**КОМПЬЮТЕРНЫЕ ИССЛЕДОВАНИЯ**
**И МОДЕЛИРОВАНИЕ 2022 Т. 14 № 2 С. 257–275**
DOI: 10.20537/2076-7633-2022-14-2-257-275

Ки&М

**МАТЕМАТИЧЕСКИЕ ОСНОВЫ И ЧИСЛЕННЫЕ МЕТОДЫ МОДЕЛИРОВАНИЯ**

УДК: 519.8

# Редукция дисперсии для минимаксных задач с небольшой размерностью одной из переменных

**Е. Л. Гладин**[1,2,3,a]**, Е. Д. Бородич**[2,b]

[1]Берлинский университет имени Гумбольдта,
Германия, 10117, г. Берлин, Унтер-ден-Линден, д. 6
[2]Московский физико-технический институт,
Россия, 141701, г. Долгопрудный, Институтский пер., д. 9
[3]Институт проблем передачи информации РАН,
Россия, 127051, г. Москва, Большой Каретный пер., д. 19, стр. 1

E-mail: [a] egor.gladin@student.hu-berlin.de, [b] borodich.ed@phystech.edu

*Получено 13.02.2022.*
*Принято к публикации 13.02.2022.*

Статья посвящена выпукло-вогнутым седловым задачам, в которых целевая функция является суммой большого числа слагаемых. Такие задачи привлекают значительное внимание математического сообщества в связи с множеством приложений в машинном обучении, включая adversarial learning, adversarial attacks и robust reinforcement learning, и это лишь некоторые из них. Отдельные функции в сумме обычно представляют собой ошибку, связанную с объектом из выборки. Кроме того, формулировка допускает (возможно, негладкий) композитный член. Такие слагаемые часто отражают регуляризацию в задачах машинного обучения. Предполагается, что размерность одной из групп переменных относительно мала (около сотни или меньше), а другой — велика. Такой случай возникает, например, при рассмотрении двойственной формулировки задачи минимизации с умеренным числом ограничений. Предлагаемый подход основан на использовании метода секущей плоскости Вайды для минимизации относительно внешнего блока переменных. Этот алгоритм оптимизации особенно эффективен, когда размерность задачи не очень велика. Неточный оракул для метода Вайды вычисляется через приближенное решение внутренней задачи максимизации, которая решается ускоренным алгоритмом с редукцией дисперсии Katyusha. Таким образом, мы используем структуру задачи для достижения быстрой сходимости. В исследовании получены отдельные оценки сложности для градиентов различных компонент относительно различных переменных. Предложенный подход накладывает слабые предположения о целевой функции. В частности, не требуется ни сильной выпуклости, ни гладкости относительно низкоразмерной группы переменных. Количество шагов предложенного алгоритма, а также арифметическая сложность каждого шага явно зависят от размерности внешней переменной, отсюда предположение, что она относительно мала.

Ключевые слова: седловые задачи, методы первого порядка, методы секущей плоскости, редукция дисперсии

Работа Е. Л. Гладина финансирована Deutsche Forschungsgemeinschaft (DFG, Немецкий исследовательский фонд) в рамках Excellence Strategy Германии — Берлинский математический исследовательский центр MATH+ (EXC-2046/1, project ID: 390685689). Работа Е. Бородич выполнена при поддержке Министерства науки и высшего образования Российской Федерации (госзадание), № 075-00337-20-03, проект № 0714-2020-0005.

COMPUTER RESEARCH AND MODELING
2022 VOL. 14 NO. 2 P. 257–275
DOI: 10.20537/2076-7633-2022-14-2-257-275

Ки&М

MATHEMATICAL MODELING AND NUMERICAL SIMULATION

UDC: 519.8

# Variance reduction for minimax problems with a small dimension of one of the variables

**E. L. Gladin[1,2,3,a], E. D. Borodich[2,b]**

[1]Humboldt University of Berlin,
6 Unter den Linden, Berlin, 10117, Germany
[2]Moscow Institute of Physics and Technology,
9 Institutskiy per., Dolgoprudny, 141701, Russia
[3]Institute for Information Transmission Problems RAS,
19/1 Bolshoy Karetny per., Moscow, 127051, Russia

E-mail: [a] egor.gladin@student.hu-berlin.de, [b] borodich.ed@phystech.edu

*Received 13.02.2022.*
*Accepted for publication 13.02.2022.*

The paper is devoted to convex-concave saddle point problems where the objective is a sum of a large number of functions. Such problems attract considerable attention of the mathematical community due to the variety of applications in machine learning, including adversarial learning, adversarial attacks and robust reinforcement learning, to name a few. The individual functions in the sum usually represent losses related to examples from a data set. Additionally, the formulation admits a possibly nonsmooth composite term. Such terms often reflect regularization in machine learning problems. We assume that the dimension of one of the variable groups is relatively small (about a hundred or less), and the other one is large. This case arises, for example, when one considers the dual formulation for a minimization problem with a moderate number of constraints. The proposed approach is based on using Vaidya's cutting plane method to minimize with respect to the outer block of variables. This optimization algorithm is especially effective when the dimension of the problem is not very large. An inexact oracle for Vaidya's method is calculated via an approximate solution of the inner maximization problem, which is solved by the accelerated variance reduced algorithm Katyusha. Thus, we leverage the structure of the problem to achieve fast convergence. Separate complexity bounds for gradients of different components with respect to different variables are obtained in the study. The proposed approach is imposing very mild assumptions about the objective. In particular, neither strong convexity nor smoothness is required with respect to the low-dimensional variable group. The number of steps of the proposed algorithm as well as the arithmetic complexity of each step explicitly depend on the dimensionality of the outer variable, hence the assumption that it is relatively small.

Keywords: saddle point problems, first-order methods, cutting-plane methods, variance reduction

Citation: *Computer Research and Modeling*, 2022, vol. 14, no. 2, pp. 257–275.

The research of E. Gladin is funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy — The Berlin Mathematics Research Center MATH+ (EXC-2046/1, project ID: 390685689). The research of E. Borodich is supported by the Ministry of Science and Higher Education of the Russian Federation (Goszadaniye) No. 075-00337-20-03, project No. 0714-2020-0005.

## Introduction

For several decades now, minimax problems of the form

$$\min_{x\in\mathcal{X}}\max_{y\in\mathcal{Y}} F(x, y), \quad \mathcal{X} \subseteq \mathbb{R}^{n_x}, \quad \mathcal{Y} \subseteq \mathbb{R}^{n_y}, \tag{1}$$

have attracted a lot of attention of researchers from a variety of branches of mathematics and other fields, including game theory, economics, statistics, imaging, etc. [Isaacs, 1999; Morgenstern, Neumann, 1953; Taskar, Lacoste-Julien, Jordan, 2005; Haber, Modersitzki, 2004]. In recent years, there has been a particularly large amount of research on this topic in the machine learning community in connection with such topics as adversarial learning [Goodfellow et al., 2014], adversarial attacks [Madry et al., 2017] and robust reinforcement learning [Pinto et al., 2017], to name a few.

Common assumptions about the problems are as follows: $\mathcal{X}$, $\mathcal{Y}$ are nonempty closed convex sets, and $F(x, y)$ is a convex-concave function, that is, $F(\cdot, y)$ is convex for any $y \in \mathcal{Y}$, and $F(x, \cdot)$ is concave for any $x \in \mathcal{X}$. This general formulation is well-studied. There exist algorithms that achieve an $\varepsilon$-solution in terms of the duality gap in $O(1/\varepsilon)$ iterations [Nemirovski, 2004]. Such algorithms are optimal in the sense that they match the lower complexity bound for smooth convex-concave saddle point problems [Ouyang, Xu, 2021].

In many cases, exploiting the structure of the particular problem can lead to better results. Below we describe three distinctive scenarios that often arise in applications. In the first of them, the objective is a sum of a large number of components, i. e.,

$$F(x, y) = \frac{1}{m}\sum_{i=1}^{m} F_i(x, y). \tag{2}$$

A lot of research has been conducted in this direction [Hien, Zhao, Haskell, 2017; Song, Wright, Diakonikolas, 2021; Tominin et al., 2021; Palaniappan, Bach, 2016]. Increased interest in this setup is often motivated by the fact that it is extremely common in machine learning, where $F$ can correspond to empirical risk, and $F_i$ may represent losses related to individual examples. Together with (or instead of) the assumption (2), many articles consider the case where $F$ has a composite form:

$$F(x, y) = g(x) + f(x, y) - h(y),$$

where $f$ is convex-concave; composite terms $g$ and $h$ are convex functions possessing some peculiar properties, e. g. nonsmooth, prox-friendly, and so on. These composite terms often reflect regularization in various machine learning models. The ability to access oracles of the terms separately leads to improved convergence rates [Alkousa et al., 2020; Gasnikov et al., 2021]. The third important case of the problem (1) is when the dimension of one of the variables ($x$ or $y$) is relatively small (e. g., about a hundred or less), and the other is large. Consider, for example, the high-dimensional convex minimization problem with a few dozens of functional constraints:

$$\min_{y\in\mathcal{Y}} h(y) \text{ s. t. } \xi_i(y) \leqslant 0,\ i = 1, \ldots, n_x.$$

The Lagrange dual of this problem has the form

$$\min_{x\in\mathbb{R}^{n_x}_{+}}\max_{y\in\mathcal{Y}} \Big\{F(x, y) := -x^\top \xi(y) - h(y)\Big\},$$

where $\xi(y)$ is a vector with elements $\xi_i(y)$, $i = 1, \ldots, n_x$. This dual problem falls exactly into the described category. For deterministic low-dimensional minimization problems, cutting plane (or center of gravity type) methods are arguably most efficient as they achieve a linear convergence rate while imposing very mild assumptions [Bubeck, 2015]. Prominent examples of such methods are the ellipsoid method [Polyak, 1987] and Vaidya's cutting plane method [Vaidya, 1989; Vaidya, 1996]. As it turns out, these methods can be successfully paired with optimal first-order methods (e. g., [Gasnikov et al., 2021; Gasnikov, Tyurin, 2019; Gasnikov, Nesterov, 2018]) and incorporated into schemes for solving minimax problems where one of the dimensions is relatively small [Gladin et al., 2020; Gladin et al., 2021]. This is possible due to the fact that ellipsoid and Vaidya's methods are insensitive to the noise in the subgradient. Thus, optimal first-order methods can be utilized for finding an approximate solution of the inner maximization problem, which enables computation of a (noisy) subgradient of a function $G(x) := \max\limits_{y\in\mathcal{Y}} F(x, y)$. This approximate subgradient, in turn, is used by a cutting plane method for the outer minimization problem $\min\limits_{x\in\mathcal{X}} G(x)$. A similar approach was later used in [Usmanova et al., 2021] for finding a projection onto convex smooth constraints via the dual formulation.

The present paper puts together the three cases described above. That is, we consider minimax problems with the objective of finite-sum type and additional composite terms. Moreover, we assume the dimensionality of one of the variables to be relatively small (up to a hundred). Below we introduce the notation used throughout the article, and necessary definitions. After that, we give a formal statement of the problem and a preview of the results.

### *Preliminaries*

Solving optimization problems with finite-sum type objectives often involves randomized algorithms. The resulting solution is therefore a random vector. In this sense, there are two common ways to define approximate solution of a problem

$$f_* = \min_{x\in\mathcal{X}} f(x). \tag{3}$$

**Definition 1.** Let $\varepsilon > 0$, $\sigma \in (0, 1)$. A random vector $\widetilde{x} \in \mathcal{X}$ is called a $(\varepsilon, \sigma)$-solution of the problem (3) if

$$\mathbb{P}\left(f(\widetilde{x}) - f_* > \varepsilon\right) \leqslant \sigma.$$

If $\sigma = 0$, then $\widetilde{x}$ is simply called an $\varepsilon$-solution.

**Definition 2.** Let $\varepsilon > 0$. A random vector $\widetilde{x} \in \mathcal{X}$ is said to be a stochastic $\varepsilon$-solution of the problem (3) if

$$\mathbb{E} f(\widetilde{x}) - f_* \leqslant \varepsilon.$$

When describing the complexity of finding an $(\varepsilon, \sigma)$-solution or a stochastic $\varepsilon$-solution, we will use notation $\widetilde{O}(\cdot)$ which means $O(\cdot)$ up to a small power of logarithmic in $\varepsilon^{-1}$ and $\sigma^{-1}$ factor. The next definition introduces the notion of an inexact subgradient.

**Definition 3.** A vector $v \in \mathbb{R}^n$ is called a $\delta$-subgradient of a convex function $f$ at $z \in \operatorname{dom} f$ (denoted $v \in \partial_\delta f(z)$) if

$$f(x) \geqslant f(z) + v^\top(x - z) - \delta \quad \forall x \in \operatorname{dom} f.$$

If $\delta = 0$, this we get the usual definition of subgradient $v \in \partial f(z)$.

We will denote the Euclidean norm by $\|\cdot\|$. Another useful object in our study is a proximal operator.

**Definition 4.** Given a function $\psi$, a proximal operator is defined as a mapping

$$\operatorname{prox}_\psi(x) := \operatorname*{argmin}_{u\in\mathbb{R}^n}\left\{\psi(u) + \frac{1}{2}\|u - x\|^2\right\}. \tag{4}$$

The following property characterizes the smoothness of a function.

**Definition 5.** The function $f$ is called $L$-smooth, if for each $x_1, x_2 \in \mathbb{R}^n$

$$\|\nabla f(x_1) - \nabla f(x_2)\| \leqslant L\|x_1 - x_2\|. \tag{5}$$

## *Formulation of the problem*

In this paper we consider a problem of the form

$$\min_{x \in \mathcal{X}} \max_{y \in \mathbb{R}^{n_y}} \left[ F(x, y) = g(x) + \frac{1}{m} \sum_{i=1}^{m} f_i(x, y) - h(y) \right] \tag{6}$$

under the following assumptions:

**Condition 1.**

1. $\mathcal{X} \subseteq \mathbb{R}^{n_x}$ *is a compact convex set with nonempty interior, dimension* $n_x$ *is relatively small* (*up to a hundred*).
2. $F$ *is convex and continuous in* $x$.
3. $f_i$ *are concave in* $y$.
4. $f(x, y) := \frac{1}{m} \sum\limits_{i=1}^{m} f_i(x, y)$ *is* $\mu$*-strongly concave in* $y$ *and satisfies for any* $x \in \mathcal{X}$, $y, y' \in \mathbb{R}^{n_y}$

$$\frac{1}{m} \sum_{i=1}^{m} \|\nabla_y f_i(x, y) - \nabla_y f_i(x, y')\|^2 \leqslant 2L_f \left( f(x, y) - f(x, y') - \langle \nabla_y f(x, y'), y - y' \rangle \right). \tag{7}$$

5. $h$ *is convex.*

Remark 1. Condition (7) is satisfied, for example, if all of the functions $f_i$ are $L_f$-smooth. Besides, note that condition (7) implies that $f$ is also $L_f$-smooth in $y$. Indeed,

$$\left\| \frac{1}{m} \sum_{i=1}^{m} \left( \nabla_y f_i(x, y) - \nabla_y f_i(x, y') \right) \right\|^2 \leqslant \frac{1}{m} \sum_{i=1}^{m} \|\nabla_y f_i(x, y) - \nabla_y f_i(x, y')\|^2.$$

All that remains is to use Theorem 2.1.5 from [Nesterov, 2018].

We will also assume that at least **one** of the following two assumptions about the function $h$ holds true:

**Condition 2.A.** *$h$ is proximal-friendly, i. e., proximal operator* (4) *for $h$ is easy to compute.*

**Condition 2.B.** *$h$ is $L_h$-smooth.*

In each of these two setups, we propose an approach to solve the problem (6) and derive its complexity, which is indicated in Table 1 along with the bounds from other studies on the subject.

Table 1. Comparison: number of gradient evaluations or proximal operator computations to find an $\varepsilon$-saddle point for the problem (6) with probability at least $1 - \sigma$ or in expectation. It is assumed that Condition 1 holds, $g$ is $\mu$-strongly convex, $g$ and $h$ are proximal-friendly or smooth

| $g$ | $h$ | **Paper** | **Complexity** | |
|---|---|---|---|---|
| **prox-friendly** | **prox-friendly** | [Palaniappan, Bach, 2016; Alacaoglu, Malitsky, 2021] | $\operatorname{prox} g\colon \widetilde{O}\left(m + \sqrt{m}\frac{L_f}{\mu}\right)$ | $\nabla_x f_i\colon \widetilde{O}\left(m + \sqrt{m}\frac{L_f}{\mu}\right)$ |
| | | | $\operatorname{prox} h\colon \widetilde{O}\left(m + \sqrt{m}\frac{L_f}{\mu}\right)$ | $\nabla_y f_i\colon \widetilde{O}\left(m + \sqrt{m}\frac{L_f}{\mu}\right)$ |
| | | [Tominin et al., 2021] | $\operatorname{prox} g\colon \widetilde{O}\left(m + m^{3/4}\sqrt{\frac{L_f}{\mu}} + \sqrt{m}\frac{L_f}{\mu}\right)$ | $\nabla_x f_i\colon \widetilde{O}\left(m + m^{3/4}\sqrt{\frac{L_f}{\mu}} + \sqrt{m}\frac{L_f}{\mu}\right)$ |
| | | | $\operatorname{prox} h\colon \widetilde{O}\left(m + m^{3/4}\sqrt{\frac{L_f}{\mu}} + \sqrt{m}\frac{L_f}{\mu}\right)$ | $\nabla_y f_i\colon \widetilde{O}\left(m + m^{3/4}\sqrt{\frac{L_f}{\mu}} + \sqrt{m}\frac{L_f}{\mu}\right)$ |
| | | **This paper** | $\nabla g \in \partial g\colon \widetilde{O}(n_x)$ | $\nabla_x f_i\colon \widetilde{O}(mn_x)$ |
| | | | $\operatorname{prox} h\colon \widetilde{O}\left(n_x\left(m + \sqrt{\frac{mL_f}{\mu}}\right)\right)$ | $\nabla_y f_i\colon \widetilde{O}\left(n_x\left(m + \sqrt{\frac{mL_f}{\mu}}\right)\right)$ |
| | **$L_h$-smooth** | [Tominin et al., 2021] | $\operatorname{prox} g\colon \widetilde{O}\left(\sqrt{\frac{L_f}{\mu}}\right)$ | $\nabla_x f_i\colon \widetilde{O}\left(\frac{mL_f}{\mu}\right)$ |
| | | | $\nabla h\colon \widetilde{O}\left(\frac{\sqrt{L_h L_f}}{\mu}\right)$ | $\nabla_y f_i\colon \widetilde{O}\left(\frac{mL_f}{\mu}\right)$ |
| | | **This paper** | $\nabla g \in \partial g\colon \widetilde{O}(n_x)$ | $\nabla_x f_i\colon \widetilde{O}(mn_x)$ |
| | | | $\nabla h\colon \widetilde{O}\left(n_x\sqrt{\frac{L_h}{\mu}}\right)$ | $\nabla_y f_i\colon \widetilde{O}\left(n_x\sqrt{\frac{mL_h}{\mu}}\right)$ |
| | | **This paper** | $\nabla g \in \partial g\colon \widetilde{O}(n_x)$ | $\nabla_x f_i\colon \widetilde{O}(mn_x)$ |
| | | | $\nabla h\colon \widetilde{O}\left(n_x\left(m + \sqrt{\frac{L_f}{m\mu}} + \sqrt{\frac{mL_h}{\mu}}\right)\right)$ | $\nabla_y f_i\colon \widetilde{O}\left(n_x\left(m + \sqrt{\frac{m(L_f+L_h)}{\mu}}\right)\right)$ |
| **$L_g$-smooth** | **prox-friendly** | [Tominin et al., 2021] | $\nabla g\colon \widetilde{O}\left(\frac{\sqrt{L_g L_f}}{\mu}\right)$ | $\nabla_x f_i\colon \widetilde{O}\left(\frac{mL_f}{\mu}\right)$ |
| | | | $\operatorname{prox} h\colon \widetilde{O}\left(\sqrt{\frac{L_f}{\mu}}\right)$ | $\nabla_y f_i\colon \widetilde{O}\left(\frac{mL_f}{\mu}\right)$ |
| | | **This paper** | $\nabla g\colon \widetilde{O}(n_x)$ | $\nabla_x f_i\colon \widetilde{O}(mn_x)$ |
| | | | $\operatorname{prox} h\colon \widetilde{O}\left(n_x\left(m + \sqrt{\frac{mL_f}{\mu}}\right)\right)$ | $\nabla_y f_i\colon \widetilde{O}\left(n_x\left(m + \sqrt{\frac{mL_f}{\mu}}\right)\right)$ |
| | **$L_h$-smooth** | [Nesterov, 2011] | $\nabla g\colon \widetilde{O}\left(\frac{L_g+L_f+L_h}{\mu}\right)$ | $\nabla_x f_i\colon \widetilde{O}\left(m\frac{L_g+L_f+L_h}{\mu}\right)$ |
| | | | $\nabla h\colon \widetilde{O}\left(\frac{L_g+L_f+L_h}{\mu}\right)$ | $\nabla_y f_i\colon \widetilde{O}\left(m\frac{L_g+L_f+L_h}{\mu}\right)$ |
| | | [Lin, Jin, Jordan, 2020] | $\nabla g\colon \widetilde{O}\left(\frac{\sqrt{(L_g+L_f)(L_f+L_h)}}{\mu}\right)$ | $\nabla_x f_i\colon \widetilde{O}\left(m\frac{\sqrt{(L_g+L_f)(L_f+L_h)}}{\mu}\right)$ |
| | | | $\nabla h\colon \widetilde{O}\left(\frac{\sqrt{(L_g+L_f)(L_f+L_h)}}{\mu}\right)$ | $\nabla_y f_i\colon \widetilde{O}\left(m\frac{\sqrt{(L_g+L_f)(L_f+L_h)}}{\mu}\right)$ |
| | | [Alkousa et al., 2020] | $\nabla g\colon \widetilde{O}\left(\sqrt{\frac{L_g}{\mu}}\right)$ | $\nabla_x f_i\colon \widetilde{O}\left(m\frac{L_f}{\mu}\right)$ |
| | | | $\nabla h\colon \widetilde{O}\left(\frac{L_f}{\mu}\sqrt{\frac{L_h}{\mu}}\right)$ | $\nabla_y f_i\colon \widetilde{O}\left(m\left(\frac{L_f}{\mu}\right)^{3/2}\right)$ |
| | | [Palaniappan, Bach, 2016; Alacaoglu, Malitsky, 2021] | $\nabla g\colon \widetilde{O}\left(\frac{L_g+L_f+L_h}{\mu}\right)$ | $\nabla_x f_i\colon \widetilde{O}\left(\frac{L_g+L_f+L_h}{\mu}\right)$ |
| | | | $\nabla h\colon \widetilde{O}\left(\frac{L_g+L_f+L_h}{\mu}\right)$ | $\nabla_y f_i\colon \widetilde{O}\left(\frac{L_g+L_f+L_h}{\mu}\right)$ |
| | | [Tominin et al., 2021] | $\nabla g\colon \widetilde{O}\left(\frac{\sqrt{L_g L_f}}{\mu}\right)$ | $\nabla_x f_i\colon \widetilde{O}\left(\frac{mL_f}{\mu}\right)$ |
| | | | $\nabla h\colon \widetilde{O}\left(\frac{\sqrt{L_h L_f}+L_f}{\mu}\right)$ | $\nabla_y f_i\colon \widetilde{O}\left(\frac{mL_f}{\mu}\right)$ |
| | | **This paper** | $\nabla g\colon \widetilde{O}(n_x)$ | $\nabla_x f_i\colon \widetilde{O}(mn_x)$ |
| | | | $\nabla h\colon \widetilde{O}\left(n_x\sqrt{\frac{L_h}{\mu}}\right)$ | $\nabla_y f_i\colon \widetilde{O}\left(n_x\sqrt{\frac{mL_h}{\mu}}\right)$ |
| | | **This paper** | $\nabla g\colon \widetilde{O}(n_x)$ | $\nabla_x f_i\colon \widetilde{O}(mn_x)$ |
| | | | $\nabla h\colon \widetilde{O}\left(n_x\left(m + \sqrt{\frac{L_f}{m\mu}} + \sqrt{\frac{mL_h}{\mu}}\right)\right)$ | $\nabla_y f_i\colon \widetilde{O}\left(n_x\left(m + \sqrt{\frac{m(L_f+L_h)}{\mu}}\right)\right)$ |

## Outline of the Approach

Problem (6) can be thought of as a composition of the inner maximization problem

$$G(x) := \max_{y \in \mathbb{R}^{n_y}} F(x, y), \tag{8}$$

and the outer minimization problem

$$\min_{x \in X} G(x). \tag{9}$$

The outer problem can be solved by some iterative method which utilizes objective's gradients (or subgradients) at its steps. These gradients can be computed *inexactly* based on the approximate solution of the inner problem. We will use a particular type of inexact subgradient called $\delta$-subgradient, see Definition 3. As a $\delta$-subgradient of $g$ at $x \in Q_x$, we can take the subgradient $\nabla_x F(x, y) \in \partial_x F(x, y)$, where $y$ is a $\widetilde{\varepsilon}$-solution of the inner problem (8) for the current $x$. The required accuracy $\widetilde{\varepsilon}$ is given by the following lemma.

**Lemma 1 (see [Polyak, 1987]).** *In the assumptions of problem* (6)*, if* $\widetilde{y} \in \mathbb{R}^{n_y}$ *satisfies* $G(x) - F(x, \widetilde{y}) \leqslant \delta$ *for some* $\delta > 0$*, then* $\partial_x F(x, \widetilde{y}) \subseteq \partial_\delta G(x)$.

Thus, we need to solve the inner problem (8) with accuracy $\delta$ to obtain the $\delta$-subgradient.

The general algorithm for solving the problem (6) is given below, and the subsequent sections showcase particular methods that can be used with this algorithm along with the respective complexity bounds.

**Algorithm 1.** General algorithm for the problem (6)

```
Require: Method $\mathcal{M}_1$ solving (9) using $\delta$-subgradients, its number of steps $N > 0$, method $\mathcal{M}_2$
    solving (8), its expected accuracy $\widetilde{\varepsilon}$, initial point $(x^0, y^0)$
1: for $k = 0, \ldots, N - 1$ do
2:     Solve (8) for a fixed $x = x^k$ by $\mathcal{M}_2$ with expected accuracy $\widetilde{\varepsilon}$ starting from $y^k$:
           $y^{k+1} := \mathcal{M}_2(x^k, y^k, \widetilde{\varepsilon})$
3:     Put $v^{k+1} := \nabla_x S(x^k, y^{k+1}) \in \partial_x S(x^k, y^{k+1})$
4:     Make one step of $\mathcal{M}_1$ from $x^k$ using approximate subgradient $v^{k+1}$:
           $x^{k+1} := \mathrm{step}(\mathcal{M}_1, x^k, v^{k+1})$
5: end for
Ensure: $x^N$.
```

The complexity of Algorithm 1 is given by the following proposition.

**Proposal 1.** *Let* $\varepsilon > 0$, $\sigma \in (0, 1)$. *If the method* $\mathcal{M}_1$ *for solving* (9) *finds an* $\varepsilon$*-solution in* $N_1(\varepsilon, \delta)$ *computations of* $\delta$*-subgradients*[1]*, and the method* $\mathcal{M}_2$ *for solving* (8) *finds* $\widetilde{\varepsilon}$*-stochastic solution in* $N_2^f(m, \widetilde{\varepsilon})$ *computations of* $\nabla_y f_i$ *and* $N_2^h(\widetilde{\varepsilon})$ *computations of* $\nabla h$ *or* $\mathrm{prox}_{\eta h}$*, then for*

[1] To assure accuracy $\varepsilon$ in a finite number of steps, a method $\mathcal{M}_1$ may require $\delta$ to be sufficiently small relative to $\varepsilon$ (e. g., $\delta < \varepsilon$). The proposition assumes this requirement to be fulfilled.

*a given $\delta > 0$, Algorithm 1 with $\widetilde{\varepsilon} := \frac{\delta\sigma}{N_1(\varepsilon, \delta)}$ achieves the $(\varepsilon, \sigma)$-solution of the problem (6) after*

$$\begin{aligned} N_1(\varepsilon, \delta) &\textit{ computations of } \nabla g \in \partial g,\\ m \cdot N_1(\varepsilon, \delta) &\textit{ computations of } \nabla_x f_i \in \partial_x f_i,\\ N_1(\varepsilon, \delta) \cdot N_2^f(m, \widetilde{\varepsilon}) &\textit{ computations of } \nabla_y f_i,\\ N_1(\varepsilon, \delta) \cdot N_2^h(\widetilde{\varepsilon}) &\textit{ computations of } \nabla h \textit{ or } \operatorname{prox}_{\eta h}. \end{aligned}$$

## The methods used

### *Vaidya's cutting plane method*

Vaidya proposed a cutting plane method from [Vaidya, 1989; Vaidya, 1996] for solving problems of the form

$$\min_{x \in \mathcal{X}} G(x), \tag{10}$$

where $\mathcal{X} \subseteq \mathbb{R}^n$ is a compact convex set with non-empty interior, and $G\colon \mathcal{X} \to \mathbb{R}$ is a continuous convex function.

We will now introduce the notation and describe the algorithm. Let $P(A, b)$ denote a bounded full-dimensional polytope of the form

$$P(A, b) = \{x \in \mathbb{R}^n : Ax \geqslant b\} \text{ where } A \in \mathbb{R}^{m \times n} \text{ and } b \in \mathbb{R}^m.$$

The logarithmic barrier for $P$ is defined as

$$L(x; A, b) := -\sum_{i=1}^{m} \ln\left(a_i^\top x - b_i\right),$$

where $a_i^\top$ is the $i^{th}$ row of $A$. The Hessian of $L(x)$ is given by

$$H(x; A, b) = \sum_{i=1}^{m} \frac{a_i a_i^\top}{\left(a_i^\top x - b_i\right)^2} \tag{11}$$

and is positive definite for all $x$ in $\operatorname{int} P$ (interior of $P$). The *volumetric barrier* for $P(A, b)$ is defined as

$$V(x; A, b) = \frac{1}{2} \ln(\det H(x; A, b)),$$

where $\det H(x; A, b)$ denotes the determinant of $H(x; A, b)$. Let also $\sigma_i(x; A, b)$ denote the values

$$\sigma_i(x; A, b) = \frac{a_i^\top (H(x; A, b))^{-1} a_i}{\left(a_i^\top x - b_i\right)^2}, \quad 1 \leqslant i \leqslant m. \tag{12}$$

The *volumetric center* of $P$ is defined as the point $x_c$ that minimizes $V(x; A, b)$ over $\operatorname{int} P$:

$$x_c := \operatorname*{argmin}_{x \in \operatorname{int} P(A, b)} V(x; A, b). \tag{13}$$

The volumetric barrier $V$ is a self-concordant function and can therefore be efficiently minimized with the Newton-type methods. For more details and theoretical analysis, refer to [Vaidya, 1996; Vaidya, 1989]. It has been proved that one can use a $\delta$-subgradient instead of the exact subgradient in Vaidya's method [Gladin et al., 2021]. Below is the version of the algorithm using $\delta$-subgradients (Algorithm 2). The method produces a sequence of pairs $(A_k, b_k) \in \mathbb{R}^{m_k \times n} \times \mathbb{R}^{m_k}$, such that the corresponding polytopes contain a solution of the problem (10). A simplex containing the set $\mathcal{X}$ is often taken as the initial polytope $(A_0, b_0)$.

**Algorithm 2.** Vaidya's method using $\delta$-subgradients for the problem (10)

```
Require: Number of steps $N > 0$, $\delta \geqslant 0$, pair $(A_0, b_0) \in \mathbb{R}^{m_0 \times n} \times \mathbb{R}^{m_0}$, defining the initial polytope,
    algorithm parameters $\eta \leqslant 10^{-4}$, $\gamma \leqslant 10^{-3} \cdot \eta$.
for $k = 0, \ldots, N-1$ do
    Find an approximate volumetric center, see (13).
    Compute $H_k^{-1} := \left(H(x_k; A_k, b_k)\right)^{-1}$ and $\left\{\sigma_i(x_k; A_k, b_k)\right\}_{i=1}^{m_k}$, see (11) and (12),
    $i_k := \operatorname*{argmin}_{1 \leqslant i \leqslant m_k} \sigma_i(x_k; A_k, b_k)$
    if $\sigma_{i_k}(x_k; A_k, b_k) < \gamma$ then
        Obtain $\left(A_{k+1}, b_{k+1}\right)$ by removing the $i_k$th row from $\left(A_k, b_k\right)$,
        $m_{k+1} := m_k - 1$.
    else
        $c_k \in -\partial_\delta G(x_k)$,
        Find $\beta_k \in \mathbb{R}$ such that $c_k^\top x_k \geqslant \beta_k$ from the equation
            $\dfrac{c_k^\top H_k^{-1} c_k}{(c_k^\top x_k - \beta_k)^2} = \dfrac{1}{2}\sqrt{\eta\gamma},$
        $A_{k+1} := \begin{pmatrix} A_k \\ c_k^\top \end{pmatrix}$, $b_{k+1} := \begin{pmatrix} b_k \\ \beta_k \end{pmatrix}$, $m_{k+1} = m_k + 1$.
    end if
end for
Ensure: $x_N = \operatorname*{argmin}_{x \in \{x_0, \ldots, x_{N-1}\}} G(x)$.
```

**Theorem 1 ([Gladin et al., 2021]).** *Let $\mathcal{B}_\rho$ and $\mathcal{B}_\mathcal{R}$ be some Euclidean balls of radii $\rho$ and $\mathcal{R}$, respectively, such that $\mathcal{B}_\rho \subseteq \mathcal{X} \subseteq \mathcal{B}_\mathcal{R}$, and let a number $B > 0$ be such that $|G(x) - G(x')| \leqslant B$ $\forall x, x' \in \mathcal{X}$. After $N \geqslant \frac{2n}{\gamma}\ln\left(\frac{n^{1,5}\mathcal{R}}{\gamma\rho}\right) + \frac{1}{\gamma}\ln\pi$ iterations Vaidya's method with δ-subgradient for the problem* (10) *returns a point $x^N$ such that*

$$G(x^N) - \min_{x \in \mathcal{X}} G(x) \leqslant \frac{Bn^{1,5}\mathcal{R}}{\gamma\rho}\exp\left(\frac{\ln\pi - \gamma N}{2n}\right) + \delta, \tag{14}$$

*where $\gamma > 0$ is the parameter of the algorithm.*

### *L-Katyusha*

In this section we consider the accelerated variance reduction algorithm L-Katyusha from [Hanzely, Kovalev, Richtarik, 2020]. This method solves the problem

$$\min_{x \in \mathbb{R}^d}\Bigg\{F(x) := \underbrace{\frac{1}{m}\sum_{i=1}^{m} f_i(x)}_{f(x)} + h(x)\Bigg\}, \tag{15}$$

where $f(x)$ is $L_f$-smooth and $\mu$-strongly convex, $h(x)$ is convex and proximal-friendly. At iteration $k$ of L-Katyusha, a set $S \subseteq \{1, 2, \ldots, m\}$ of indices is sampled from the distribution defined by $p_i :=$ $:= \mathbb{P}(i \in S)$, $i = 1, \ldots, m$. The method uses the following unbiased gradient estimate:

$$\widehat{g}^k = \frac{1}{m}\sum_{i \in S}\frac{1}{p_i}(\nabla f_i(x^k) - \nabla f_i(w^k)) + \nabla f(w^k).$$

The pseudocode of L-Katyusha is presented as Algorithm 3. We formulate the following assumption to present the convergence theorem for Algorithm 3.

**Condition 3.** *There exists $\mathcal{L} \geqslant 0$ such that for all k we have*

$$\mathbb{E}\left[\|\widehat{g}^k - \nabla f(x^k)\|^2\right] \leqslant 2\mathcal{L}D_f(w^k, x^k),$$

*where $D_f(x, x') := f(x) - f(x') - \nabla f(x')^\top (x - x')$ is Bregman divergence.*

**Algorithm 3.** L-Katyusha

```
Input: starting point $x^0 \in \mathbb{R}^d$, number of iterations $K$, parameters $0 < \theta_1, \theta_2 < 1$, $\eta, \beta, \gamma > 0$,
probability $\rho$, $f(x) = \frac{1}{m}\sum_{i=1}^{m} f_i(x)$, $h(x)$
Initialization: $y^0 = z^0 = w^0 = x^0$
for $k = 0, 1, 2, \ldots, K - 1$ do
    $x^k = \theta_1 z^k + \theta_2 w^k + (1 - \theta_1 - \theta_2)y^k$,
    Sample random $S \subseteq \{1, 2, \ldots, m\}$
    $\widehat{g}^k = \nabla f(w^k) + \frac{1}{m}\sum_{i\in S}\frac{1}{p_i}\left(\nabla f_i(x^k) - \nabla f_i(w^k)\right)$,
    $y^{k+1} = \mathrm{prox}_{\eta h}(x^k - \eta\widehat{g}^k)$
    $z^{k+1} = \beta z^k + (1 - \beta)x^k + \frac{\gamma}{\eta}(y^{k+1} - x^k)$
    $w^{k+1} = \begin{cases} w^k, \text{ with prob. } 1 - \rho, \\ y^k, \text{ with prob. } \rho \end{cases}$
end for
```

**Theorem 2 (Corollary 5.3 from [Hanzely, Kovalev, Richtarik, 2020]).** *The iteration complexity of Algorithm* 3 *to find a stochastic ε-solution of the problem* (15)*, while Assumption* 3 *holds, can be bounded by*

$$N := O\left(\left(\frac{1}{\rho} + \sqrt{\frac{L_f}{\mu}} + \sqrt{\frac{\mathcal{L}}{\rho\mu}}\right)\log\frac{1}{\varepsilon}\right). \tag{16}$$

In our study, we fix $|S| = 1$ and $\rho = \frac{1}{m}$. To link Assumption 3 with our setup, we will use the following observation.

**Proposal 2.** *It holds that*

$$\mathbb{E}\left[\|\widehat{g}^k - \nabla f(x^k)\|^2\right] \leqslant \frac{1}{m}\sum_{i=1}^{m}\frac{1}{mp_i}\|\nabla f_i(x^k) - \nabla f_i(w^k)\|^2.$$

Let us set $p_i := \frac{1}{m}$, then Proposal 2 together with the property (7) implies condition 3. That brings us to the following complexity estimate which we give in the form of a corollary.

**Corollar 1.** *Let f be μ-strongly convex and satisfy* (7)*, then a stochastic ε-solution of the problem* (15) *can be achieved after the expected number of computations of* $\nabla f_i(\cdot)$ *and* $\mathrm{prox}_{\eta h}$

$$N = O\left\{\left(m + \sqrt{\frac{mL_f}{\mu}}\right)\log\frac{1}{\varepsilon}\right\}.$$

### *Special case of L-Katyusha*

Consider again the problem (15) preserving the same assumptions about $f$ but with $h$ now being $L_h$-smooth (and possibly not prox-friendly). It appears that L-Katyusha can be efficiently used to solve this problem if we treat it as a finite sum minimization with $m + 1$ terms and choose hyperparameters in an appropriate way. We present this special case of L-Katyusha in the form of Algorithm 4.

**Algorithm 4.** Special case of L-Katyusha

```
Input: starting point $x^0 \in \mathbb{R}^d$, number of iterations $K$, parameters $0 < \theta_1, \theta_2 < 1$, $\eta, \beta, \gamma > 0$, probabilities $p, \rho$
Initialization: $y^0 = z^0 = w^0 = x^0$
for $k = 0, 1, 2, \ldots, K-1$ do
    $x^k = \theta_1 z^k + \theta_2 w^k + (1 - \theta_1 - \theta_2) y^k$,
    Generate $\xi^k = \begin{cases} 1, \text{ with probability } 1-p, \\ 0, \text{ with probability } p \end{cases}$
    if $\xi^k = 0$ then
        $g^k = \frac{1}{p}\left(\nabla h(x^k) - \nabla h(w^k)\right) + \nabla f(w^k) + \nabla h(w^k)$
    else
        Sample random $i \in \{1, 2, \ldots, m\}$
        $g^k = \frac{1}{1-p}\left(\nabla f_i(x^k) - \nabla f_i(w^k)\right) + \nabla f(w^k) + \nabla h(w^k)$,
    end if
    $y^{k+1} = x^k - \eta g^k$
    $z^{k+1} = \beta z^k + (1-\beta) x^k + \frac{\gamma}{\eta}(y^{k+1} - x^k)$
    $w^{k+1} = \begin{cases} w^k, \text{ with prob. } 1-\rho, \\ y^k, \text{ with prob. } \rho \end{cases}$
end for
```

Based on Theorem 2, we prove the following complexity bound for Algorithm 4.

**Theorem 3.** *Algorithm* 4 *finds an $\varepsilon$-stochastic solution of the problem* (15) *with $L_h$-smooth composite term $h$ after the total expected number of computations of $\nabla f_i(\cdot)$ bounded by*

$$O\left((1 - p + m\rho)\left(\frac{1}{\rho} + \sqrt{\frac{L_f + L_h}{\mu}} + \sqrt{\frac{\mathcal{L}}{\rho\mu}}\right)\log\frac{1}{\varepsilon}\right),$$

*and the total expected number of computations of $\nabla h(\cdot)$ bounded by*

$$O\left((p + \rho)\left(\frac{1}{\rho} + \sqrt{\frac{L_f + L_h}{\mu}} + \sqrt{\frac{\mathcal{L}}{\rho\mu}}\right)\log\frac{1}{\varepsilon}\right),$$

*where* $\mathcal{L} = \max\left\{\frac{L_f}{1-p}, \frac{L_h}{p}\right\}$.

## Solving Saddle-Point Problems

This section presents algorithms for solving the minimax problem (6) assuming that one of Conditions 2.A, 2.B holds. As was pointed out in the section “Outline of the Approach”, we view this problem as a composition of inner maximization and outer minimization problems, (8) and (9), respectively. We apply Algorithm 1 with Vaidya's cutting plane method (Algorithm 2) as the method $\mathcal{M}_1$. We will use Theorem 1 and put $\delta = \frac{\varepsilon}{2}$, obtaining

$$N_1\left(\varepsilon, \frac{\varepsilon}{2}\right) = O\left(n_x \log \frac{n_x}{\varepsilon}\right) \tag{17}$$

steps of Vaidya's method. The full gradient in $x$ is calculated at each step, resulting in

$$O\left(mn_x \log \frac{n_x}{\varepsilon}\right) \tag{18}$$

evaluations of $\partial_x f_i$, while the number of computations of $\nabla g \in \partial g$ is given by (17).

Remark 2. As far as the arithmetic complexity of an iteration is concerned, Vaidya's cutting plane method involves inversions of $n_x \times n_x$ matrices, hence the assumption that $n_x$ is relatively small.

Further, for each of Conditions 2.A, 2.B we solve the inner problem in a specific way described in what follows.

### *Proximal point algorithm*

If the function $h(y)$ is proximal friendly, we propose to use L-Katyusha (Algorithm 3) as the method $\mathcal{M}_2$. According to Theorem 2, the number of oracle calls performed by L-Katyusha to ensure expected accuracy $\widetilde{\varepsilon}$ equals

$$N_2(m, \widetilde{\varepsilon}) = O\left(\left(m + \sqrt{\frac{mL_f}{\mu}}\right)\log\frac{1}{\widetilde{\varepsilon}}\right). \tag{19}$$

Now, estimates (18) and (19) together with Proposal 1 result in the following complexity bounds.

**Theorem 4.** *Assume that Conditions* 1 *and* 2.A *hold for the problem* (6)*, and let* $\varepsilon > 0$, $\sigma \in (0, 1)$. *Algorithm* 1 *with* $\delta := \varepsilon/2$, $\mathcal{M}_1$ := *Algorithm* 2, $\mathcal{M}_2$ := *Algorithm* 3 *arrives at* $(\varepsilon, \sigma)$*-solution of the problem* (6) *after*

$$\widetilde{O}(n_x) \text{ computations of } \nabla g \in \partial g,$$

$$\widetilde{O}(mn_x) \text{ computations of } \nabla_x f_i \in \partial_x f_i,$$

$$\widetilde{O}\left(n_x\left(m + \sqrt{\frac{mL_f}{\mu}}\right)\right) \text{ computations of } \nabla_y f_i,\ \mathrm{prox}_{\eta h}.$$

### *Smooth Algorithm*

If the function $h(y)$ is $L_h$-smooth, we propose to use the special case of L-Katyusha (Algorithm 4) as the method $\mathcal{M}_2$. According to Theorem 3, Algorithm 4 can ensure the expected accuracy $\widetilde{\varepsilon}$ after

$$O\left((1 - p + m\rho)\left(\frac{1}{\rho} + \sqrt{\frac{L_f + L_h}{\mu}} + \sqrt{\frac{\mathcal{L}}{\rho\mu}}\right)\log\frac{1}{\widetilde{\varepsilon}}\right)$$

computations of $\nabla_y f_i(\cdot)$ (in average) and

$$O\left((p + \rho)\left(\frac{1}{\rho} + \sqrt{\frac{L_f + L_h}{\mu}} + \sqrt{\frac{\mathcal{L}}{\rho\mu}}\right)\log\frac{1}{\widetilde{\varepsilon}}\right)$$

computations of $\nabla h(\cdot)$ (in average), where $\mathcal{L} = \max\left\{\frac{L_f}{1-p}, \frac{L_h}{p}\right\}$. Choose $p = \frac{L_h}{L_f + L_h}$ and get $\mathcal{L} = L_f + L_h$.

- If we choose $\rho = p$, we get

$$O\left((1-p+m\rho)\left(\frac{1}{\rho}+\sqrt{\frac{L_f+L_h}{\mu}}+\sqrt{\frac{\mathcal{L}}{\rho\mu}}\right)\log\frac{1}{\tilde{\varepsilon}}\right)=$$

$$=O\left(m\rho\left(\frac{1}{\rho}+\sqrt{\frac{L_f+L_h}{\mu}}+\sqrt{\frac{L_f+L_h}{\rho\mu}}\right)\log\frac{1}{\tilde{\varepsilon}}\right)=$$

$$=O\left(\left(m+m\rho\sqrt{\frac{L_f+L_h}{\mu}}+m\sqrt{\frac{\rho(L_f+L_h)}{\mu}}\right)\log\frac{1}{\tilde{\varepsilon}}\right)=$$

$$=O\left(\left(m+m\sqrt{\frac{L_h}{\mu}}\right)\log\frac{1}{\tilde{\varepsilon}}\right)=O\left(m\sqrt{\frac{L_h}{\mu}}\log\frac{1}{\tilde{\varepsilon}}\right)$$

computations of $\nabla_y f_i(\cdot)$ (in average) and

$$O\left((p+\rho)\left(\frac{1}{\rho}+\sqrt{\frac{L_f+L_h}{\mu}}+\sqrt{\frac{\mathcal{L}}{\rho\mu}}\right)\log\frac{1}{\tilde{\varepsilon}}\right)=$$

$$=O\left(\rho\sqrt{\frac{L_f+L_h}{\mu}}+\sqrt{\frac{\rho\mathcal{L}}{\mu}}\log\frac{1}{\tilde{\varepsilon}}\right)=O\left(\sqrt{\frac{L_h}{\mu}}\log\frac{1}{\tilde{\varepsilon}}\right)$$

computations of $\nabla h(\cdot)$ (in average).

- If we choose $\rho = \frac{1}{m}$, we get

$$O\left((1-p+m\rho)\left(\frac{1}{\rho}+\sqrt{\frac{L_f+L_h}{\mu}}+\sqrt{\frac{\mathcal{L}}{\rho\mu}}\right)\log\frac{1}{\tilde{\varepsilon}}\right)=O\left(\left(m+\sqrt{\frac{m(L_f+L_h)}{\mu}}\right)\log\frac{1}{\tilde{\varepsilon}}\right)$$

computations of $\nabla_y f_i(\cdot)$ (in average) and

$$O\left((p+\rho)\left(\frac{1}{\rho}+\sqrt{\frac{L_f+L_h}{\mu}}+\sqrt{\frac{\mathcal{L}}{\rho\mu}}\right)\log\frac{1}{\tilde{\varepsilon}}\right)=$$

$$=O\left(\left(m+\sqrt{\frac{L_h}{\mu}}+\sqrt{\frac{L_f+L_h}{m\mu}}+\sqrt{\frac{mL_h}{\mu}}\right)\log\frac{1}{\tilde{\varepsilon}}\right)=O\left(\left(m+\sqrt{\frac{L_f}{m\mu}}+\sqrt{\frac{mL_h}{\mu}}\right)\log\frac{1}{\tilde{\varepsilon}}\right)$$

computations of $\nabla h(\cdot)$ (in average).

**Theorem 5.** *Assume that Conditions* 1 *and* 2.B *hold for the problem* (6)*, and let* $\varepsilon > 0$, $\sigma \in (0, 1)$. *Algorithm* 1 *with* $\delta := \varepsilon/2$, $\mathcal{M}_1 :=$ *Algorithm* 2, $\mathcal{M}_2 :=$ *Algorithm* 4 *arrives at* $(\varepsilon,\sigma)$*-solution of the problem* (6) *after*

$$\widetilde{O}(n_x) \textit{ computations of } \nabla g \in \partial g,$$
$$\widetilde{O}(mn_x) \textit{ computations of } \nabla_x f_i \in \partial_x f_i,$$

*and the number of computations of* $\nabla_y f_i$, $\nabla h$ *which depends on the choice of* $\rho$ *in Algorithm* 4:

- *if* $\rho = p = \frac{L_h}{L_f + L_h}$, *then*

$$\widetilde{O}\left(n_x m \sqrt{\frac{L_h}{\mu}}\right) \textit{ computations of } \nabla_y f_i,$$

$$\widetilde{O}\left(n_x \sqrt{\frac{L_h}{\mu}}\right) \textit{ computations of } \nabla h;$$

- *if* $\rho = \frac{1}{m}$, *then*

$$\widetilde{O}\left(\left(m + \sqrt{\frac{m(L_f + L_h)}{\mu}}\right)\right) \textit{ computations of } \nabla_y f_i,$$

$$\widetilde{O}\left(\left(m + \sqrt{\frac{L_f}{m\mu}} + \sqrt{\frac{mL_h}{\mu}}\right)\right) \textit{ computations of } \nabla h.$$

## Conclusion

The present work provides a framework for solving convex-concave minimax problems of the form

$$\min_{x \in \mathcal{X}} \max_{y \in \mathbb{R}^{n_y}} \left[F(x, y) = g(x) + \frac{1}{m}\sum_{i=1}^{m} f_i(x, y) - h(y)\right], \tag{20}$$

where $\mathcal{X} \subseteq \mathbb{R}^{n_x}$ is a compact convex set with nonempty interior, and the dimension $n_x$ is relatively small (up to a hundred). The problem is treated as a composition of outer minimization and inner maximization problems. The proposed framework comes in the form of Algorithm 1 which can be used with various optimization methods $\mathcal{M}_1$ and $\mathcal{M}_2$ for solving outer and inner problems, respectively. The main requirement for the method $\mathcal{M}_1$ is to tolerate inaccuracy in the (sub)gradient. Cutting plane methods (e. g., the ellipsoid method and Vaidya's method [Vaidya, 1989; Vaidya, 1996]) are well suited for this role for two reasons. First, they provide linear convergence while working in very mild assumptions about the objective function. Second, some of these algorithms can be used with an inexact subgradient without accumulating the error, as proved in [Gladin et al., 2020] for the ellipsoid method and in [Gladin et al., 2021] for Vaidya's cutting plane method. The main drawback of these algorithms, however, is the dependence of the convergence rate on the dimension of the problem. Moreover, the arithmetic complexity of each step also grows fast with the dimension. Thus, we recommend applying them in the case where $n_x$ is relatively small.

Together with the framework for solving the problems of the form (20), we provide two examples of its use. In both of them, Vaidya's cutting plane method and L-Katyusha [Hanzely, Kovalev, Richtarik, 2020] are chosen as the methods $\mathcal{M}_1$ and $\mathcal{M}_2$, respectively. The first example considers the case of prox-friendly composite term $h(y)$, and in the second one it is assumed to be $L_h$-smooth. In each of these two scenarios, L-Katyusha has to be used in a special manner, which results in two sets of upper complexity bounds for the problem (20). These bounds are given in Theorems 4, 5 and Table 1. Compared to the known methods for this type of problems, the proposed approach has a number of advantages. First, strong convexity with respect to the $x$ variables is not required. Second, we do not assume the objective to be smooth in that variable group. Third, the complexity is proportional only to the square root of the conditional number, while most of other studies have a heavier dependence on this value. Our approach, however, has some limitations, and one of them is the inability to take advantage of the case where $g(x)$ is proximal-friendly. Another restriction is related to a relatively small dimensionality of $x$ caused by the fact that the number of iterations is proportional to $n_x$.

The aforementioned results have a theoretical importance on their own. However, the authors are in the process of conducting numerical experiments that will illustrate the performance of the proposed approach in practice. Another prospective task is to obtain lower complexity bounds for the considered type of minimax problems. Such bounds are already known for large-scale smooth strongly convex-strongly concave saddle point problems with objectives of sum type [Xie et al., 2020]. In our study, however, the dimension of one of the variables is assumed to be relatively small, which enables the use of cutting plane methods leading to a more favorable dependence on the conditional number of the problem.

## References

*Alacaoglu A., Malitsky Y.* Stochastic Variance Reduction for Variational Inequality Methods // arxiv.org. — 2021. — URL: https://arxiv.org/abs/2102.08352 (date of access: 12.02.2022).

*Alkousa M. S., Gasnikov A. V., Dvinskikh D. M., Kovalev D. A., Stonyakin F. S.* Accelerated methods for saddle-point problem // Computational Mathematics and Mathematical Physics. — 2020. — Vol. 60, No. 11 — P. 1787–1809.

*Bubeck S.* Convex optimization: Algorithms and complexity // Found. Trends Mach. Learn. — 2015. — Vol. 8, No. 3–4 — P. 231–357.

*Gasnikov A. V., Nesterov Yu. E.* Universal method for stochastic composite optimization problems // Computational Mathematics and Mathematical Physics. — 2018. — Vol. 58, No. 1 — P. 48–64.

*Gasnikov A. V., Dvinskikh D. M., Dvurechensky P. E., Kamzolov D. I., Matyukhin V. V., Pasechnyuk D. A., Tupitsa N. K., Chernov A. V.* Accelerated meta-algorithm for convex optimization problems // Computational Mathematics and Mathematical Physics. — 2021. — Vol. 61, No. 1 — P. 17–28.

*Gasnikov A. V., Tyurin A. I.* Fast gradient descent for convex minimization problems with an oracle producing a ($\delta$, $L$)-model of function at the requested point // Computational Mathematics and Mathematical Physics. — 2019. — Vol. 59, No. 7 — P. 1085–1097.

*Gladin E., Kuruzov I., Stonyakin F., Pasechnyuk D., Alkousa M., Gasnikov A.* Solving strongly convex-concave composite saddle point problems with a small dimension of one of the variables // arxiv.org. — 2020. — URL: https://arxiv.org/abs/2010.02280 (date of access: 12.02.2022).

*Gladin E., Sadiev A., Gasnikov A. V., Dvurechensky P. E., Beznosikov A., Alkousa M.* Solving smooth min-min and min-max problems by mixed oracle algorithms // Mathematical Optimization Theory and Operations Research: Recent Trends. 20th International Conference / eds. by A. Strekalovsky, Y. Kochetov, T. Gruzdeva, and A. Orlov. — Springer, 2021. — P. 19–40.

*Goodfellow I., Pouget-Abadie J., Mirza M., Xu B., Warde-Farley D., Ozair S., Courville A., Bengio Y.* Generative adversarial nets // Advances in Neural Information Processing Systems 27. Proceedings of the 2014 Conference / eds. by Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger. — The MIT Press, 2014. — Vol. 27.

*Haber E., Modersitzki J.* Numerical methods for volume preserving image registration // Inverse problems. — 2004. — Vol. 20, No. 5 — P. 1621.

*Hanzely F., Kovalev D., Richtarik P.* Variance reduced coordinate descent with acceleration: New method with a surprising application to finite-sum problems // Proceedings of the 37th International Conference on Machine Learning / eds. by H. Daumé III and A. Singh. — PMLR, 2020. — P. 4039–4048.

*Hien L. T. K., Zhao R., Haskell W. B.* An inexact primal-dual smoothing framework for large-scale non-bilinear saddle point problems // arxiv.org. — 2017. — URL: https://arxiv.org/abs/1711.03669 (date of access: 12.02.2022).

*Isaacs R.* Differential games: a mathematical theory with applications to warfare and pursuit, control and optimization. — Courier Corporation, 1999. — 384 p.

*Lin T., Jin C., Jordan M. I.* Near-optimal algorithms for minimax optimization // Proceedings of 33rd Conference on Learning Theory / eds. by J. Abernethy and S. Agarwal. — PMLR, 2020. — P. 2738–2779.

*Madry A., Makelov A., Schmidt L., Tsipras D., Vladu A.* Towards deep learning models resistant to adversarial attacks // arxiv.org. — 2017. — URL: https://arxiv.org/abs/1706.06083 (date of access: 12.02.2022).

*Morgenstern O., Neumann J. V.* Theory of games and economic behavior. — Princeton: Princeton university press, 1953. — 704 p.

*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. — 2004. — Vol. 15, No. 1 — P. 229–251.

*Nesterov Yu. E.* Solving strongly monotone variational and quasi-variational inequalities // Discrete & Continuous Dynamical Systems. — 2011. — Vol. 31, No. 4 — P. 1383.

*Nesterov Yu. E.* Lectures on convex optimization. — Berlin: Springer International Publishing, 2018. — 612 p.

*Ouyang Y., Xu Y.* Lower complexity bounds of first-order methods for convex-concave bilinear saddle-point problems // Mathematical Programming. — 2021. — Vol. 185, No. 1 — P. 1–35.

*Palaniappan B., Bach F.* Stochastic variance reduction methods for saddle-point problems // Advances in Neural Information Processing Systems 29. Annual Conference on Neural Information Processing Systems 2016 / eds. by D. D. Lee, U. von Luxburg, R. Garnett, M. Sugiyama, and I. Guyon. — The Curran Associates Inc., 2016. — P. 1416–1424.

*Pinto L., Davidson J., Sukthankar R., Gupta A.* Robust adversarial reinforcement learning // ICML’17: Proceedings of the 34th International Conference on Machine Learning / eds. by D. Precup and Y. W. Teh. — JMLR.org, 2017. — P. 2817–2826.

*Polyak B. T.* Introduction to optimization. — New York: Publications Division, Inc., 1987. — 464 p.

*Song C., Wright S. J., Diakonikolas J.* Variance reduction via primal-dual accelerated dual averaging for nonsmooth convex finite-sums // arxiv.org. — 2021. — URL: https://arxiv.org/abs/2102.13643 (date of access: 12.02.2022).

*Taskar B., Lacoste-Julien S., Jordan M.* Structured prediction via the extragradient method // Advances in Neural Information Processing Systems 18. Proceedings of the 2005 Conference / eds. by Y. Weiss, B. Schölkopf, and J. Platt. — The MIT Press, 2006. — P. 1345–1352.

*Tominin V., Tominin Y., Borodich E., Kovalev D., Gasnikov A., Dvurechensky P.* On accelerated methods for saddle-point problems with composite structure // arxiv.org. — 2021. — URL: https://arxiv.org/abs/2103.09344 (date of access: 12.02.2022).

*Usmanova I., Kamgarpour M., Krause A., Levy K.* Fast projection onto convex smooth constraints // Proceedings of the 38th International Conference on Machine Learning. Proceedings of the 2021 Conference / eds. by M. Meila and T. Zhang. — PMLR, 2021. — Vol. 139.

*Vaidya P. M.* A new algorithm for minimizing convex functions over convex sets // 30th Annual Symposium on Foundations of Computer Science. — 1989. — P. 338–343.

*Vaidya P. M.* A new algorithm for minimizing convex functions over convex sets // Mathematical programming. — 1996. — Vol. 73, No. 3 — P. 291–341.

*Xie G., Luo L., Lian Y., Zhang Z.* Lower complexity bounds for finite-sum convex-concave minimax optimization problems // Proceedings of the 37th International Conference on Machine Learning. Proceedings of the 2020 Conference / eds. by H. Daumé III and A. Singh. — PMLR, 2020. — Vol. 119.

## Proofs

### *Proof of Proposal 1*

Suppose that at each iteration of Algorithm 1, the probability to solve the inner problem (8) with accuracy worse than $\delta$ does not exceed $\frac{\sigma}{N_1(\varepsilon,\delta)}$. Then the probability to solve the inner problem with accuracy worse than $\delta$ at any of the $N_1(\varepsilon, \delta)$ iterations does not exceed $\sigma$. Let us calculate the complexity of finding the $\left(\delta, \frac{\sigma}{N_1(\varepsilon,\delta)}\right)$-solution of the inner problem by the method $\mathcal{M}_2$. For a random variable $X$, Markov's inequality yields

$$\mathbb{P}\,(X \geqslant \delta) \leqslant \frac{\mathbb{E}X}{\delta}.$$

Let $X$ be the accuracy of the output of $\mathcal{M}_2$, which is a random variable. The desired expected accuracy $\widetilde{\varepsilon}$ provided by the method $\mathcal{M}_2$ is then defined by

$$\frac{\mathbb{E}X}{\delta} = \frac{\sigma}{N_1(\varepsilon, \delta)} \iff \mathbb{E}X = \frac{\delta\sigma}{N_1(\varepsilon, \delta)} =: \widetilde{\varepsilon}.$$

Thus, the specified choice of $\widetilde{\varepsilon}$ ensures that with probability $1-\sigma$ the inner problem will be solved with accuracy $\delta$ at all the iterations of the outer loop, which means the availability of $\delta$-subgradients of $g$ for the method $\mathcal{M}_1$, see Lemma 1. We finish the proof by observing that computation of $\nabla_x F \in \partial_x F$ involves $m$ computations of $\nabla_x f_i \in \partial_x f_i$, and that the method $\mathcal{M}_2$ is executed at each iteration of Algorithm 1, hence the multiplication of its complexity by $N_1(\varepsilon, \delta)$ in the final estimates.

### *Proof of Proposal 2*

$$\mathbb{E}\left[\|\widetilde{g}^k - \nabla f(x^k)\|^2\right] = \sum_{i=1}^{m} p_i \left\|\frac{1}{m}\frac{1}{p_i}\left(\nabla f_i(x^k) - \nabla f_i(w^k)\right) - \left(\nabla f(x^k) - \nabla f(w^k)\right)\right\|^2 =$$

$$= \sum_{i=1}^{m} \frac{1}{m^2 p_i} \left\|\nabla f_i(x^k) - \nabla f_i(w^k)\right\|^2 - 2\sum_{i=1}^{m} \frac{1}{m}\left\langle \nabla f_i(x^k) - \nabla f_i(w^k), \nabla f(x^k) - \nabla f(w^k)\right\rangle +$$

$$+ \sum_{i=1}^{m} p_i \left\|\nabla f(x^k) - \nabla f(w^k)\right\|^2.$$

The last two terms equal

$$-2\left\langle \nabla f(x^k) - \nabla f(w^k), \nabla f(x^k) - \nabla f(w^k)\right\rangle + \left\|\nabla f(x^k) - \nabla f(w^k)\right\|^2 = -\left\|\nabla f(x^k) - \nabla f(w^k)\right\|^2,$$

and the statement follows.

### *Proof of Corollary 1*

Since $f$ satisfies (7), it is also $L_f$-smooth, see Remark 1. At each iteration of L-Katyusha, the expected number of computations of $\nabla f_i(\cdot)$ is $1 - \rho + \rho(m + 1) = 1 + \rho m$. Choose $|S| = 1$, $p_i = \frac{1}{m}$ and $\rho = \frac{1}{m}$. The statement follows from Theorem 2 and Proposition 2.

### *Proof of Theorem 3*

- First, we reduce the problem to a finite sum minimization with $m + 1$ terms:
$$F(x) = \frac{1}{m+1}\sum_{i=1}^{m}\frac{m+1}{m}f_i(x) + \frac{m+1}{m+1}h(x) = \frac{1}{m+1}\sum_{i=1}^{m}\widetilde{f}_i(x),$$
where
$$\widetilde{f}_i(x) = \begin{cases} \dfrac{m+1}{m}f_i(x), & i = 1, \ldots, m, \\ (m+1)h(x), & i = m+1. \end{cases} \tag{21}$$
Now we can apply L-Katyusha (Algorithm 3) to this problem. If $|S| = 1$, then the gradient estimate writes as
$$\widehat{g}^k = \frac{1}{(m+1)p_i}\left(\nabla\widetilde{f}_i(x^k) - \nabla\widetilde{f}_i(w^k)\right) + \nabla F(w^k).$$
Let $p_{m+1} := p \in (0, 1)$ and $p_i = \frac{1-p}{m}$, $i = 1, \ldots, m$, then (21) yields
$$\widehat{g}^k = \begin{cases} \dfrac{1}{1-p}\left(\nabla f_i(x^k) - \nabla f_i(w^k)\right) + \nabla F(w^k), & i = 1, \ldots, m, \\ \dfrac{1}{p}\left(\nabla h(x^k) - \nabla h(w^k)\right) + \nabla F(w^k), & i = m+1, \end{cases}$$
which is reflected in our special case of Katyusha (Algorithm 4).
- Next, let us define the constant $\mathcal{L}$ for Algorithm 4. Due to Proposal 2,
$$\mathbb{E}\left[\|\widehat{g}^k - \nabla F(x^k)\|^2\right] \leqslant \frac{1}{m+1}\sum_{i=1}^{m+1}\frac{1}{(m+1)p_i}\|\nabla\widetilde{f}_i(x^k) - \nabla\widetilde{f}_i(w^k)\|.$$
Using the definitions of $p_i$ and $\widetilde{f}_i$, we get
$$\mathbb{E}\left[\|\widehat{g}^k - \nabla F(x^k)\|^2\right] \leqslant \frac{1}{1-p}\sum_{i=1}^{m}\frac{1}{m}\left\|\nabla f_i(x^k) - \nabla f_i(w^k)\right\|^2 + \frac{1}{p}\left\|\nabla h(x^k) - \nabla h(w^k)\right\|^2.$$
Now, property (7) and $L_h$-smoothness of $h$ imply (see also Theorem 2.1.5 from [Nesterov, 2018])
$$\mathbb{E}\left[\|\widehat{g}^k - \nabla F(x^k)\|^2\right] \leqslant \frac{2L_f}{1-p}D_f(w^k, x^k) + \frac{2L_h}{p}D_h(w^k, x^k).$$
Choose $\mathcal{L} = \max\left\{\frac{L_f}{1-p}, \frac{L_h}{p}\right\}$, then
$$\mathbb{E}\left[\|\widehat{g}^k - \nabla F(x^k)\|^2\right] \leqslant 2\mathcal{L}D_F(w^k, x^k).$$
Assumption 3 holds.
- The expected number of computations of $\nabla f_i(\cdot)$ per iteration of Algorithm 4 is 0, if $\xi^k = 0$, $w^{k+1} = w^k$; 1, if $\xi^k = 1$, $w^{k+1} = w^k$; $m + 1$, if $\xi^k = 1$, $w^{k+1} = y^k$, and $m$, if $\xi^k = 0$, $w^{k+1} = y^k$. Then the total expected computation of $\nabla f_i(\cdot)$ is
$$O\left(((1-p)(1-\rho) + (m+1)(1-p)\rho + mp\rho)\cdot\left(\frac{1}{\rho} + \sqrt{\frac{L_f + L_h}{\mu}} + \sqrt{\frac{\mathcal{L}}{\rho\mu}}\right)\log\frac{1}{\varepsilon}\right) =$$
$$= O\left((1 - p + m\rho)\left(\frac{1}{\rho} + \sqrt{\frac{L_f + L_h}{\mu}} + \sqrt{\frac{\mathcal{L}}{\rho\mu}}\right)\log\frac{1}{\varepsilon}\right),$$

- The expected number of computations of $\nabla h(\cdot)$ in Algorithm 4 is the sum of complexity coming from the full gradient computation (if the statement includes $w^{k+1} = y^k$) and the rest (if the statement includes $\xi^k$). The former requires a computation of $\nabla h(\cdot)$, if $w^{k+1} = y^k$, the latter if $\xi^k$ is equal to 0. The total expected number of computations of $\nabla h(\cdot)$ is $O(\rho + p)$ per iteration. Thus, the total expected number of computations of $\nabla h(\cdot)$ is bounded by

$$O\left((p+\rho)\left(\frac{1}{\rho} + \sqrt{\frac{L_f + L_h}{\mu}} + \sqrt{\frac{\mathcal{L}}{\rho\mu}}\right)\log\frac{1}{\varepsilon}\right).$$