is subsequently accumulated with data that was lost in previous compressions.

    Device m sends its message s_m^k to the server;

  end

  // The second step of FBF algorithm with VR.

  Server updates $z^{k+1} = u_H^k + \frac{\gamma}{n}\sum_{i=0}^{n-1} s_i^k$;

  // Here we decide whether we do another synchronization or not.

  // The third step of FBF Algorithm with VR.

  Server updates $m^{k+1} = \begin{cases} z^{k+1}, & \text{with probability } p, \\ m^k, & \text{with probability } 1-p; \end{cases}$

  if $m^{k+1} = z^k$ **then**

    Server broadcasts m^{k+1} to devices;

    Devices in parallel compute $F_m\left(m^{k+1}\right)$ and send them to the server;

    Server broadcasts $F\left(m^{k+1}\right) = \frac{1}{n}\sum_{m=0}^{n-1} F_m\left(m^{k+1}\right)$ to devices;

  end

end

We apply the FBF algorithm with VR to the VI problem with the main operator

$$\left[\frac{1}{n}\sum_{m=1}^{n} F_m(z) - F_1(z)\right]$$

instead of

$$\frac{1}{n}\sum_{m=1}^{n} F_m(z).$$

The resolvent operator J generated by the regularizer $F_1(z)$ solves the local problem $F_1(z) + \frac{1}{\gamma}(z - z_s) = 0$, where $z_s = (1-\tau)z^k + \tau m^k - \gamma\left(F\left(m^k\right) - F_1\left(m^k\right)\right)$ is taken from (10) of the FBF algorithm with VR for the new main operator. We can compute the resolvent locally on the first device because no communication is required to find the solution, we need only the reference point (known to all devices) and the operator F_1. To find a solution of this local problem, we run H iterations of Extra Gradient in Lines 3–6, with the output solution named u_H^k.

In the second line (11) of the snippet, the difference of stochastic realizations of operators is used. In our algorithm, we consider the average of the compressed differences $\left\{F_m\left(u_H^k\right) - F_1\left(u_H^k\right) - \left(F_m\left(m^k\right) - F_1\left(m^k\right)\right)\right\}_{m=1}^{n}$ in Lines 8–13. Similarly to FBF algorithm with VR, variance reduction allows us to use the compression effectively [Beznosikov et al., 2021b]. Those differences are similar to those in the snippet since we use

$$\left[\frac{1}{n}\sum_{m=1}^{n} F_m(z) - F_1(z)\right]$$

as the main operator, as stated previously.

Since the biased compression operator is considered, we employ an error feedback technique [Seide et al., 2014]. The error feedback parts are highlighted by orange. Each time device m sends a compressed message, we add the difference between the real and compressed message to e_m in Line (10). This variable contains everything we have not sent during communication rounds. We then use the value of e_m to correct the message during the next communication round in Line (9).

Finally, each change of reference point (12) results in the additional communication round when the full message is sent from every device (Lines 14–19). With high p, we have more additional communications but better convergence in terms of iterations because the reference point is not outdated. And vice versa, the smaller p is, the cheaper one iteration is in terms of communication. But this leads to an increase in the number of iterations and hence the number of communications. We will discuss the optimal choice of p later.

Convergence analysis

Below one can see the theoretical convergence of the algorithm and its analysis.

Theorem 1. *Consider* $\{z^k\}_{k\geqslant 0}$ *as a sequence that defines the iterates of Algorithm* 1 *to get the solution of the problem* (1) *under Assumptions* 1, 2, 3. *With the stepsizes* γ, η *and the momentum* τ *chosen as follows*:

$$\tau = p \leqslant \frac{1}{8}, \quad \gamma = \min\left\{\frac{p}{6\mu}, \frac{\sqrt{p}}{12\delta}, \frac{1}{L}\cdot\left(\frac{H}{4\log\frac{220L\omega}{\mu p}} - 1\right), \frac{\sqrt{p}}{\sqrt{128}\omega\delta}\right\},$$
$$\eta = \frac{1}{4\left(L + \frac{1}{\gamma}\right)}, \quad H \geqslant 8\log\frac{220L\omega}{\mu p};$$

we get the following convergence:

$$\mathbb{E}\left\|\widetilde{z}^K - z^*\right\|^2 \leqslant \left(1 - \frac{\gamma\mu}{2}\right)^K \cdot 2\mathbb{E}\left\|z^0 - z^*\right\|^2,$$

where

$$\widetilde{z}^{k+1} = z^{k+1} + \frac{\gamma}{n}\sum_{i=1}^{n} e_i^{k+1}.$$

One can see the proof of Theorem 1 in the supplementary material on the journal's website.

Discussion. From the theorem above one can derive the number of outer iterations K to obtain some accuracy ϵ:

$$K = O\left(\frac{1}{\gamma\mu}\log\left(\frac{\left\|z^0 - z^*\right\|^2}{\epsilon}\right)\right).$$

By substituting all learning rates from the theorem and noting that $\omega > 1$, we derive that

$$K = O\left(\left[\frac{1}{p} + \frac{L}{H\mu} + \frac{\omega\delta}{\sqrt{p}\mu}\right]\log\left(\frac{\left\|z^0 - z^*\right\|^2}{\epsilon}\right)\right).$$

The question is what p and H to choose. We start from counting the communications. In each iteration, one device needs to communicate twice: once after solving the local problem and once with some probability p. We assume that the uncompressed message is 1. During the mandatory communication round, a device sends the compressed message $\phi \in (0, 1]$ instead of the full message. In the communication round that occurs with some probability, a device should send the full uncompressed message. Therefore, one device in average sends $O(1 \cdot \phi + p \cdot 1)$ amount of information. From this estimation we derive that the optimal choice of p is ϕ. Then,

$$K = O\left(\left[\frac{1}{\phi} + \frac{\omega\delta}{\sqrt{\phi}\mu} + \frac{L}{H\mu}\right]\log\left(\frac{\left\|z^0 - z^*\right\|^2}{\epsilon}\right)\right).$$

One can see that the optimal value of H is $\frac{\sqrt{\phi}L}{\delta\omega}$, which leads to

$$K = O\left(\left[\frac{1}{\phi} + \frac{\omega\delta}{\sqrt{\phi}\mu}\right]\log\left(\frac{\left\|z^0 - z^*\right\|^2}{\epsilon}\right)\right).$$

With the number of outer iterations, we can compute the communication complexity. Every device sends $O(\phi)$ data. Thus, the amount of information that needs to be sent by one device to achieve the precision of the solution ϵ is the following:

$$K \cdot O(\phi) = O\left(\left[1 + \frac{\sqrt{\phi}\omega\delta}{\mu}\right]\log\frac{1}{\epsilon}\right).$$

Finally, we would like to notice that our theoretical results are not better than those for unbiased Three Pillars algorithm [Beznosikov, Takác, Gasnikov, 2024] with any compressor, since their algorithm achieves lower bound. In the case of the majority of biased compressors, the convergence is worse [Beznosikov et al., 2023b]. Moreover, the algorithm takes more running time and memory per each iteration. However, as we will see in the next section, the gain is in practice.

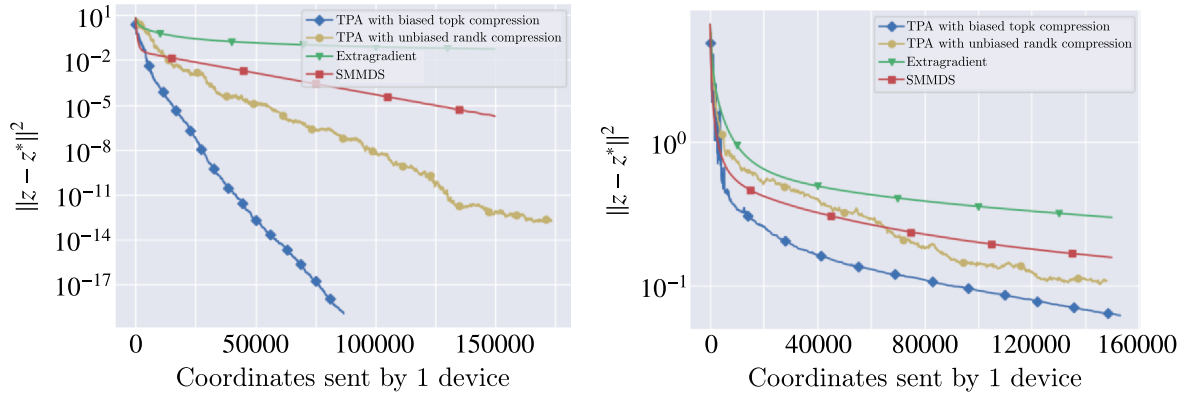
Experiments

Bilinear problem

We conduct experiments on the distributed bilinear saddle point problem (4) with the following local function on device m:

$$f_m(x, y) = x^\top M_m y + a_m^\top x + b_m^\top y + \frac{\lambda}{2}\|x\|^2 - \frac{\lambda}{2}\|y\|^2, \quad \forall m \in \overline{1, n},$$

where $M_m \in \mathbb{R}^{d_x \times d_y}$, $a_m \in \mathbb{R}^{d_x}$, $b_m \in \mathbb{R}^{d_y}$. This problem is λ-strongly monotone. We generate entries of servers M_1, and entries a_m, b_m from Standard Gaussian distribution. Devices' M_m, $m \in \overline{2, n}$, are generated by perturbing every entry from M_1 with the normal noise zero mean and controlled variance. Since data are synthetic, we can control the similarity of local functions. The dimensionality of the problem $d = 50$, the number of devices $n = 5$, regularization parameter $\lambda = 0.1$. As a compressor, we use Top K operator. The learning rate and K are tuned such that the best convergence is achieved. The solution z^* is found analytically.



(a) high similarity (b) low similarity

Figure 1. Experiments on bilinear problem. The x-axis denotes the number of coordinates transmitted by one of the devices. The y-axis denotes the logarithm of the squared distance to the solution

As competitors, we implement the Extra Gradient, SMMDS [Beznosikov et al., 2021c], and Three Pillars algorithm [Beznosikov, Takác, Gasnikov, 2024] with the unbiased Rand K compressor. The learning rate and parameter for compression are tuned for the best convergence. The results are presented in Fig. 1. The number of coordinates transmitted by any device is used as x-axis. The squared Euclidian distance to the solution is used as y-axis.

Our algorithm outperforms all competing algorithms. The margin between our algorithm and others becomes larger with increasing data similarity.

Robust linear regression

In this section, we conduct experiments on the following robust linear regression problem, that is, the instance of (5):

$$\min_{w \in \mathbb{R}^d} \max_{\|r_i\| \leqslant R} \frac{1}{2N} \sum_i^N \left(w^\top (a_i + r_i) - b_i\right)^2 + \frac{\lambda}{2}\|w\|^2 - \frac{\beta}{2}\|r\|^2.$$

This problem is λ-strongly convex and β-strongly concave. We test algorithms on the datasets `abalone` and `triazines` from the LibSVM library [Chang, Lin, 2011]. The training samples are

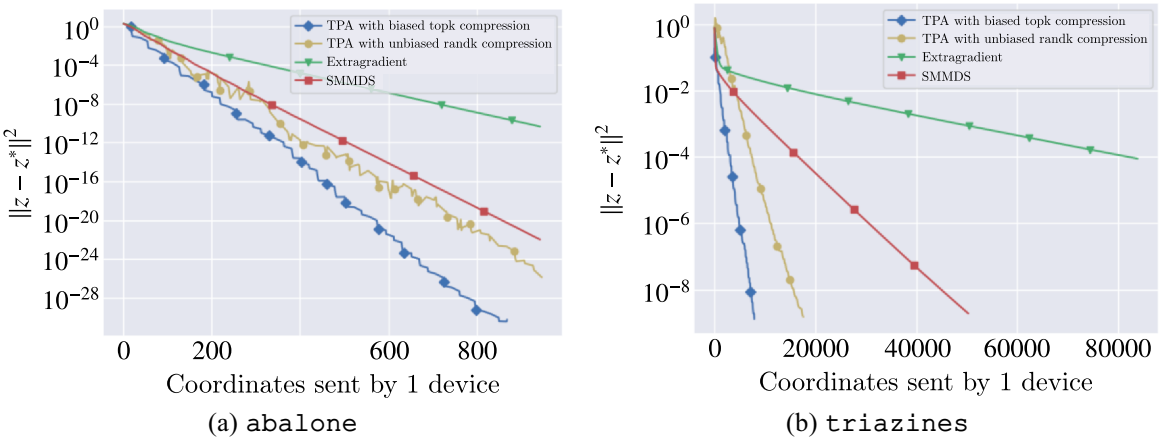


Figure 2. Experiment on robust linear regression. The x-axis denotes the number of coordinates transmitted by one of the devices. The y-axis denotes the logarithm of the squared distance to the solution

distributed equally among $n = 5$ devices. The solution x^* is found with a large number of Extra Gradient iterations. $R = 0.5$ and regularization parameters $\lambda = \beta = 0.1$. As previously, the compressor is the Top K operator; learning rates, number of inner iterations and K were tuned to achieve the best convergence. Also, the algorithms for comparison and their configurations are the same as in the previous experiment. The results can be seen in Fig. 2.

Again, our algorithm demonstrates superior performance in terms of communication complexity when compared to competitors. This supports the previous experiment.

Conclusion

We have created a new algorithm that solves distributed VI problems. The algorithm combines biased compression, local steps, and similarity. The algorithm outperforms existing methods in practice, but not in theory. For future works, we plan to derive lower bounds for the communication complexity with biased compression.

References

Alacaoglu A., Malitsky Y. Stochastic variance reduction for variational inequality methods // arXiv preprint. — 2021. — arXiv:2102.08352

Alistarh D., Grubic D., Li J., Tomioka R., Vojnovic M. QSGD: Communication efficient SGD via gradient quantization and encoding // Advances in Neural Information Processing Systems. — 2017. — P. 1709–1720.

Basu D., Data D., Karakus C., Diggavi S. Qsparse-local-sgd: Distributed sgd with quantization, sparsification and local computations // Advances in Neural Information Processing Systems. — 2019.

Beznosikov A., Dvurechensky P., Koloskova A., Samokhin V., Stich S. U., Gasnikov A. Decentralized local stochastic extra-gradient for variational inequalities // arXiv preprint. — 2021a. — arXiv:2106.08315

Beznosikov A., Dvurechensky P., Koloskova A., Samokhin V., Stich S. U., Gasnikov A. Decentralized local stochastic extra-gradient for variational inequalities // arXiv. — 2023a. — https://arxiv.org/abs/2106.08315

Beznosikov A., Gasnikov A. Compression and data similarity: Combination of two techniques for communication-efficient solving of distributed variational inequalities // International Conference on Optimization and Applications. — 2022. — P. 151–162.

Beznosikov A., Horváth S., Richtárik P., Safaryan M. On biased compression for distributed learning // Journal of Machine Learning Research. — 2023b. — Vol. 24, No. 276. — P. 1–50.

Beznosikov A., Richtárik P., Diskin M., Ryabinin M., Gasnikov A. Distributed methods with compressed communication for solving variational inequalities, with theoretical guarantees // arXiv preprint. — 2021b. — arXiv:2110.03313

Beznosikov A., Scutari G., Rogozin A., Gasnikov A. Distributed saddle-point problems under data similarity // Advances in Neural Information Processing Systems. — 2021c. — Vol. 34. — P. 8172–8184.

Beznosikov A., Takác M., Gasnikov A. Similarity, compression and local steps: three pillars of efficient communications for distributed variational inequalities // Advances in Neural Information Processing Systems. — 2024. — Vol. 36.

Chang C.-C., Lin C.-J. LIBSVM: A library for support vector machines // ACM transactions on intelligent systems and technology. — 2011.

Chavdarova T., Gidel G., Fleuret F., Lacoste-Julien S. Reducing noise in gan training with variance reduced extragradient // arXiv preprint. — 2019. — arXiv:1904.08598

Condat L., Agarsky I., Richtárik P. Provably doubly accelerated federated learning: The first theoretically successful combination of local training and compressed communication // arXiv preprint. — 2022. — arXiv:2210.13277

Daskalakis C., Ilyas A., Syrgkanis V., Zeng H. Training gans with optimism // arXiv preprint. — 2017. — arXiv:1711.00141

Deng Y., Mahdavi M. Local stochastic gradient descent ascent: Convergence analysis and communication efficiency // International Conference on Artificial Intelligence and Statistics. — 2021.

Facchinei F., Pang J.-S. Finite-dimensional variational inequalities and complementarity problems. — Springer Series in Operations Research, 2003.

Gidel G., Berard H., Vignoud G., Vincent P., Lacoste-Julien S. A variational inequality perspective on generative adversarial networks // arXiv preprint. — 2018. — arXiv:1802.10551

Goodfellow I. NIPS 2016 tutorial: generative adversarial networks // arXiv preprint. — 2016. — arXiv:1701.00160

Goodfellow I., Pouget-Abadie J., Mirza M., Xu B., Warde-Farley D., Ozair S., Courville A., Bengio Y. Generative adversarial net // Advances in Neural Information Processing Systems. — 2014.

Goodfellow I., Shlens J., Szegedy C. Explaining and harnessing adversarial examples // arXiv preprint. — 2014. — arXiv:1412.6572

Gorbunov E., Burlachenko K., Li Z., Richtárik P. MARINA: faster non-convex distributed learning with compression // arXiv. — 2022. — https://arxiv.org/abs/2102.07845

Hendrikx H., Xiao L., Bubeck S., Bach F., Massoulie L. Statistically preconditioned accelerated gradient method for distributed optimization // International conference on machine learning. — 2020. — P. 4203–4227.

Horváth S., Kovalev D., Mishchenko K., Stich S., Richtárik P. Stochastic distributed learning with gradient quantization and variance reduction // arXiv. — 2019. — https://arxiv.org/abs/1904.05115

Hou C., Thekumparampil K. K., Fanti G., Oh S. Efficient algorithms for federated saddle point optimization // arXiv preprint. — 2021. — arXiv:2102.06333

Jin Y., Sidford A. Efficiently solving MDPs with stochastic mirror descent // Proceedings of the 37th International Conference on Machine Learning. — 2020. — Vol. 119. — P. 4890–4900.

Kairouz P., McMahan H. B., Avent B., Bellet A., Bennis M., Bhagoji A. N., Bonawitz K., Charles Z., Cormode G., Cummings R. et al. Advances and open problems in federated learning // Foundations and trends® in machine learning. — 2021. — Vol. 14, No. 1–2. — P. 1–210.

Karimireddy S. P., Kale S., Mohri M., Reddi S., Stich S., Suresh A. T. SCAFFOLD: stochastic controlled averaging for federated learning // Proceedings of the 37th International Conference on Machine Learning. — 2020.

Karimireddy S. P., Rebjock Q., Stich S., Jaggi M. Error feedback fixes signsgd and other gradient compression schemes // International Conference on Machine Learning. — 2019. — P. 3252–3261.

Khaled A., Mishchenko K., Richtárik P. Tighter theory for local SGD on identical and heterogeneous data // International Conference on Artificial Intelligence and Statistics. — 2020. — P. 4519–4529.

Konečnỳ J. Federated learning: strategies for improving communication efficiency // arXiv preprint. — 2016. — arXiv:1610.05492

Kovalev D., Beznosikov A., Borodich E., Gasnikov A., Scutari G. Optimal gradient sliding and its application to optimal distributed optimization under similarity // Advances in Neural Information Processing Systems. — 2022. — Vol. 35. — P. 33494–33507.

Li T., Sahu A. K., Zaheer M., Sanjabi M., Talwalkar A., Smith V. Federated optimization in heterogeneous networks // Proceedings of Machine learning and systems. — 2020. — Vol. 2. — P. 429–450.

Madry A., Makelov A., Schmidt L., Tsipras D., Vladu A. Towards deep learning models resistant to adversarial attacks // International Conference on Learning Representations. — 2018.

Malinovsky G., Yi K., Richtárik P. Variance reduced proxskip: Algorithm, theory and application to federated learning // arXiv preprint. — 2022. — arXiv:2207.04338

Mangasarian L. O. Parallel gradient distribution in unconstrained optimization // SIAM Journal on Control and Optimization. — 1995. — Vol. 33, No. 6. — P. 1916–1925.

Mishchenko K., Gorbunov E., Takáč M., Richtárik P. Distributed learning with compressed gradient differences // arXiv. — 2019. — https://arxiv.org/abs/1901.09269

Mishchenko K., Malinovsky G., Stich S., Richtarik P. ProxSkip: Yes! Local gradient steps provably lead to communication acceleration! Finally! // Proceedings of the 39th International Conference on Machine Learning. — 2022. — Vol. 162. — P. 15750–15769. — https://proceedings.mlr.press/v162/mishchenko22b.html

Nadiradze G., Sabour A., Davies P., Markov I., Li S., Alistarh D. Decentralized sgd with asynchronous, local and quantized updates // arXiv preprint. — 2019. — arXiv:1910.12308

Richtárik P., Sokolov I., Fatkhullin I. EF21: A new, simpler, theoretically better, and practically faster error feedback // Advances in Neural Information Processing Systems. — 2021. — Vol. 34. — P. 4384–4396.

Seide F., Fu H., Droppo J., Li G., Yu D. 1-bit stochastic gradient descent and its application to data-parallel distributed training of speech dnns // Annual Conference of the International Speech Communication Association (INTERSPEECH). — 2014.

Shalev-Shwartz S., Ben-David S. Understanding machine learning: from theory to algorithms. — Cambridge university press, 2014.

Shamir O., Srebro N., Zhang T. Communication-efficient distributed optimization using an approximate Newton-type method // International conference on machine learning. — 2014. — P. 1000–1008.

Stich S. U., Cordonnier J.-B., Jaggi M. Sparsified sgd with memory // arXiv preprint. — 2018. — arXiv:1809.07599

Stich S. U., Karimireddy S. P. The error-feedback framework: SGD with delayed gradients // Journal of Machine Learning Research. — 2020. — Vol. 21, No. 237. — P. 1–36.

Szlendak R., Tyurin A., Richtárik P. Permutation compressors for provably faster distributed nonconvex optimization // arXiv preprint. — 2021. — arXiv:2110.03300

Vogels T., Karimireddy S. P., Jaggi M. PowerSGD: practical low-rank gradient compression for distributed optimization // 33rd Conference on Neural Information Processing Systems. — 2019.

Von Neumann J., Morgenstern O. Theory of games and economic behavior. — Princeton University Press, 1944.

Woodworth B., Patel K. K., Srebro N. Minibatch vs local sgd for heterogeneous distributed learning // arXiv preprint. — 2020. — arXiv:2006.04735

Woodworth B. E., Bullins B., Shamir O., Srebro N. The min-max complexity of distributed stochastic convex optimization with intermittent communication // Conference on Learning Theory. — 2021. — P. 4386–4437.

Yuan X.-T., Li P. On convergence of distributed approximate newton methods: Globalization, sharper bounds and beyond // arXiv preprint. — 2019. — arXiv:1908.02246

Zinkevich M., Weimer M., Li L., Smola A. J. Parallelized stochastic gradient descent // Advances in Neural Information Processing Systems. — 2010. — Vol. 23. — P. 2595–2603.