COMPUTER RESEARCH AND MODELING
2023 VOL. 15 NO. 2 P. 433–467
DOI: 10.20537/2076-7633-2023-15-2-433-467

КиМ

MATHEMATICAL MODELING AND NUMERICAL SIMULATION

UDC: 519.8

On Accelerated Methods for Saddle-Point Problems with Composite Structure

Y. D. Tominin[1,a], V. D. Tominin[1,b], E. D. Borodich[1,c], D. A. Kovalev[2,d], P. E. Dvurechensky[3,5,e], A. V. Gasnikov[1,f], S. V. Chukanov[4,g]

[1]Moscow Institute of Physics and Technology,
9 Institutskiy per., Dolgoprudny, Moscow region, 141701, Russia
[2]King Abdullah University of Science and Technology,
Thuwal, 23955 Thuwal, Makkah province, Saudi Arabia
[3]Weierstrass Institute for Applied Analysis and Stochastics,
39 Mohren st., Berlin, 10117, Germany
[4]Research Center "Computer Science and Control" of Russian Academy of Sciences,
44/2 Vavilova st., Moscow, 119333, Russia
[5]IITP RAS,
19/1 Bolshoy Karetny per., Moscow, 127051, Russia

E-mail: [a] tominin.yad@phystech.edu, [b] tominin.vd@phystech.edu, [c] borodich.ed@phystech.edu, [d] dakovalev1@gmail.com, [e] pavel.dvurechensky@wias-berlin.de, [f] gasnikov@yandex.ru, [g] chukanov47@mail.ru

Received 22.02.2023.
Accepted for publication 23.02.2023.

We consider strongly-convex-strongly-concave saddle-point problems with general non-bilinear objective and different condition numbers with respect to the primal and dual variables. First, we consider such problems with smooth composite terms, one of which has finite-sum structure. For this setting we propose a variance reduction algorithm with complexity estimates superior to the existing bounds in the literature. Second, we consider finite-sum saddle-point problems with composite terms and propose several algorithms depending on the properties of the composite terms. When the composite terms are smooth we obtain better complexity bounds than the ones in the literature, including the bounds of a recently proposed nearly-optimal algorithms which do not consider the composite structure of the problem. If the composite terms are prox-friendly, we propose a variance reduction algorithm that, on the one hand, is accelerated compared to existing variance reduction algorithms and, on the other hand, provides in the composite setting similar complexity bounds to the nearly-optimal algorithm which is designed for noncomposite setting. Besides, our algorithms allow one to separate the complexity bounds, i. e. estimate, for each part of the objective separately, the number of oracle calls that is sufficient to achieve a given accuracy. This is important since different parts can have different arithmetic complexity of the oracle, and it is desired to call expensive oracles less often than cheap oracles. The key thing to all these results is our general framework for saddle-point problems, which may be of independent interest. This framework, in turn is based on our proposed Accelerated Meta-Algorithm for composite optimization with probabilistic inexact oracles and probabilistic inexactness in the proximal mapping, which may be of independent interest as well.

Keywords: saddle-point problem, minimax optimization, composite optimization, accelerated algorithms

Citation: *Computer Research and Modeling*, 2023, vol. 15, no. 2, pp. 433–467.

The work in Sections 3–5 was funded by Russian Science Foundation (project 18-71-10108, https://rscf.ru/project/18-71-10108/). The work of E. Borodich in the rest part of the paper was supported by the Ministry of Science and Higher Education of the Russian Federation (Goszadaniye) 075-00337-20-03, project No. 0714-2020-0005.

КОМПЬЮТЕРНЫЕ ИССЛЕДОВАНИЯ
И МОДЕЛИРОВАНИЕ 2023 Т. 15 № 2 С. 433–467
DOI: 10.20537/2076-7633-2023-15-2-433-467

Ки&М

МАТЕМАТИЧЕСКИЕ ОСНОВЫ И ЧИСЛЕННЫЕ МЕТОДЫ МОДЕЛИРОВАНИЯ

УДК: 519.8

Об ускоренных методах для седловых задач с композитной структурой

Я. Д. Томинин[1,a], В. Д. Томинин[1,b], Е. Д. Бородич[1,c], Д. А. Ковалёв[2,d], П. Е. Двуреченский[3,5,e], А. В. Гасников[1,f], С. В. Чуканов[4,g]

[1]Московский физико-технический институт (национальный исследовательский университет),
Россия, 141701, Московская обл., г. Долгопрудный, Институтский пер., 9
[2]Университет науки и технологий короля Абдаллы,
Саудовская Аравия, 23955, провинция Мекка, г. Тувал
[3]Институт прикладного анализа и стохастики Вейерштрасса,
Германия, 10117, г. Берлин, ул. Морен, 39
[4]Федеральный исследовательский центр «Информатика и управление» Российской академии наук,
Россия, 119333, г. Москва, ул. Вавилова, 44/2
[5]Институт проблем передачи информации им. А. А. Харкевича,
Россия, 127051, г. Москва, Большой Каретный переулок, д. 19, стр. 1

E-mail: [a] tominin.yad@phystech.edu, [b] tominin.vd@phystech.edu, [c] borodich.ed@phystech.edu,
[d] dakovalev1@gmail.com, [e] pavel.dvurechensky@wias-berlin.de, [f] gasnikov@yandex.ru, [g] chukanov47@mail.ru

Получено 22.02.2023.
Принято к публикации 23.02.2023.

В данной работе рассматриваются сильно-выпукло сильно-вогнутые не билинейные седловые задачи с разными числами обусловленности по прямым и двойственным переменным. Во-первых, мы рассматриваем задачи с гладкими композитами, один из которых имеет структуру с конечной суммой. Для этой задачи мы предлагаем алгоритм уменьшения дисперсии с оценками сложности, превосходящими существующие ограничения в литературе. Во-вторых, мы рассматриваем седловые задачи конечной суммы с композитами и предлагаем несколько алгоритмов в зависимости от свойств составных членов. Когда составные члены являются гладкими, мы получаем лучшие оценки сложности, чем в литературе, включая оценки недавно предложенных почти оптимальных алгоритмов, которые не учитывают составную структуру задачи. Кроме того, наши алгоритмы позволяют разделить сложность, т. е. оценить для каждой функции в задаче количество вызовов оракула, достаточное для достижения заданной точности. Это важно, так как разные функции могут иметь разную арифметическую сложность оракула, а дорогие оракулы желательно вызывать реже, чем дешевые. Ключевым моментом во всех этих результатах является наша общая схема для седловых задач, которая может представлять самостоятельный интерес. Эта структура, в свою очередь, основана на предложенном нами ускоренном мета-алгоритме для композитной оптимизации с вероятностными неточными оракулами и вероятностной неточностью в проксимальном отображении, которые также могут представлять самостоятельный интерес.

Ключевые слова: седловая задача, минимаксная оптимизация, композитная оптимизация, ускоренные алгоритмы

Работа в разделах 3–5 выполнена при поддержке Российского научного фонда (проект 18-71-10108, https://rscf.ru/project/18-71-10108/). В остальных разделах работа Екатерины Бородич выполнена при поддержке Министерства науки и высшего образования Российской Федерации (госзадание) 075-00337-20-03, номер проекта 0714-2020-0005.

Introduction

Saddle-point optimization problems have many applications in different areas of modelling an optimization. The most classical example is, perhaps, two-player zero-sum games [Morgenstern, Von Neumann, 1953; Nash, John, 1950], including differential games [Isaacs, 1999]. More recent examples include imaging problems [Chambole, Pock, 2011] and machine learning problems [Shalev-Shwartz, Zhang, 2014], where primal-dual saddle-point representations of large-scale optimization problems are constructed and primal-dual methods are used. Many nonsmooth optimization problems, such as ℓ_∞ or ℓ_1 regression admit a saddle-point representation, which allows one to propose methods [Nesterov, 2005b; Nemirovski, 2004] having faster convergence than the standard subgradient scheme. Recently, saddle-point problems have started to attract more attention from the machine learning community in application to generative adversarial networks training, where the training process consists of a competition of a generator of nonreal images and a discriminator which tries to distinguish between real and artificial images. Other application examples are equilibrium problems in two-stage congested traffic flow models [Gasnikov, 2016].

From the algorithmic viewpoint the most studied setting deals with saddle-point problems having bilinear structure [Nesterov, 2005b; Nemirovski, 2004; Carmon et al., 2019; Song, Wright, Diakonikolas, 2021; Xie, Han, Zhang, 2021], where the cross term between the primal and dual variable is linear in each variable. The extensions include bilinear problems with prox-friendly (i. e. admitting a proximal operator in closed form) composite terms [Chambole, Pock, 2011; Lan, 2019]. A related line of research studies variational inequalities [Nemirovski, 2004; Lan, 2019] since any convex-concave saddle-point problem can be reformulated as a variational inequality problem with monotone operator. In this area lower bounds for first-order methods are known [Nemirovsky, Yudin, 1983] and optimal methods exist [Nemirovski, 2004; Nesterov, 2007; Nesterov, Scrimali, 2011; Chen, Lan, Ouyang, 2017; Lan, 2019]. Notably, these works do not rely on the bilinear structure and allow one to solve convex-concave saddle-point problems with Lipschitz-continuous gradients, including differential games [Dvurechensky, Nesterov, Spokoiny, 2015]. An alternative approach, which mostly inspired this paper, is based on representation of a saddle-point problem $\min\limits_x \max\limits_y G(x, y)$ as either a primal minimization problem with an implicitly given objective $g(x) = \max\limits_y G(x, y)$ or a dual maximization problem with an implicitly given objective $\widetilde{g}(y) = \min\limits_x G(x, y)$. This approach was used in [Nesterov, 2005b; Nesterov, 2005a] for problems with bilinear structure and later extended in [Hien, Zhao, Haskell, 2020] for general saddle-point problems. Such a connection with optimization turned out to be quite productive since it allows accelerated optimization methods to be exploited. In particular, recent advances in this direction are due to an observation [Gasnikov, Dvurechensky, Nesterov, 2016; Alkousa et al., 2020; Ibrahim et al., 2020] that primal and dual problems can have different condition numbers, which opens up a possibility to obtain faster algorithms.

In this paper we focus on strongly-convex-strongly-concave saddle-point problems with different condition numbers κ_x, κ_y of the primal and dual problems, respectively. The classical upper bound $\widetilde{O}(\kappa_x + \kappa_y)$ for this setting is achieved by the algorithm of [Nesterov, Scrimali, 2011]. Recently, [Ibrahim et al., 2020] proved a lower complexity bound $\widetilde{\Omega}\left(\sqrt{\kappa_x \kappa_y}\right)$ for first-order methods, which raised the question of whether first-order methods can be accelerated for this setting. Independently [Alkousa et al., 2019] proposed accelerated methods with improved, yet suboptimal complexity bounds. In [Lin, Jin, Jordan, 2020] the authors improved the bounds of [Alkousa et al., 2020] and proposed an algorithm with an optimal up to a polylogarithmic factor complexity bound $\widetilde{O}\left(\sqrt{\kappa_x \kappa_y}\right)$. Subsequently, the logarithmic factors have been improved independently in the papers (we cite them in chronological order) [Dvinskikh et al., 2020; Wang, Li, 2020; Yang et al., 2020]. The papers listed above consider large-scale regime when primal and dual problems have large dimension and use gradient-type

methods. If, say, the dimension of the primal variable x is moderate, one can use cutting-plane methods [Gladin et al., 2020; Gladin et al., 2021] in combination with gradient-type methods. We also mention the following papers which are related, but consider a different (from ours) setting of convex-concave saddle-point problems [Zhu, Liu, Tran-Dinh, 2020], strongly-convex-concave and nonconvex-concave [Thekumparampil et al., 2019], nonconvex-concave [Ostrovskii, Lowy, Razaviyayn, 2020; Xu et al., 2020].

When an optimization problem has a special structure of finite-sum, also known as empirical risk minimization problems, variance reduction [Lan, 2020; Lin, Jin, Jordan, 2020] techniques are often exploited to reduce the complexity bounds. We are interested also in application of such techniques for saddle-point problems. Variance reduction methods for saddle-point problems were proposed in [Palaniappan, Bach, 2016] and recently improved in [Alacaoglu, Malitsky, 2021], yet without distinguishing between primal and dual condition numbers.

In this paper we continue the line of research [Alkousa et al., 2020; Dvinskikh et al., 2020] by exploring additional structure of the problem, such as finite-sum form and presence of composite terms. We also develop algorithms which allow one to separate the complexity bounds for different parts of the problem. The latter, in particular, means that for each part of the objective we estimate separately the number of its gradient evaluations. This allows further acceleration to be obtained if the smoothness constants and complexities of an oracle call for different parts are different since more expensive oracles are called less frequently than it would be required by existing methods. Next, we consider two main problem formulations which have additional structure and which we explore in this paper. We also give a detailed explanation of the difference of our setting and bounds with the literature.

The first problem formulation we are interested in is the strongly-convex-strongly-concave saddle-point problem of the form

$$\min_{x\in\mathbb{R}^{d_x}} \max_{y\in\mathbb{R}^{d_y}} \{f(x) + G(x, y) - h(y)\}, \quad h(y) := \frac{1}{m_h}\sum_{i=1}^{m_h} h_i(y), \tag{1}$$

where $G(x, y)$ is convex in x and concave in y and is L_G-smooth in each variable, $f(x)$ is μ_x-strongly convex and L_f-smooth, $h(y)$ is μ_y-strongly convex and L_h-smooth. We refer to the functions f and h as composite terms. In this setting it is natural to define condition numbers $\kappa_x = \frac{L_G}{\mu_x}$ and $\kappa_y = \frac{L_G}{\mu_y}$ for the primal minimization and dual maximization problems, respectively. As already mentioned, the most studied [Chambole, Pock, 2011; Lan, 2019] setting corresponds to a particular case of $m_h = 1$ and bilinear function $G(x, y) = \langle Ax, y\rangle$ for some linear operator A and the functions f, g being prox-friendly, i. e. admit a tractable proximal operator [Moreau, 1965], e. g. evaluation of the point $\operatorname{argmin}_x \left\{f(x) + \frac{1}{2}\|x - \overline{x}\|_2^2\right\}$ in the case of f. Existing algorithms [Palaniappan, Bach, 2016; Alkousa et al., 2019; Alkousa et al., 2020; Lin, Jin, Jordan, 2020; Dvinskikh et al., 2020; Wang, Li, 2020; Yang et al., 2020] for problem (1) with non-bilinear structure do not exploit the finite-sum structure of the function h and when it is smooth require calculation of the gradient of the whole sum, which may be expensive when $m_h \gg 1$. Unlike them, we incorporate the variance reduction technique to make the number of evaluations of $\nabla h_i(y)$ smaller than by the existing methods. Unlike [Palaniappan, Bach, 2016; Lin, Jin, Jordan, 2020; Wang, Li, 2020; Yang et al., 2020], we separate the complexity estimates for each part of the objective, i. e. we estimate separately a sufficient number of evaluations of $\nabla f(x)$, $\nabla_x G(x, y)$, $\nabla_y G(x, y)$, $\nabla h_i(y)$ to achieve a given accuracy. This allows us to call each oracle less number of times than it is required by existing methods and is important since evaluation of each gradient can have different arithmetic operations complexity, and it is desired to call expensive oracles less often than cheap oracles. Compared to [Alkousa et al., 2019; Alkousa et al., 2020], where the complexities are also separated, we obtain better complexity bounds for each part of the objective. Moreover, for the

particular case when $f = h = 0$, our bounds are the same as the best known bounds [Wang, Li, 2020; Yang et al., 2020] and are optimal up to logarithmic factors. Otherwise, when $m_h > 1$ and/or f, h are nonzero we obtain the best, to our knowledge, complexity bounds. We summarize comparison of ours results and those reported elsewhere for the case $m_h > 1$ in Table 1 and for the particular case $m_h = 1$ in Table 2.

Table 1. Comparison of gradient complexities for problem (1) with $m_h > 1$, i. e. the number of corresponding gradient evaluations, to find an ε-saddle point with probability at least $1 - \sigma$. Notation $\widetilde{O}(X)$ hides constant factors polylogarithmic in ε^{-1} and σ^{-1}. For a function F, we denote $\kappa_x^{(F)} = L_F/\mu_x$, $\kappa_y^{(F)} = L_F/\mu_y$. The results of Theorem 6 are obtained under additional assumptions $m_h(4L_G + \mu_y) \leqslant L_h$, $2L_G + \mu_x \leqslant L_f, \mu_y \leqslant L_G$, $\mu_x \leqslant L_G$

References	**Complexity**		**Variance reduction**	**Complexity separation**
[Nesterov, Scrimali, 2011]	$\nabla f\colon \widetilde{O}\left(\kappa_x^{(f+G)} + \kappa_y^{(G+h)}\right)$	$\nabla_x G\colon \widetilde{O}\left(\kappa_x^{(f+G)} + \kappa_y^{(G+h)}\right)$	✗	✗
	$\nabla h_i\colon \widetilde{O}\left(m_h\kappa_x^{(f+G)} + m_h\kappa_y^{(G+h)}\right)$	$\nabla_y G\colon \widetilde{O}\left(\kappa_x^{(f+G)} + \kappa_y^{(G+h)}\right)$		
[Lin, Jin, Jordan, 2020; Wang, Li, 2020; Yang et al., 2020]	$\nabla f\colon \widetilde{O}\left(\sqrt{\kappa_x^{(f+G)}\kappa_y^{(G+h)}}\right)$	$\nabla_x G\colon \widetilde{O}\left(\sqrt{\kappa_x^{(f+G)}\kappa_y^{(G+h)}}\right)$	✗	✗
	$\nabla h_i\colon \widetilde{O}\left(m_h\sqrt{\kappa_x^{(f+G)}\kappa_y^{(G+h)}}\right)$	$\nabla_y G\colon \widetilde{O}\left(\sqrt{\kappa_x^{(f+G)}\kappa_y^{(G+h)}}\right)$		
[Alkousa et al., 2020]	$\nabla f\colon \widetilde{O}\left(\sqrt{\kappa_x^{(f)}}\right)$	$\nabla_x G\colon \widetilde{O}\left(\sqrt{\kappa_x^{(G)}\kappa_y^{(G)}}\right)$	✗	✓
	$\nabla h_i\colon \widetilde{O}\left(m_h\sqrt{\kappa_x^{(G)}\kappa_y^{(G)}\kappa_y^{(h)}}\right)$	$\nabla_y G\colon \widetilde{O}\left(\kappa_y^{(G)}\sqrt{\kappa_x^{(G)}}\right)$		
This paper (Theorem 6)	$\nabla f\colon \widetilde{O}\left(\sqrt{\kappa_x^{(f)}\kappa_y^{(G)}}\right)$	$\nabla_x G\colon \widetilde{O}\left(\sqrt{\kappa_x^{(G)}\kappa_y^{(G)}}\right)$	✓	✓
	$\nabla h_i\colon \widetilde{O}\left(\sqrt{m_h\kappa_x^{(G)}\kappa_y^{(h)}}\right)$	$\nabla_y G\colon \widetilde{O}\left(\sqrt{\kappa_x^{(G)}\kappa_y^{(G)}}\right)$		

The second problem formulation we are interested in is the strongly-convex-strongly-concave saddle-point problem of the form

$$\min_{x\in\mathbb{R}^{d_x}}\max_{y\in\mathbb{R}^{d_y}}\{f(x) + G(x, y) - h(y)\}, \quad G(x, y) := \frac{1}{m_G}\sum_{i=1}^{m_G} G_i(x, y), \tag{2}$$

where each $G_i(x, y)$ is convex in x and concave in y and is L_G^i-smooth in each variable, $f(x)$ is μ_x-strongly convex, and $h(y)$ is μ_y-strongly convex. In this setting it is natural to define condition numbers $\kappa_x = \frac{L_G}{\mu_x}$ and $\kappa_y = \frac{L_G}{\mu_y}$ for the primal minimization and dual maximization problems, respectively, where $L_G = \frac{1}{m_G}\sum\limits_{i=1}^{m_G} L_G^i$. We consider this problem under three different additional assumptions: a) $f(x)$ is L_f-smooth, $h(y)$ is L_h-smooth; b) $f(x)$ is L_f-smooth, $h(y)$ is smooth and prox-friendly; c) $f(x)$ and $h(y)$ are both prox-friendly. Under assumption a) and b), similarly to [Lin, Jin, Jordan, 2020; Wang, Li, 2020; Yang et al., 2020] we do not exploit the finite-sum structure of the function G. Yet, unlike these papers and [Palaniappan, Bach, 2016], where variance reduction methods are proposed, we separate the complexity bounds for the number of oracle calls for each part of the objective, i. e. we estimate a sufficient number of evaluations of $\nabla f(x)$, $\nabla_x G_i(x, y)$, $\nabla_y G_i(x, y)$, $\nabla h(y)$ to achieve a given accuracy. This allows us to call each oracle less number of times than it is required by existing methods and is important since evaluation of each gradient can have different arithmetic operations complexity, and it is desired to call expensive oracles less often than cheap

Table 2. Comparison of gradient complexities for problem (1) with $m_h = 1$, i. e. the number of corresponding gradient evaluations, to find an ε-saddle point for the problem. Notation $\widetilde{O}(X)$ hides constant factors polylogarithmic in ε^{-1}. **CS** stands for complexity separation. For a function F, we denote $\kappa_x^{(F)} = \frac{L_F}{\mu_x}$, $\kappa_y^{(F)} = \frac{L_F}{\mu_y}$

References	Complexity		Assumptions	CS
Lower bounds [Ibrahim et al., 2020]	$\nabla f\colon \widetilde{\Omega}\left(\sqrt{\kappa_x^{(f+G)}\kappa_y^{(G+h)}}\right)$	$\nabla_x G\colon \widetilde{\Omega}\left(\sqrt{\kappa_x^{(f+G)}\kappa_y^{(G+h)}}\right)$	f is L_f-smooth, h is L_h-smooth	✗
	$\nabla h_i\colon \widetilde{\Omega}\left(\sqrt{\kappa_x^{(f+G)}\kappa_y^{(G+h)}}\right)$	$\nabla_y G\colon \widetilde{\Omega}\left(\sqrt{\kappa_x^{(f+G)}\kappa_y^{(G+h)}}\right)$		
[Nesterov, Scrimali, 2011]	$\nabla f\colon \widetilde{O}\left(\kappa_x^{(f+G)} + \kappa_y^{(G+h)}\right)$	$\nabla_x G\colon \widetilde{O}\left(\kappa_x^{(f+G)} + \kappa_y^{(G+h)}\right)$	f is L_f-smooth, h is L_h-smooth	✗
	$\nabla h_i\colon \widetilde{O}\left(\kappa_x^{(f+G)}\kappa_y^{(G+h)}\right)$	$\nabla_y G\colon \widetilde{O}\left(\kappa_x^{(f+G)} + \kappa_y^{(G+h)}\right)$		
[Lin, Jin, Jordan, 2020; Wang, Li, 2020; Yang et al., 2020]	$\nabla f\colon \widetilde{O}\left(\sqrt{\kappa_x^{(f+G)}\kappa_y^{(G+h)}}\right)$	$\nabla_x G\colon \widetilde{O}\left(\sqrt{\kappa_x^{(f+G)}\kappa_y^{(G+h)}}\right)$	f is L_f-smooth, h is L_h-smooth	✗
	$\nabla h_i\colon \widetilde{O}\left(\sqrt{\kappa_x^{(f+G)}\kappa_y^{(G+h)}}\right)$	$\nabla_y G\colon \widetilde{O}\left(\sqrt{\kappa_x^{(f+G)}\kappa_y^{(G+h)}}\right)$		
[Alkousa et al., 2020]	$\nabla f\colon \widetilde{O}\left(\sqrt{\kappa_x^{(f)}}\right)$	$\nabla_x G\colon \widetilde{O}\left(\sqrt{\kappa_x^{(G)}\kappa_y^{(G)}}\right)$	f is L_f-smooth, h is L_h-smooth	✓
	$\nabla h_i\colon \widetilde{O}\left(\sqrt{\kappa_x^{(G)}\kappa_y^{(G)}\kappa_y^{(h)}}\right)$	$\nabla_y G\colon \widetilde{O}\left(\kappa_y^{(G)}\sqrt{\kappa_x^{(G)}}\right)$		
This paper (Corollary 4)	$\nabla f\colon \widetilde{O}\left(\sqrt{\kappa_x^{(f)}\kappa_y^{(G)}}\right)$	$\nabla_x G\colon \widetilde{O}\left(\sqrt{\kappa_x^{(G)}\kappa_y^{(G)}}\right)$	f is L_f-smooth, h is L_h-smooth	✓
	$\nabla h\colon \widetilde{O}\left(\max\left\{\sqrt{\kappa_x^{(G)}\kappa_y^{(h)}}, \sqrt{\kappa_x^{(G)}\kappa_y^{(G)}}\right\}\right)$	$\nabla_y G\colon \widetilde{O}\left(\sqrt{\kappa_x^{(G)}\kappa_y^{(G)}}\right)$		
This paper (Theorem 8)	$\nabla f\colon \widetilde{O}\left(\sqrt{\kappa_x^{(f)}\kappa_y^{(G)}}\right)$	$\nabla_x G\colon \widetilde{O}\left(\sqrt{\kappa_x^{(G)}\kappa_y^{(G)}}\right)$	f is L_f-smooth, h is L_h-smooth prox-friendly	✓
	$\nabla h\colon \widetilde{O}\left(\sqrt{\kappa_y^{(G)}}\right)$	$\nabla_y G\colon \widetilde{O}\left(\sqrt{\kappa_x^{(G)}\kappa_y^{(G)}}\right)$		

oracles. Compared to [Alkousa et al., 2019; Alkousa et al., 2020], where the complexities are also separated, we obtain better complexity bounds for each part of the objective. Moreover, for the particular case where $f = h = 0$, our bounds are the same as the best known bounds [Wang, Li, 2020; Yang et al., 2020].

We summarize the comparison of ours results and those reported elsewhere in Table 3.

Our approach

To solve the described saddle-point problems under different assumptions, we first propose a general framework and then specialize it to problem (1) or problem (2). Our approach to saddle-point problems is based on considering them as minimization problems with objective implicitly given as a solution to a maximization problem. Thus, to develop our general framework, we first consider an optimization problem of the form

$$\min_{x\in\mathbb{R}^d}\{F(x) := \varphi(x) + \psi(x)\}, \tag{3}$$

Table 3. Comparison of gradient complexities for problem (2), i. e. the number of corresponding gradient evaluations, to find an ε-saddle point with probability at least $1-\sigma$. Notation $\widetilde{O}(X)$ hides constant factors polylogarithmic in ε^{-1} and σ^{-1}. For a function F, we denote $\kappa_x^{(F)} = \frac{L_F}{\mu_x}$, $\kappa_y^{(F)} = \frac{L_F}{\mu_y}$. **Prox-f** (**Prox-h**) stands for f (h) being prox-friendly. **CS** stands for complexity separation. **VR** stands for variance reduction

Referenses	**Complexity**		**Prox-f**	**Prox-h**	**VR**	**CS**
[Nesterov, Scrimali, 2011]	$\nabla f\colon \widetilde{O}\left(\kappa_x^{(f+G)} + \kappa_y^{(G+h)}\right)$	$\nabla_x G_i\colon \widetilde{O}\left(m_G\kappa_x^{(f+G)} + m_G\kappa_y^{(G+h)}\right)$	✗	✗	✗	✗
	$\nabla h\colon \widetilde{O}\left(\kappa_x^{(f+G)} + m_h\kappa_y^{(G+h)}\right)$	$\nabla_y G_i\colon \widetilde{O}\left(m_G\kappa_x^{(f+G)} + m_G\kappa_y^{(G+h)}\right)$				
[Lin, Jin, Jordan, 2020]	$\nabla f\colon \widetilde{O}\left(\sqrt{\kappa_x^{(f+G)}\kappa_y^{(G+h)}}\right)$	$\nabla_x G_i\colon \widetilde{O}\left(m_G\sqrt{\kappa_x^{(f+G)}\kappa_y^{(G+h)}}\right)$	✗	✗	✗	✗
	$\nabla h\colon \widetilde{O}\left(\sqrt{\kappa_x^{(f+G)}\kappa_y^{(G+h)}}\right)$	$\nabla_y G_i\colon \widetilde{O}\left(m_G\sqrt{\kappa_x^{(f+G)}\kappa_y^{(G+h)}}\right)$				
[Alkousa et al., 2020]	$\nabla f\colon \widetilde{O}\left(\sqrt{\kappa_x^{(f)}}\right)$	$\nabla_x G_i\colon \widetilde{O}\left(m_G\sqrt{\kappa_x^{(G)}\kappa_y^{(G)}}\right)$	✗	✗	✗	✓
	$\nabla h\colon \widetilde{O}\left(\sqrt{\kappa_x^{(G)}\kappa_y^{(G)}\kappa_y^{(h)}}\right)$	$\nabla_y G_i\colon \widetilde{O}\left(m_G\kappa_y^{(G)}\sqrt{\kappa_x^{(G)}}\right)$				
[Palaniappan, Bach, 2016; Alacaoglu, Malitsky, 2021]	$\nabla f\colon \widetilde{O}\left(\sqrt{m_G}\frac{\max\{L_G+L_f, L_G+L_h\}}{\min\{\mu_x,\mu_y\}}\right)$	$\nabla_x G_i\colon \widetilde{O}\left(\sqrt{m_G}\frac{\max\{L_G+L_f, L_G+L_h\}}{\min\{\mu_x,\mu_y\}}\right)$	✗	✗	✓	✗
	$\nabla h\colon \widetilde{O}\left(\sqrt{m_G}\frac{\max\{L_G+L_f, L_G+L_h\}}{\min\{\mu_x,\mu_y\}}\right)$	$\nabla_y G_i\colon \widetilde{O}\left(\sqrt{m_G}\frac{\max\{L_G+L_f, L_G+L_h\}}{\min\{\mu_x,\mu_y\}}\right)$				
This paper (Theorem 7)	$\nabla f\colon \widetilde{O}\left(\sqrt{\kappa_x^{(f)}\kappa_y^{(G)}}\right)$	$\nabla_x G_i\colon \widetilde{O}\left(m_G\sqrt{\kappa_x^{(G)}\kappa_y^{(G)}}\right)$	✗	✗	✗	✓
	$\nabla h\colon \widetilde{O}\left(\max\left\{\sqrt{\kappa_x^{(G)}\kappa_y^{(h)}}, \sqrt{\kappa_x^{(G)}\kappa_y^{(G)}}\right\}\right)$	$\nabla_y G_i\colon \widetilde{O}\left(m_G\sqrt{\kappa_x^{(G)}\kappa_y^{(G)}}\right)$				
This paper (Theorem 8)	$\nabla f\colon \widetilde{O}\left(\sqrt{\kappa_x^{(f)}\kappa_y^{(G)}}\right)$	$\nabla_x G_i\colon \widetilde{O}\left(m_G\sqrt{\kappa_x^{(G)}\kappa_y^{(G)}}\right)$	✗	✓	✗	✓
	$\nabla h\colon \widetilde{O}\left(\sqrt{\kappa_y^{(G)}}\right)$	$\nabla_y G_i\colon \widetilde{O}\left(m_G\sqrt{\kappa_x^{(G)}\kappa_y^{(G)}}\right)$				
Lower bounds [Han, Xie, Zhang, 2021]	$\nabla f\colon \widetilde{\Omega}\left(\sqrt{m_G}\sqrt{\kappa_x^{(f+G)}\kappa_y^{(G+h)}}\right)$	$\nabla_x G_i\colon \widetilde{\Omega}\left(\sqrt{m_G}\sqrt{\kappa_x^{(f+G)}\kappa_y^{(G+h)}}\right)$	✗	✗	✓	✗
	$\nabla h\colon \widetilde{\Omega}\left(\sqrt{m_G}\sqrt{\kappa_x^{(f+G)}\kappa_y^{(G+h)}}\right)$	$\nabla_y G_i\colon \widetilde{\Omega}\left(\sqrt{m_G}\sqrt{\kappa_x^{(f+G)}\kappa_y^{(G+h)}}\right)$				

and develop a novel inexact accelerated gradient method (Algorithm 1) which uses inexact first-order information on φ and ψ and inexact proximal steps. Then we note that the problems (1) or (2) can be rewritten as

$$\min_{x\in\mathbb{R}^{d_x}}\{F(x) := f(x) + \underbrace{\max_{y\in\mathbb{R}^{d_y}}\{G(x,y) - h(y)\}}_{g(x)=G(x,y^*(x))-h(y^*(x))}\}, \tag{4}$$

which is consistent with the problem formulation (3). Using this representation, we can apply our Algorithm 1 with $\varphi(x) = f(x)$ and $\psi(x) = g(x)$ to solve this problem. In each step we need to obtain a first-order information about the function g, which we can do inexactly by solving the inner

maximization problem by the same Algorithm 1, but now with $\varphi(y) = -G(x, y)$ and $\psi(y) = h(y)$. To obtain near-optimal upper complexity bounds and separate oracle complexity for different parts of the problem (4), we introduce additional inner-outer cycles, which will be described in detail below.

As stated above, our framework is based on the system of inner-outer loops, where in each loop an accelerated gradient method is applied to obtain better complexity results. To implement our approach we then need a flexible accelerated method which can be applied in a number of different situations. In some sense we need an accelerated meta-algorithm, or an accelerated envelope, which uses any method at the lower level to solve an auxiliary problem of the upper level and, as a result, obtain an accelerated version of the method used at the lower level. Existing algorithms of this type [Lin, Mairal, Harchaoui, 2015; Monteiro, Svaiter, 2013; d'Aspremont, Scieur, Taylor, 2021] are based on the accelerated proximal point method that uses some algorithm at the lower level to implement inexact proximal mapping. Unfortunately, we cannot use these existing methods in our case since in our system of inner-outer loops a loop at the lower level leads to inexact gradient information at the upper level. Moreover, if a randomized method is used at the lower level, one obtains stochastic inexactness at the upper level. These kinds of inexactness of the oracles for φ, ψ are not accounted for in the existing general acceleration frameworks [Lin, Mairal, Harchaoui, 2015; Monteiro, Svaiter, 2013; d'Aspremont, Scieur, Taylor, 2021]. Motivated by this gap in the literature, we develop a generic accelerated meta-algorithm with probabilistic inexact oracles. Moreover, we also implement an adaptive stopping criterion for the method at the lower level which guarantees an appropriate quality of the inexact proximal mapping and leads to the accelerated convergence rate at the upper level.

Contributions

To sum up, our contributions are as follows. First, we provide a general inexact accelerated meta-algorithm (AM) listed as Algorithm 1 for convex optimization problems of the form (3) with inexact oracles. We also obtain an accelerated linearly convergent version of this algorithm by employing the restart technique with the resulting algorithm listed as Algorithm 2. We provide a theoretical analysis of this algorithm under stochastic inexactness in different parts of this algorithm, i. e. an inexact oracle and inexact proximal step. Unlike existing accelerated proximal methods, we consider composite problems (3) and use an inexact proximal step only with respect to φ. Next, we use this AM to construct a new general framework to systematically obtain new algorithms and complexity bounds for saddle-point problems with the structure (1) or (2). As a result, we obtain new accelerated methods for general saddle-point problems, including accelerated variance reduction methods, which leads to better complexity bounds than those existing in the literature. Moreover, our algorithms allow separation of complexity bounds for the number of oracle calls for each part of the problem formulation, i. e., for problem (1) we estimate a sufficient number of evaluations of $\nabla f(x)$, $\nabla_x G(x, y)$, $\nabla_y G(x, y)$, $\nabla h_i(y)$ to achieve a given accuracy. For problem (2) we estimate a sufficient number of evaluations of $\nabla f(x)$, $\nabla_x G_i(x, y)$, $\nabla_y G_i(x, y)$, $\nabla h(y)$ to achieve a given accuracy. This complexity separation is important since evaluation of each gradient can have different arithmetic operations complexity, and it is desired to call expensive oracles less often than cheap oracles.

Paper organization

In the section "Inexact Accelerated Meta-Algorithm", we propose an Accelerated Meta-Algorithm and extend it for a strongly convex setting with probabilistic inexact oracle and probabilistic inexactness in the proximal step. Then, in the section "Accelerated Framework for Saddle-Point Problems", by sequentially applying the Accelerated Meta-Algorithm, we obtain a general framework for solving saddle-point problems. This framework is based on two main assumptions for the possibility of solving two optimization problems. In the section "Accelerated Method for Saddle-Point Problems" we specialize the general framework to solve problem (1) by showing how to satisfy its two main

assumptions, and providing the resulting algorithm. Finally, in the section "Accelerated Methods for Saddle-Point Problems with Finite-Sum Structure" we consider problem (2) under additional assumptions: a) $f(x)$ is L_f-smooth, $h(y)$ is L_h-smooth; b) $f(x)$ is L_f-smooth, $h(y)$ is smooth and prox-friendly. We specialize the general framework for this setting and propose accelerated algorithms.

Notation and definitions

We introduce some notation and necessary definitions used throughout the paper. We denote by $\|x\|$ and $\|y\|$ the standard Euclidean norms for $x \in \mathbb{R}^{d_x}$ and $y \in \mathbb{R}^{d_y}$, respectively. This leads to the Euclidean norm on $\mathbb{R}^{d_x} \times \mathbb{R}^{d_y}$ defined as $\|(x_1, x_2) - (y_1, y_2)\|^2 = \|x_1 - x_2\|^2 + \|y_1 - y_2\|^2$, $x_1, x_2 \in \mathbb{R}^{d_x}$, $y_1, y_2 \in \mathbb{R}^{d_y}$.

We say that a function f is L_f-smooth if its gradient is Lipshitz-continuous, i. e.,

$$\|\nabla f(x_1) - \nabla f(x_2)\| \leqslant L_f \|x_1 - x_2\|, \quad x_1, x_2 \in \operatorname{dom} f \tag{5}$$

for some $L_f > 0$. We say that a function f is μ_f-strongly convex if, for some $\mu_f > 0$ and for any its subgradient, it holds that

$$f(x_2) \geqslant f(x_1) + \langle \nabla f(x_1), x_2 - x_1 \rangle + \frac{\mu_f}{2}\|x_1 - x_2\|^2, \quad x_1, x_2 \in \operatorname{dom} f. \tag{6}$$

We say that a pair $(\varphi_{\delta,L,\mu}(x), \nabla\varphi_{\delta,L,\mu}(x))$ is called (δ, L, μ)-oracle of a convex function φ at a point x if

$$\frac{\mu}{2}\|z - x\|^2 - \delta_1 \leqslant \varphi(z) - (\varphi_{\delta,L,\mu}(x) + \langle \nabla\varphi_{\delta,L,\mu}(x), z - x\rangle) \leqslant \frac{L}{2}\|z - x\|^2 + \delta_2 \quad \text{for all } z \in \mathbb{R}^{d_x}, \tag{7}$$

where $\delta = (\delta_1, \delta_2)$ and $\delta_1, \delta_2 > 0$. We use the notation (δ, L)-oracle if $(\delta_1, \delta_2) = (0, \delta)$.

We say that a function f is prox-friendly if it admits a tractable proximal operator [Moreau, 1965]. This means that the evaluation of the point

$$\operatorname{prox}_f^{\lambda}(\overline{x}) = \operatorname*{argmin}_{x \in \operatorname{dom} f} \left\{ \lambda f(x) + \frac{1}{2}\|x - \overline{x}\|^2 \right\} \tag{8}$$

for some fixed $\overline{x} \in \mathbb{R}^{d_x}$, $\lambda > 0$ can be made either in closed form or numerically very efficiently up to machine precision.

For an optimization problem $\min\limits_x f(x)$, we say that a random point $\widehat{x}$ is an (ε, σ)-solution to this problem for some $\varepsilon > 0$ and $\sigma \in (0, 1)$ if

$$f(\widehat{x}) - \min_x f(x) \leqslant \varepsilon \quad \text{with probability at least } 1 - \sigma. \tag{9}$$

We refer to ε as *accuracy* and to σ as *confidence level*.

We say that a function $G(x, y)$ is (strongly)-convex-(strongly)-concave if the function $G(\cdot, y)$ is (strongly)-convex for any fixed y and the function $G(x, \cdot)$ is (strongly)-concave for any fixed x. For a strongly-convex-strongly-concave saddle-point problem $\min\limits_x \max\limits_y G(x, y)$ a point $(\widehat{x}, \widehat{y})$ is called an (ε, σ)-solution for some $\varepsilon > 0$ and $\sigma \in (0, 1)$ if

$$\max_y G(\widehat{x}, y) - \min_x G(x, \widehat{y}) \leqslant \varepsilon \quad \text{with probability at least } 1 - \sigma. \tag{10}$$

Note that since the saddle-point problem is strongly-convex-strongly-concave, the quantity in the l.h.s. of (10) is correctly defined.

Notation $\widetilde{O}(\cdot)$ hides constant factors polylogarithmic in ε^{-1} and σ^{-1}. More precisely, $\psi_1(\varepsilon, \sigma) = \widetilde{O}(\psi_2(\varepsilon, \sigma))$ if there exist constants $C > 0$, a, b such that, for all $\varepsilon > 0$, $\sigma \in (0, 1)$, $\psi_1(\varepsilon, \sigma) \leqslant C\psi_2(\varepsilon, \sigma) \ln^a \frac{1}{\varepsilon} \ln^b \frac{1}{\sigma}$. We use $O(\cdot)$-notation when $a = b = 0$. For a function $\xi(\varepsilon)$ where $\varepsilon \in \mathbb{R}_+$ we write $\xi(\varepsilon) = \text{poly}(\varepsilon)$ if $\xi(\cdot) = \widetilde{O}(f(\varepsilon))$, where $f(\varepsilon)$ is a polynomial function of ε with nonnegative, possibly fractional powers. For a function $\xi(\varepsilon, \sigma)$, where $\varepsilon, \sigma \in \mathbb{R}_+$ we write $\xi(\varepsilon, \sigma) = \text{poly}(\varepsilon, \sigma)$ if $\xi(\cdot, \sigma)$ is a polynomial function of ε and $\xi(\varepsilon, \cdot)$ is a polynomial function of σ.

Inexact Accelerated Meta-algorithm

As described above, our approach is based on an accelerated composite optimization method. In this section we describe this method in an inexact oracle model to apply it to saddle-point problems. In this section we focus on the following optimization problem:

$$\min_{x \in \mathbb{R}^{d_x}} \{F(x) := \varphi(x) + \psi(x)\} \tag{11}$$

under the following assumption.

To motivate the study of this section, we slightly rewrite problem (1) in the following way:

$$\min_{x \in \mathbb{R}^{d_x}} \{F(x) := \varphi(x) + \underbrace{\max_{y \in \mathbb{R}^{d_y}} \{G(x, y) - h(y)\}}_{\psi(x) := G(x, y^*(x)) - h(y^*(x))}\}, \tag{12}$$

where $y^*(x)$ is the solution to the problem defining $\psi(x)$ for a fixed x. In other words, we can represent problem (1) as an optimization problem $\min\limits_{x \in \mathbb{R}^{d_x}} \varphi(x) + \psi(x)$ with a particular choice of φ, ψ:

$$\varphi = f(x), \quad \psi = \max_{y \in \mathbb{R}^{d_y}} \{G(x, y) - h(y)\}. \tag{13}$$

Importantly, we have no access to the exact gradients of $\psi(x)$ since we cannot solve exactly the problem defining $\psi(x)$. At the same time, according to Lemma 2 from [Alkousa et al., 2020], we can get (a precise definition is given below) an inexact $(\delta, 2L_\psi)$ oracle, where δ depends on the accuracy of the solution of the problem $\max\limits_{y \in \mathbb{R}^{d_y}} \{G(x, y) - h(y)\}$. Thus, we need to develop an accelerated algorithm for problem (12) which takes into account the inaccuracy of the oracles for functions $\varphi(x)$, $\psi(x)$ caused by the inexact solution to the optimization problem defining $\psi(x)$.

The situation is even more complicated if we consider problem (1) with $m_h > 1$ or problem (2) with $m_G > 1$ and apply variance reduction techniques. Application of known variance reduction methods guarantees us a solution to the problem $\max\limits_{y \in \mathbb{R}^{d_y}} \{G(x, y) - h(y)\}$ only with some high probability $1 - \sigma$. Thus, when using the variance reduction setting we obtain an inexact oracle for $\psi(x)$ only with some probability.

To sum up the motivation part, we need to develop a generic acceleration scheme which works with inexact oracles including inexact oracles with high probability. The rest of this section is devoted to the precise definitions of inexact oracles, description of such an accelerated algorithm and stating its convergence properties. Main technical proofs are deferred to the appendices. Since we believe that the proposed accelerated algorithm with inexact oracles can be of independent interest, we spend some effort to establish more results than we need for the main purpose of this paper. So, first we consider optimization with deterministic oracle, and then move to the setting of probabilistic inexact oracles.

Deterministic setting

Having in mind the above motivation, we introduce necessary notation and definitions. We start with a definition which corresponds to convex functions with Lipschitz-continuous gradient and is a small generalization of inexact oracle introduced in [Devolder, Glineur, Nesterov, 2014].

Definition 1. Let $\delta = (\delta_1, \delta_2)$, where $\delta_1, \delta_2 > 0$. Then the pair $(\varphi_{\delta,L}(x), \nabla\varphi_{\delta,L}(x))$ is called (δ, L)-oracle of a convex function $\varphi(x)$ at a point x, if

$$-\delta_1 \leqslant \varphi(z) - (\varphi_{\delta,L}(x) + \langle \nabla\varphi_{\delta,L}(x), z - x\rangle) \leqslant \frac{L}{2}\|z - x\|^2 + \delta_2 \quad \text{for all } z \in \mathbb{R}^{d_x}. \tag{14}$$

With a slight abuse of notation, we use the same notation (δ, L)-oracle for the case $(\delta_1, \delta_2) = (0, \delta)$.

Our Accelerated Meta-algorithm (AM) is listed below as Algorithm 1. The method generates three sequences, which are denoted by the same letter x with either no superscript or one of the two superscripts x^t, x^{md}. Since later we will use this algorithm in a system of inner-outer loops, we will change the letter to denote the sequences, but will not change the superscripts. The idea of the algorithm is inspired by the Monteiro – Svaiter algorithm [Monteiro, Svaiter, 2013], but there are several important differences. The first one is that in (15) we linearize the function φ instead of making an inexact proximal step for the whole objective F as it is done in [Monteiro, Svaiter, 2013]. The second difference is that we use inexact oracles for the functions φ and ψ, and as a corollary inexact oracle for F. This affects the measure of inexact solution to problem (15) and Step 7 of the algorithm. Thirdly, below we introduce a method more convenient in practice to control the accuracy of the solution to the inexact proximal step (15). To do that, we quantify with which accuracy one needs to solve the problem (15) in terms of its objective residual, so that the whole Algorithm 1 outputs a solution to the problem (11) with a desired accuracy. This makes it easy to apply Algorithm 1 in a system of inner-outer loops. Finally, the algorithm in [Monteiro, Svaiter, 2013] is not proved to obtain accelerated linear convergence rate in the case where the objective is strongly convex. For our algorithm we propose an extension which has an accelerated linear convergence rate under the additional assumption of inexact strong convexity.

The next theorem gives the convergence rate of Algorithm 1 when applied to the problem (11).

Theorem 1. *Assume that the starting point x_0 of Algorithm* 1 *satisfies $\|x_0 - x_*\| \leqslant R$ for some $R > 0$, and that the parameter H is chosen to satisfy $H \geqslant 2L_\varphi$. Assume also that the algorithm uses the (δ, L_φ)-oracle of convex function $\varphi(x)$ and (δ, L_ψ)-oracle of convex function $\psi(x)$, and that the auxiliary subproblem* (15) *is solved inexactly in each iteration in such a way that inequality* (16) *holds. Then, for all $k \geqslant 0$, the sequence x_k^t generated by Algorithm* 1 *satisfies*

$$F\left(x_k^t\right) - F(x_*) \leqslant \frac{4HR^2}{k^2} + 2\left(\sum_{i=1}^{k} A_i\right)\frac{\delta_2}{A_k} + \delta_1 + \left(\sum_{i=1}^{k-1} A_i\right)\frac{\delta_1}{A_k}. \tag{17}$$

We prove this theorem in the Appendix.

We now move further to the strongly-convex setting, which will allow us to solve strongly-convex-strongly-concave saddle-point problems in later sections. The next definition is an extension of Definition (1) and [Devolder, 2013] corresponding to strongly convex functions with Lipschitz-continuous gradient.

It is straightforward that a (δ, L, μ)-oracle is also a (δ, L)-oracle, and, thus, we can use (δ, L, μ)-oracle in Algorithm 1.

Next, we consider the case where $F(x)$ in (11) is convex and admits a (δ, L, μ)-oracle. Then, we use the convergence rate result in Theorem 1 and obtain a linear convergence rate by applying the restart technique. The restarted algorithm is listed as Algorithm 2, and its convergence rate when applied to the problem (11) is given in Theorem 2.

Algorithm 1. Accelerated Meta-algorithm (AM) with inexact (δ, L)-oracles

Input: objective $F = \varphi + \psi$ where φ, ψ are convex, parameter $H \geqslant 2L_\varphi$, inexactness $\delta \geqslant 0$, starting point x_0; $\left(\varphi_{\delta,L_\varphi}, \nabla\varphi_{\delta,L_\varphi}\right)$ — (δ, L_φ)-oracle of φ, $\left(\psi_{\delta,L_\psi}, \nabla\psi_{\delta,L_\psi}\right)$ — (δ, L_ψ)-oracle of ψ.
Set $A_0 = 0$, $x_0^t = x_0$, $x_0^{md} = x_0$.
for $k = 0$ **to** $k = K - 1$ **do**
  Set $a_{k+1} = \frac{1+\sqrt{1+8HA_k}}{4H}$, $A_{k+1} = A_k + a_{k+1}$.
  Set $x_k^{md} = \frac{A_k}{A_{k+1}} x_k^t + \frac{a_{k+1}}{A_{k+1}} x_k$.
  Find x_{k+1}^t as an approximate solution to the minimization problem

$$x_{k+1}^t \approx \operatorname*{argmin}_{z\in\mathbb{R}^{d_x}} \left\{\varphi_{\delta,L_\varphi}\left(x_k^{md}\right) + \left\langle \nabla\varphi_{\delta,L_\varphi}\left(x_k^{md}\right), z - x_k^{md}\right\rangle + \psi(x) + \frac{H}{2}\left\|z - x_k^{md}\right\|^2\right\}, \tag{15}$$

  such that

$$\left\|\nabla\varphi_{\delta,L_\varphi}\left(x_k^{md}\right) + \nabla\psi_{\delta,L_\psi}\left(x_{k+1}^t\right) + H\left(x_{k+1}^t - x_k^{md}\right)\right\| \leqslant \frac{H}{4}\left\|x_{k+1}^t - x_k^{md}\right\| - 2\sqrt{2\delta_2 L_\varphi}. \tag{16}$$

  $x_{k+1} = x_k - a_{k+1}\nabla\varphi_{\delta,L}\left(x_{k+1}^t\right) - a_{k+1}\nabla\psi_{\delta,L}\left(x_{k+1}^t\right)$.
end for
return x_K^t

Algorithm 2. Restarted AM (R-AM)

Input: objective $F = \varphi + \psi$ admits (δ, L, μ)-oracle, parameter $H \geqslant 2L_\varphi$, inexactness $\delta \geqslant 0$, starting point z_0; $\left(\varphi_{\delta,L_\varphi}, \nabla\varphi_{\delta,L_\varphi}\right)$ — (δ, L_φ)-oracle of convex function φ, $\left(\psi_{\delta,L_\psi}, \nabla\psi_{\delta,L_\psi}\right)$ — (δ, L_ψ)-oracle of convex function ψ.
for $k = 0$, **to** K **do**
  Set

$$N_k = \max\left\{\left\lceil\left(\frac{128H}{\mu}\right)^{1/2}\right\rceil, 1\right\}. \tag{18}$$

  Set $z_{k+1} := x_{N_k}^t$ as the output of Algorithm 1 started from z_k and run for N_k steps.
end for
return z_K

Theorem 2. *Assume that the starting point z_0 of Algorithm 2 satisfies $\|z_0 - x_*\| \leqslant R$ for some $R > 0$, and that the parameter H is chosen to satisfy $H \geqslant 2L_\varphi$. Further, assume that (δ, L, μ)-oracle of $F(x)$, (δ, L_φ)-oracle of convex function $\varphi(x)$, (δ, L_ψ)-oracle of convex function $\psi(x)$ are available, and, in each iteration of Algorithm 1 which is used as a building block of Algorithm 2, the auxiliary subproblem (15) is solved inexactly in such a way that inequality (16) holds. Finally, assume that the oracle inexactness δ_1, δ_2 is chosen to satisfy*

$$\forall k\colon \delta_1 + \delta_2 + 2\left(\sum_{i=1}^{k} A_i\right)\frac{\delta_2}{A_k} + \left(\sum_{i=1}^{k-1} A_i\right)\frac{\delta_1}{A_k} \leqslant \frac{\varepsilon}{2}, \tag{19}$$

$$\frac{4\sqrt{2\delta_2 L}}{\mu} \leqslant \frac{\varepsilon}{2}, \tag{20}$$

where ε is the desired accuracy of the solution to problem (11). *Then, under the listed assumptions, Algorithm* 2 *with* $K = 2\log_2 \frac{\mu R_0^2}{4\varepsilon}$ *guarantees that its output point* z_K *is an* ε*-solution to problem* (11), *i. e.* $F(z_K) - F(x^*) \leqslant \varepsilon$. *Moreover, the total number* N_F *of calls to inexact oracles both for* φ *and for* ψ *satisfies the following inequality*:

$$N_F \leqslant \left(16\sqrt{2}\sqrt{\frac{H}{\mu}} + 2\right)\log_2 \frac{\mu R_0^2}{\varepsilon} = \widetilde{O}\left(\max\left\{\sqrt{\frac{H}{\mu}}, 1\right\}\right). \tag{21}$$

We prove this theorem in the Appendix.

As we see from the above theorems, to ensure that AM and R-AM algorithms provide an ε-solution to problem (11), we need to guarantee that the oracle error $\delta = (\delta_1, \delta_2)$ is sufficiently small and that the auxiliary problem (15) is solved inexactly in such a way that inequality (16) is satisfied. For our purposes it is more convenient to consider the inexact solution of the problem (15) not in terms of inequality (16), but rather in terms of the objective residual in this problem bounded by some tolerance $\widetilde{\varepsilon}_f$. Next, we provide sufficient conditions on the values of δ and $\widetilde{\varepsilon}_f$ which guarantee that the conditions of the above theorems hold and that R-AM is guaranteed to find an ε-solution to problem (11).

Theorem 3. *Assume that the starting point* z_0 *of Algorithm* 2 *applied to problem* (11) *satisfies* $\|z_0 - x_*\| \leqslant R$ *for some* $R > 0$, *and that the parameter* H *is chosen to satisfy* $H \geqslant 2L_\varphi$. *Further, assume that* $F(x)$ *is convex,* (δ, L, μ)*-oracle of* $F(x)$, (δ, L_φ)*-oracle of convex function* $\varphi(x)$, (δ, L_ψ)*-oracle of convex function* $\psi(x)$ *are available, and, in each iteration of Algorithm* 1 *which is used as a building block of Algorithm* 2, *the auxiliary subproblem* (15) *is solved inexactly in such a way that the inexact solution* x_{k+1}^t *satisfies*

$$\left(\left\langle \nabla\varphi_{\delta,L_\varphi}\left(x_k^{md}\right), x_{k+1}^t - x_k^{md}\right\rangle + \psi\left(x_{k+1}^t\right) + \frac{H}{2}\left\|x_{k+1}^t - x_k^{md}\right\|^2\right) - \\ - \min_{z\in\mathbb{R}^{d_x}}\left(\left\langle \nabla\varphi_{\delta,L_\varphi}\left(x_k^{md}\right), z - x_k^{md}\right\rangle + \psi(x) + \frac{H}{2}\left\|z - x_k^{md}\right\|^2\right) \leqslant \widetilde{\varepsilon}_f, \tag{22}$$

where

$$\widetilde{\varepsilon}_f \leqslant \frac{\varepsilon\mu^2}{864^2(L+H)^2}. \tag{23}$$

Finally, assume that the oracle errors δ_1 *and* δ_2 *satisfy*

$$\delta_1, \delta_2 \leqslant \min\left\{\frac{\varepsilon\mu}{864^2 L_\varphi}, \frac{\varepsilon\mu}{864^2 L_\psi}, \frac{\varepsilon\mu^2}{864^2(L+H)^2}, \frac{\varepsilon^{3/2}}{5\sqrt{8HR^2}}\right\}. \tag{24}$$

Then, under the listed assumptions, Algorithm 2 *with* $K = 2\log_2 \frac{\mu R_0^2}{4\varepsilon}$ *guarantees that its output point* z_K *is an* ε*-solution to problem* (11), *i. e.* $F(z_K) - F(x^*) \leqslant \varepsilon$. *Moreover, the total number* N_F *of calls to inexact oracles both for* φ *and for* ψ *satisfies the following inequality*:

$$N_F \leqslant \left(16\sqrt{2}\sqrt{\frac{H}{\mu}} + 2\right)\log_2 \frac{\mu R_0^2}{\varepsilon} = \widetilde{O}\left(\max\left\{\sqrt{\frac{H}{\mu}}, 1\right\}\right). \tag{25}$$

We prove this theorem in the Appendix.

An important feature of the above bounds on δ_1, δ_2 and $\widetilde{\varepsilon}_f$ is that they depend polynomially on the target accuracy ε. This means that, if we can control these errors by some algorithms which

have complexity logarithmically depending on δ_1, δ_2 and $\widetilde{\varepsilon}_f$, then the total complexity of the whole algorithm R-AM will be logarithmic in the target accuracy ε, which makes it reasonable to apply this algorithm in a system of inner-outer loops. In the next subsection we extend the above theory for a stochastic setting.

Stochastic setting

As discussed at the beginning of this section, we would like to apply stochastic variance reduction methods or other randomized methods in order to provide an inexact solution to the auxiliary problem (15) and in order to obtain inexact oracle for F. In the former case inequality (22) can be guaranteed only with some probability. To illustrate the latter case, we consider function ψ in (13) with h given in (1) with $m_h \gg 1$, i. e.

$$\psi = \max_{y \in \mathbb{R}^{d_y}} \left\{ G(x, y) - \frac{1}{m_h} \sum_{i=1}^{m_h} h_i(y) \right\}. \tag{26}$$

According to Lemma 2 from [Alkousa et al., 2020], we can get an inexact $(\delta, 2L_\psi)$-oracle, where δ depends on the accuracy of the solution of this maximization problem. If we solve this maximization problem by a randomized method, we can obtain inexact $(\delta, 2L_\psi)$-oracle only with some probability. Thus, below we give a formal generalization of the results obtained in the previous subsection to a stochastic setting. We start with the definition of probabilistic inexact oracle.

Definition 2. Let $\delta = (\delta_1, \delta_2)$, where $\delta_1, \delta_2 > 0$. Then the pair $(\varphi_{\delta,L,\mu}(x), \nabla\varphi_{\delta,L,\mu}(x))$ is called $(\delta, \sigma_0, L, \mu)$-oracle of a convex function φ at a point x if

$$\frac{\mu}{2}\|z-x\|^2 - \delta_1 \leqslant \varphi(z) - \left(\varphi_{\delta,L,\mu}(x) + \left\langle \nabla\varphi_{\delta,L,\mu}(x), z - x \right\rangle\right) \leqslant \frac{L}{2}\|z-x\|^2 + \delta_2, \quad \text{for all } \in \mathbb{R}^{d_x} \text{ w.p. } 1-\sigma_0. \tag{27}$$

In the case of $\mu = 0$, we say that $(\varphi_{\delta,L}(x), \nabla\varphi_{\delta,L}(x))$ is called (δ, σ_0, L)-oracle of a function φ at a point x. With a slight abuse of notation, we use the same notation $(\delta, \sigma_0, L, \mu)$-oracle for the case $(\delta_1, \delta_2) = (0, \delta)$.

One should distinguish the following notation: the (δ, σ_0, L)-oracle of a function φ and (δ, L, μ)-oracle of a function φ.

The following is a simple lemma, which states that such a defined inexact oracle is additive.

Lemma 1. *Let the following assumptions hold.*

1. $\left(\varphi_{\delta_\varphi,L_\varphi,\mu_\varphi}(x), \nabla\varphi_{\delta_\varphi,L_\varphi,\mu_\varphi}(x)\right)$ *is* $(\delta_\varphi, \sigma_\varphi, L_\varphi, \mu_\varphi)$*-oracle for a convex function* φ*,*
2. $\left(\psi_{\delta_\psi,L_\psi,\mu_\psi}(x), \nabla\psi_{\delta_\psi,L_\psi,\mu_\psi}(x)\right)$ *is* $(\delta_\psi, \sigma_\psi, L_\psi, \mu_\psi)$*-oracle for a convex function* ψ*.*

Then $\left(\varphi_{\delta_\varphi,L_\varphi,\mu_\varphi}(x) + \psi_{\delta_\psi,L_\psi,\mu_\psi}(x), \nabla\varphi_{\delta_\varphi,L_\varphi,\mu_\varphi}(x) + \nabla\psi_{\delta_\psi,L_\psi,\mu_\psi}(x)\right)$ *is* $(\delta_\varphi + \delta_\psi, \sigma_\varphi + \sigma_\psi, L_\varphi + L_\psi, \mu_\varphi + \mu_\psi)$*-oracle for* $\varphi + \psi$*.*

We provide the proof of this lemma in the Appendix.

To illustrate why such an inexact oracle appears to be useful in the setting of saddle-point problems, we provide the following lemma, which extends the results of [Alkousa et al., 2019; Hien, Zhao, Haskell, 2020] to our stochastic setting and which will be very important for the derivations in the next section. This lemma contains some novelty in comparison with the literature: it is proved in the stochastic setting.

Lemma 2. *Let us consider the function*

$$g(x) = \max_{y \in \mathbb{R}^{d_y}} \{\widehat{S}(x, y) = F(x, y) - w(y)\}, \tag{28}$$

where $F(x, y)$ is convex in x, concave in y and is L_F-smooth as a function of (x, y), $w(y)$ is μ_y-strongly convex. Then $g(x)$ is L_g-smooth with $L_g = L_F + \frac{2L_F^2}{\mu_y}$ and $y^*(\cdot)$ is $\frac{2L_F}{\mu_y}$ Lipschitz continuous, where the point y^* is defined as

$$y^*(x) := \operatorname*{argmax}_{y \in \mathbb{R}^{d_y}} \widehat{S}(x, y) \quad \forall x \in \mathbb{R}^{d_x}. \tag{29}$$

Moreover, if a point $\widetilde{y}_{\delta/2}(x)$ is a $\left(\frac{\delta}{2}, \sigma\right)$-solution to (28)*, i. e. satisfies the inequality*

$$\max_{y \in \mathbb{R}^{d_y}} \{\widehat{S}(x, y)\} - \widehat{S}(x, \widetilde{y}_{\delta/2}(x)) \leqslant \frac{\delta}{2} \text{ w.p. } 1 - \sigma, \tag{30}$$

then $\nabla_x F(x, \widetilde{y}_{\delta/2}(x))$ is $(\delta, \sigma, 2L_g)$-oracle of g.

We prove this lemma in the Appendix.

Armed with Definition 2, we can now formulate the following theorem, which is a generalization of Theorem 3, and which is the main result of this section. This theorem provides the iteration complexity of Algorithm 2 to obtain an (ε, σ)-solution of problem (11) in the stochastic setting under the assumptions of probabilistic inexact oracles for φ, ψ in the sense of Definition 2 and also under the assumption that the auxiliary problem (15), which needs to be solved many times in each iteration of Algorithm 2, is solved inexactly with accuracy controlled in a probabilistic sense.

Theorem 4. *Consider the optimization problem* (11)

$$\min_{x \in \mathbb{R}^{d_x}} F(x) = \varphi(x) + \psi(x),$$

where $F(x)$ is convex. Let the target accuracy $\varepsilon > 0$ and the target confidence level $\sigma \in (0, 1)$ be given. Let also be given $H \geqslant 2L_\varphi$, a starting point z_0 and a number $R_0 > 0$ such that $\|z_0 - x_*\| \leqslant R_0$, where x_* is the solution to (11)*. Let the following two main assumptions of this theorem hold.*

1. (*Inexact oracle.*) *Inexact $(\delta, \sigma_0, L, \mu)$-oracle of $F(x)$, $(\delta, \sigma_0, L_\varphi)$-oracle of convex function $\varphi(x)$, $(\delta, \sigma_0, L_\psi)$-oracle of convex function $\psi(x)$ are available, where $\delta_1(\varepsilon)$, $\delta_2(\varepsilon)$ satisfy the following polynomial dependence on ε*:

$$\delta_1(\varepsilon), \delta_2(\varepsilon) \leqslant \min\left\{\frac{\varepsilon\mu}{864^2 L_\varphi}, \frac{\varepsilon\mu}{864^2 L_\psi}, \frac{\varepsilon\mu^2}{864^2 (L+H)^2}, \frac{\varepsilon^{3/2}}{5\sqrt{8HR_0^2}}\right\}, \tag{31}$$

and $\sigma_0(\varepsilon, \sigma)$ satisfies the following polynomial dependence on ε and σ:

$$\sigma_0(\varepsilon, \sigma) \leqslant \frac{\sigma}{2\left(16\sqrt{2}\sqrt{\frac{H}{\mu}} + 2\right)\log_2 \frac{\mu R_0^2}{\varepsilon}}. \tag{32}$$

2. (*Inexact solution of the auxiliary problem* (15).) *Algorithm* 2 *is applied to solve problem* (11) *and, in each iteration of Algorithm* 1 *used as a building block in Algorithm* 2*, an $(\widetilde{\varepsilon}_f, \widetilde{\sigma})$-solution to the auxiliary problem* (15) *is available, i. e., with probability at least $1 - \widetilde{\sigma}$*

$$\left(\left\langle \nabla\varphi_{\delta, L_\varphi}\left(x_k^{md}\right), x_{k+1}^t - x_k^{md}\right\rangle + \psi\left(x_{k+1}^t\right) + \frac{H}{2}\left\|x_{k+1}^t - x_k^{md}\right\|^2\right) - \\ - \min_{z \in \mathbb{R}^{d_x}}\left(\left\langle \nabla\varphi_{\delta, L_\varphi}\left(x_k^{md}\right), z - x_k^{md}\right\rangle + \psi(x) + \frac{H}{2}\left\|z - x_k^{md}\right\|^2\right) \leqslant \widetilde{\varepsilon}_f, \tag{33}$$

where $\widetilde{\varepsilon}_f(\varepsilon)$ *and* $\widetilde{\sigma}(\varepsilon, \sigma)$ *satisfy the following polynomial dependences on* ε *and* σ

$$\widetilde{\varepsilon}_f(\varepsilon) \leqslant \frac{\varepsilon\mu^2}{864^2(L+H)^2}, \tag{34}$$

$$\widetilde{\sigma}(\varepsilon, \sigma) \leqslant \frac{\sigma}{2\left(16\sqrt{2}\sqrt{\frac{H}{\mu}}+2\right)\log_2 \frac{\mu R_0^2}{\varepsilon}}. \tag{35}$$

Then, under the listed assumptions, Algorithm 2 *with* $K = 2\log_2 \frac{\mu R_0^2}{4\varepsilon}$ *guarantees that its output point* z_K *is an* (ε, σ)*-solution to problem* (11). *Moreover, the number* N_F *of the calls to inexact oracle both for* φ *and for* ψ *satisfies the following inequality*:

$$N_F \leqslant \left(16\sqrt{2}\sqrt{\frac{H}{\mu}}+2\right)\log_2 \frac{\mu R_0^2}{\varepsilon} = \widetilde{O}\left(\max\left\{\sqrt{\frac{H}{\mu}}, 1\right\}\right), \tag{36}$$

and the number of times the auxiliary problem (15) *is solved is also equal to* N_F.

We prove this theorem in the Appendix.

REMARK 1. We state the above theorem in the full generality. In the next sections we use its particular version with $\delta_1 = 0$.

Accelerated Framework for Saddle-Point Problems

In this section we consider the saddle-point problem with composite structure

$$\min_{x\in\mathbb{R}^{d_x}} \max_{y\in\mathbb{R}^{d_y}} \{f(x) + G(x, y) - h(y)\}. \tag{37}$$

We describe a general accelerated framework for this problem in order to use it to develop accelerated methods for saddle-point problems (1) and (2). As previously discussed, our general framework consists of several loops, which require us to solve optimization problems with some special structure. Thus, the general framework in this section is developed under two additional assumptions on two problems with a special structure (see Conditions 2, 3 below), which we need to solve in two loops of the framework. Then, in the following paragraphs we show, how these assumptions can be satisfied, which allows us to obtain the main results as a corollary of the main theorem of this section. In this section we introduce the main assumptions on the problem (37) and two additional assumptions for applying the framework. Next, we discuss the structure of the problem (37) and slightly reformulate it in an equivalent way. Then, we describe the main part of the framework by giving details of each loop, and finish with the main complexity theorem.

Preliminaries

We start with the main conditions that are used in the general framework. These are conditions on the functions f, G, h in the problem (37).

Condition 1.

1. *Function* f *is* L_f*-smooth,* μ_x*-strongly convex and there exists a basic oracle* O_f *for* f *such that* τ_f *calls of this basic oracle produce the gradient* $\nabla f(x)$.

2. *Function* $G(x, y)$ *is* L_G*-smooth, i. e. for each* $x = (x_1, x_2)$, $y = (y_1, y_2) \in \mathbb{R}^{d_x} \times \mathbb{R}^{d_y}$

$$\|\nabla G(x_1, x_2) - \nabla G(y_1, y_2)\| \leqslant L_G \|(x_1, x_2) - (y_1, y_2)\|, \tag{38}$$

there exists a basic oracle O_G^x *for* $G(\cdot, y)$ *such that* τ_G *calls of this basic oracle produce the gradient* $\nabla_x G(x, y)$ *and a basic oracle* O_G^y *for* $G(x, \cdot)$ *such that* τ_G *calls of this basic oracle produce the gradient* $\nabla_y G(x, y)$.

3. *Function* h *is* L_h*-smooth,* μ_y*-strongly convex and there exists a basic oracle* O_h *for* h *such that* τ_h *calls of this basic oracle produce the gradient* $\nabla h(y)$.

We will apply this general framework to solve, in particular, problem (1). This problem formulation is not symmetric w.r.t. the variables x and y since different assumptions are imposed on function f and function h. Our preliminary derivations, which we do not report here, have shown that better complexity bounds are obtained if we change the order of maximization in y and minimization in x in the problem (37) and write the following equivalent problem:

$$\min_{y \in \mathbb{R}^{d_y}} \left\{ h(y) + \max_{x \in \mathbb{R}^{d_x}} \{-G(x, y) - f(x)\} \right\}. \tag{39}$$

This reformulation allows us to solve problem (39) by an algorithm which consists of a series of inner-outer loops, where in each loop Algorithm 2 is applied to solve some auxiliary problem which has the form (11). The above equivalent reformulation of (37) naturally leads to the following lemma about approximate solutions.

Lemma 3. *Assume that Condition* 1 *for the problem* (37) *holds. Let an approximate solution* $(\widehat{x}, \widehat{y})$ *for* (39) *satisfy*

1. $\widehat{y}$ *is an* $(\varepsilon_y, \sigma_y)$*-solution to the outer problem in* (39)*, i. e.* (9) *holds*;

2. $\widehat{x}$ *is an* $(\varepsilon_x, \sigma_x)$*-solution to the inner problem* $\max\limits_{x \in \mathbb{R}^{d_x}} \{-G(x, \widehat{y}) - f(x)\}$.

Then, the following inequalities hold with probability at least $1 - \sigma_y - \sigma_x$:

$$\|\widehat{y} - y_\star\|^2 \leqslant \frac{2\varepsilon_y}{\mu_y}, \tag{40}$$

$$\|\widehat{x} - x_*\|^2 \leqslant 8\left(\frac{L_G}{\mu_x}\right)^2 \|\widehat{y} - y_*\|^2 + \frac{4\varepsilon_x}{\mu_x}, \tag{41}$$

$$\max_{x \in \mathbb{R}^{d_x}} \min_{y \in \mathbb{R}^{d_y}} \{h(y) - G(x, y) - f(x)\} - \min_{y \in \mathbb{R}^{d_y}} \{h(y) - G(\widehat{x}, y) - f(\widehat{x})\} \leqslant$$

$$\leqslant 2\left(L_f + L_G + \frac{2L_G^2}{\mu_y}\right)\left(\frac{\varepsilon_x}{\mu_x} + \left(\frac{L_G}{\mu_x}\right)^2 \frac{4\varepsilon_y}{\mu_y}\right), \tag{42}$$

where (x_*, y_*) *is the saddle point for problem* (37).

By swapping the variables x and y we can get a useful corollary that we will use later.

Corollary 1. *Assume that in the problem*

$$\min_{x \in \mathbb{R}^{d_x}} \left\{ f(x) + \max_{y \in \mathbb{R}^{d_y}} \{F(x, y) - w(y)\} \right\} \tag{43}$$

the functions f, F, w *are* L_f, L_F, L_w *— smooth and the functions* f, w *are* μ_x, μ_y*-strongly convex, respectively. Let an approximate solution* $(\widehat{x}, \widehat{y})$ *for* (43) *satisfy*

1. $\widehat{x}$ *is an* $(\varepsilon_x, \sigma_x)$*-solution to the outer problem* (43)*, i. e. inequality* (9) *holds*;

2. $\widehat{y}$ *is an* $(\varepsilon_y, \sigma_y)$*-solution to the inner problem* $\max\limits_{y\in\mathbb{R}^{d_y}}\{F(\widehat{x}, y) - w(y)\}$.

Then, the following inequalities hold with probability at least $1 - \sigma_y - \sigma_x$:

$$\|\widehat{x} - x_\star\|^2 \leqslant \frac{2\varepsilon_x}{\mu_x}, \tag{44}$$

$$\|\widehat{y} - y_*\|^2 \leqslant 8\left(\frac{L_F}{\mu_y}\right)^2 \|\widehat{x} - x_*\|^2 + \frac{4\varepsilon_y}{\mu_y}, \tag{45}$$

$$w(\widehat{y}) + \max_{x\in\mathbb{R}^{d_x}}\{-F(x, \widehat{y}) - f(x)\} - \min_{y\in\mathbb{R}^{d_y}}\max_{x\in\mathbb{R}^{d_x}}\{w(y) - F(x, y) - f(x)\} = \tag{46}$$

$$= \max_{y\in\mathbb{R}^{d_y}}\min_{x\in\mathbb{R}^{d_x}}\{f(x) + F(x, y) - w(y)\} - \min_{x\in\mathbb{R}^{d_x}}\{f(x) + F(x, \widehat{y}) - w(\widehat{y})\} \leqslant \tag{47}$$

$$\leqslant 2\left(L_w + L_F + \frac{2L_F^2}{\mu_x}\right)\left(\frac{\varepsilon_y}{\mu_y} + 4\left(\frac{L_F}{\mu_y}\right)^2\frac{\varepsilon_x}{\mu_x}\right), \tag{48}$$

where (x_*, y_*) *is the saddle point for problem* (43).

The next two assumptions are made to obtain a general framework. In this section we assume that two functions defined via auxiliary maximization problems and arising in loops of our general framework can be provided with an inexact oracle. In the following sections we show how to satisfy these two conditions and apply the general framework.

Condition 2. *Let* $\varepsilon > 0$ *and* $\sigma \in (0, 1)$*, and a function* g *be defined as*

$$g(x) = \max_{y\in\mathbb{R}^{d_y}}\left\{G(x, y) - h(y) - \frac{H}{2}\|y - y_0\|^2\right\}, \tag{49}$$

where $G(x, y)$*,* $h(y)$ *satisfy Condition 1,* $H > 0$*, and* y_0 *is some fixed point in* $\mathbb{R}^{d_y}$.

Then, we assume that, for any $\delta(\varepsilon) = poly(\varepsilon)$ *and any* $\sigma_0(\varepsilon, \sigma) = poly(\varepsilon, \sigma)$*, it is possible to evaluate a* $\left(\frac{\delta(\varepsilon)}{2}, \sigma_0(\varepsilon, \sigma)\right)$*-solution to this problem and* $\left(\delta(\varepsilon), \sigma_0(\varepsilon, \sigma), 2L_G + 4\frac{L_G^2}{\mu_y + H}\right)$*-oracle for the function* g*. Moreover, we assume that this solution can be evaluated using* $\mathcal{N}_G^y(\tau_G, H)\mathcal{K}_G^y(\varepsilon, \sigma)$ *calls of the basic oracle* O_G^y *of* $G(x, \cdot)$*,* $\mathcal{N}_h(\tau_h, H)\mathcal{K}_h(\varepsilon, \sigma)$ *calls of the basic oracle* O_h *of* h *and this inexact oracle can be evaluated using* $\mathcal{N}_G^y(\tau_G, H)\mathcal{K}_G^y(\varepsilon, \sigma)$ *calls of the basic oracle* O_G^y *of* $G(x, \cdot)$*,* τ_G *calls of the basic oracle* O_G^x *of* $G(\cdot, y)$ *and* $\mathcal{N}_h(\tau_h, H)\mathcal{K}_h(\varepsilon, \sigma)$ *calls of the basic oracle* O_h *of* h*, where* $\mathcal{K}_G^y(\varepsilon, \sigma) = \widetilde{O}(1)$ *and* $\mathcal{K}_h(\varepsilon, \sigma) = \widetilde{O}(1)$.

Condition 3. *Let* $\varepsilon > 0$ *and* $\sigma \in (0, 1)$*, and a function* r *be defined as*

$$r(y) = \min_{x\in\mathbb{R}^{d_x}}\{G(x, y) + f(x)\}, \tag{50}$$

where $G(x, y)$*,* $f(x)$ *satisfy Condition* 1.

Then, we assume that, for any $\delta(\varepsilon) = poly(\varepsilon)$ *and any* $\sigma_0(\varepsilon, \sigma) = poly(\varepsilon, \sigma)$*, it is possible to evaluate a* $\left(\frac{\delta(\varepsilon)}{2}, \sigma_0(\varepsilon, \sigma)\right)$*-solution to this problem and* $\left(\delta(\varepsilon), \sigma_0(\varepsilon, \sigma), 2L_G + 4\frac{L_G^2}{\mu_x}\right)$*-oracle for the function* r *in the sense of Definition 2 with* $\delta_1 = 0$*. Moreover, we assume that this solution can be evaluated using* $\mathcal{N}_G^x(\tau_G)\mathcal{K}_G^x(\varepsilon, \sigma)$ *calls of the basic oracle* O_G^x *for* $G(\cdot, y)$*,* $\mathcal{N}_f(\tau_f)\mathcal{K}_f(\varepsilon, \sigma)$ *calls of the basic oracle* O_f *for* f *and this inexact oracle can be evaluated using* τ_G *calls of the basic oracle* O_G^y *of* $G(x, \cdot)$*,* $\mathcal{N}_G^x(\tau_G)\mathcal{K}_G^x(\varepsilon, \sigma)$ *calls of the basic oracle* O_G^x *for* $G(\cdot, y)$ *and* $\mathcal{N}_f(\tau_f)\mathcal{K}_f(\varepsilon, \sigma)$ *calls of the basic oracle* O_f *for* f*, where* $\mathcal{K}_G^x(\varepsilon, \sigma) = \widetilde{O}(1)$ *and* $\mathcal{K}_f(\varepsilon, \sigma) = \widetilde{O}(1)$.

Let us shortly illustrate how this can be achieved by a simple example.

EXAMPLE 1. Assume, for simplicity, that in (49) $h = 0$ and the full gradients $\nabla_x G(x, y)$, $\nabla_y G(x, y)$ are available, meaning that in Condition 1 $\tau_G = 1$. Then, the objective in the maximization problem (49) has L_G-smooth in y part $G(x, y)$ and H-strongly concave part $-\frac{H}{2}\|y - y_0\|^2$. Thus, if we apply the accelerated gradient method for composite optimization [Nesterov, 2013], we find that a $\frac{\delta(\varepsilon)}{2}$-solution $\widetilde{y}_{\delta(\varepsilon)/2}(x)$ to this problem can be obtained in $O\left(\sqrt{\frac{L_G}{H}} \ln \frac{1}{\delta(\varepsilon)}\right)$ iterations of the accelerated method. Each iteration requires us to evaluate $\nabla_y G(x, y)$, which means that the number of calls of the basic oracle O_G^y for $G(x, \cdot)$ is $O\left(\tau_G \sqrt{\frac{L_G}{H}} \ln \frac{1}{\delta(\varepsilon)}\right)$. Since $\delta(\varepsilon) = \text{poly}(\varepsilon)$, we find that the number of O_G^y calls is $\mathcal{N}_G^y(\tau_G, H)\mathcal{K}_G^y(\varepsilon, \sigma) = O\left(\tau_G \sqrt{\frac{L_G}{H}} \ln \frac{1}{\varepsilon}\right)$, i. e. $\mathcal{K}_G^y(\varepsilon, \sigma) = \widetilde{O}(1)$. Moreover, by Lemma 2, $\nabla_x G\left(x, \widetilde{y}_{\delta(\varepsilon)/2}(x)\right)$ is $\left(\delta(\varepsilon), 2L_G + \frac{2L_G^2}{H}\right)$-oracle for the function g, which means that we need also τ_G calls of the basic oracle O_G^x for $G(\cdot, y)$. Thus, Condition 2 holds.

General framework for saddle-point problems

Next, we describe in detail the resulting structure of our framework which consists of three inner-outer loops. We also summarize the steps of the algorithm in Table 4. In each loop we apply Algorithm 2 with different value of parameter H, which defines its complexity. In the next subsection we carefully choose the value of this parameter at each level of the loops. Later, in the next sections this allows us to obtain the desired results on near-optimal complexity bounds for saddle-point problems (1) or (2). Further, in each loop we have a target accuracy ε and a confidence level σ which define the required quality of the solution to an optimization problem in this loop. These quantities define the inexactness of the oracle in this loop via inequalities (31) and (32) and the target accuracy and confidence level for the optimization problem in the next loop via (34), (35). Due to inexact strong convexity provided by (δ, σ, L, μ)-oracle, Algorithm 2 has logarithmic dependence of the complexity on the target accuracy and confidence level (see Theorem 4). Since the dependences on the target accuracy and confidence level in (31), (32), (34) and (35) are polynomial, we find that the dependence of the complexity in each loop on the target accuracy and confidence level in the first loop, i. e. target accuracy and confidence level for the solution to problem (37), is logarithmic.

Loop 1

The goal of this loop is to find an (ε, σ)-solution of problem (39). This problem is reformulated as a minimization problem in y with the objective given in the form of an auxiliary maximization problem in x.

Finding an (ε, σ)-solution of this minimization problem gives an approximate solution to the saddle-point problem (37).

To find this pair, we solve problem (39) using Algorithm 2 with

$$\varphi = 0, \quad \psi = h(y) + \max_{x \in \mathbb{R}^{d_x}} \{-G(x, y) - f(x)\} \tag{51}$$

and parameter $H = H_1$ to be chosen later. The function φ is, clearly, convex and is known exactly. What makes solving problem (39) not straightforward is that the exact value of ψ is not available. At the same time we can construct an inexact oracle for this function. The function h is μ_y-strongly convex, L_h-smooth and its exact gradient is available. By Condition 3 it is possible to construct a $\left(\delta^{(1)}(\varepsilon), \sigma_0^{(1)}(\varepsilon, \sigma), 2L_G + 4\frac{L_G^2}{\mu_x}\right)$-oracle for the function $r(y) = \max\limits_{x \in \mathbb{R}^{d_x}} \{-f(x) - G(x, y)\}$ for any $\delta^{(1)}(\varepsilon) = \text{poly}(\varepsilon)$ and $\sigma_0^{(1)}(\varepsilon, \sigma) = \text{poly}(\varepsilon, \sigma)$. Using Lemma 1, we find that we can construct a $\left(\delta^{(1)}(\varepsilon), \sigma_0^{(1)}(\varepsilon, \sigma), L_h + 2L_G + 4\frac{L_G^2}{\mu_x}, \mu_y\right)$-oracle for ψ. Let $\delta^{(1)}(\varepsilon)$ and $\sigma_0^{(1)}(\varepsilon, \sigma)$ satisfy (31) and (32).

In each iteration k of Algorithm 1 the problem

$$y_{k+1}^t = \arg\min_{y\in\mathbb{R}^{d_y}} \left\{ h(y) + \max_{x\in\mathbb{R}^{d_x}} \{-G(x, y) - f(x)\} + \frac{H_1}{2}\left\|y - y_k^{md}\right\|^2 \right\}, \tag{52}$$

needs to be solved inexactly. Assume that, for each k we can find an $\left(\widetilde{\varepsilon}_f^{(1)}(\varepsilon), \widetilde{\sigma}^{(1)}(\varepsilon, \sigma)\right)$-solution to the problem (15), where $\widetilde{\sigma}^{(1)}(\varepsilon, \sigma)$, $\widetilde{\varepsilon}_f^{(1)}(\varepsilon)$ satisfy (34) and (35) respectively. Applying Theorem 4, we find that we require $\widetilde{O}\left(1 + \left(\frac{H_1}{\mu_y}\right)^{1/2}\right)$ iterations of Algorithm 1, $\widetilde{O}\left(1 + \left(\frac{H_1}{\mu_y}\right)^{1/2}\right)\tau_h$ calls of the basic oracle for h, $\widetilde{O}\left(1 + \left(\frac{H_1}{\mu_y}\right)^{1/2}\right)\tau_G$ calls of the basic oracle of $G(x, \cdot)$, $\widetilde{O}\left(1 + \left(\frac{H_1}{\mu_y}\right)^{1/2}\right)\mathcal{N}_G^x(\tau_G)\mathcal{K}_G^x(\varepsilon, \sigma)$ calls of the basic oracle for $G(\cdot, y)$, $\widetilde{O}\left(1 + \left(\frac{H_1}{\mu_y}\right)^{1/2}\right)\mathcal{N}_f(\tau_f)\mathcal{K}_f(\varepsilon, \sigma)$ calls of the basic oracle for f.

It remains to show how to find the $\left(\widetilde{\varepsilon}_f^{(1)}(\varepsilon), \widetilde{\sigma}^{(1)}(\varepsilon, \sigma)\right)$-solution to problem (52) in each iteration of Algorithm 1. This is organized in Loop 2 below.

Loop 2

As mentioned in the previous Loop 1, in each iteration of Algorithm 2 in Loop 1 we need many times to find an $(\varepsilon_2', \sigma_2')$-solution of the auxiliary problem (52), where we denoted for simplicity $\sigma_2' = \widetilde{\sigma}^{(1)}(\varepsilon, \sigma)$ and $\varepsilon_2' = \widetilde{\varepsilon}_f^{(1)}(\varepsilon)$. To that end, we reformulate problem (52) as follows:

$$\min_{x\in\mathbb{R}^{d_x}} \left\{ f(x) + \max_{y\in\mathbb{R}^{d_y}} \left\{ G(x, y) - h(y) - \frac{H_1}{2}\left\|y - y_k^{md}\right\|^2 \right\} \right\}. \tag{53}$$

Assume that we can find an $(\varepsilon_2, \sigma_2)$-solution $\widehat{x}$ of the minimization problem (53). By Condition 2 we can also obtain a point $\widehat{y}$ which is a $(\frac{\overline{\delta}(\varepsilon_2)}{2}, \overline{\sigma}_0(\sigma_2))$-solution to the inn problem

$$\max_{y\in\mathbb{R}^{d_y}} \left\{ G(x, y) - h(y) - \frac{H_1}{2}\left\|y - y_k^{md}\right\|^2 \right\}, \tag{54}$$

where $\overline{\delta}(\varepsilon_2)$, $\overline{\sigma}_0(\sigma_2)$ satisfy the following polynomial dependences:

$$\overline{\delta}(\varepsilon_2) \leqslant \frac{H_1 + \mu_y}{4\mu_x\left(\frac{H_1+\mu_y}{4L_G}\right)^2}\varepsilon_2, \quad \overline{\sigma}_0(\sigma_2) \leqslant \sigma_2. \tag{55}$$

If we choose ε_2, σ_2, $\overline{\delta}(\varepsilon_2)$, $\overline{\sigma}_0(\sigma_2)$ satisfying

$$\varepsilon_2 \leqslant \left(\frac{H_1 + \mu_y}{4L_G}\right)^2 \frac{\mu_x}{L_h + H_1 + L_G + \frac{2L_G^2}{\mu_x}}\varepsilon_2', \tag{56}$$

$$\sigma_2 \leqslant \frac{\sigma_2'}{2}, \tag{57}$$

$$\overline{\sigma}_0(\sigma_2) \overset{(55)}{\leqslant} \sigma_2 \leqslant \frac{\sigma_2'}{2}, \quad \overline{\delta}(\varepsilon_2) \leqslant \frac{H_1 + \mu_y}{4\mu_x\left(\frac{H_1+\mu_y}{4L_G}\right)^2}\varepsilon_2 \overset{(55)}{\leqslant} \frac{H_1 + \mu_y}{4L_h + 4H_1 + 4L_G + \frac{8L_G^2}{\mu_x}}\varepsilon_2', \tag{58}$$

then

$$2\frac{L_h + H_1 + L_G + \frac{2L_G^2}{\mu_x}}{H_1 + \mu_y}\overline{\delta}(\varepsilon_2) + 8\left(\frac{L_G}{H_1 + \mu_y}\right)^2 \frac{L_h + H_1 + L_G + \frac{2L_G^2}{\mu_x}}{\mu_x}\varepsilon_2 \leqslant \varepsilon_2', \tag{59}$$

$$\sigma_2 + \overline{\sigma}_0(\sigma_2) \leqslant \sigma_2'. \tag{60}$$

Thus, applying Corollary 1 to the minimization problem (53) with $F(x, y) = G(x, y)$, $w(y) = h(y) + \frac{H_1}{2}\left\|y - y_k^{md}\right\|^2$, $\varepsilon_x = \varepsilon_2$, $\sigma_x = \sigma_2$, $\varepsilon_y = \overline{\delta}(\varepsilon_2)$, $\sigma_y = \overline{\sigma}_0(\sigma_2)$ we find (see (46), (48)) that $\widehat{y}$ satisfies the inequality

$$h(\widehat{y}) + \frac{H_1}{2}\left\|\widehat{y} - y_k^{md}\right\|^2 + \max_{x\in\mathbb{R}^{d_x}}\{-G(x, \widehat{y}) - f(x)\} - \min_{y\in\mathbb{R}^{d_y}}\max_{x\in\mathbb{R}^{d_x}}\left\{h(y) + \frac{H_1}{2}\left\|y - y_k^{md}\right\|^2 - G(x, y) - f(x)\right\} \leqslant \varepsilon_2' \quad (61)$$

with probability at least σ_2'. Thus, it is an $(\varepsilon_2', \sigma_2')$-solution of the problem (52). By Assumption 2, calculation of $\widehat{y}$ requires $\mathcal{N}_G^y(\tau_G, H)\mathcal{K}_G^y(\varepsilon_2, \sigma_2)$ calls of the basic oracle O_G^y of $G(x, \cdot)$, τ_G calls of the basic oracle O_G^x of $G(\cdot, y)$ and $\mathcal{N}_h(\tau_h, H)\mathcal{K}_h(\varepsilon_2, \sigma_2)$ calls of the basic oracle O_h of h.

Our next step is to provide an $(\varepsilon_2, \sigma_2)$-solution to the minimization problem (53) and we apply Algorithm 2 with

$$\varphi = \max_{y\in\mathbb{R}^{d_y}}\left\{G(x, y) - h(y) - \frac{H_1}{2}\left\|y - y_k^{md}\right\|^2\right\}, \quad \psi = f(x). \quad (62)$$

The function ψ is μ_x-strongly convex, L_f-smooth and its exact gradient is available. What makes solving problem (53) not straightforward is that the exact value of φ is not available. At the same time we can construct an inexact oracle for this function. Thanks to Assumption 2, it is possible to construct a $\left(\delta^{(2)}(\varepsilon_2), \sigma_0^{(2)}(\varepsilon_2, \sigma_2), 2L_G + 4\frac{L_G^2}{H_1+\mu_y}\right)$-oracle for the function φ for any $\delta^{(2)}(\varepsilon_2) = \mathrm{poly}(\varepsilon_2)$ and $\sigma_0^{(2)}(\varepsilon_2, \sigma_2) = \mathrm{poly}(\varepsilon_2, \sigma_2)$. Using Lemma 1, we find that we can construct a $\left(\delta^{(2)}(\varepsilon_2), \sigma_0^{(2)}(\varepsilon_2, \sigma_2), L_f + 2L_G + 4\frac{L_G^2}{H_1+\mu_y}, \mu_x\right)$-oracle for the function $\varphi + \psi$. Thus, we can apply Algorithm 2 with parameter $H = H_2 \geqslant 2L_G + 4\frac{L_G^2}{H_1+\mu_y}$, which will be chosen later, to solve the problem (53). Moreover, since Assumption 2 requires $\delta^{(2)}(\varepsilon_2) = \mathrm{poly}(\varepsilon_2)$ and $\sigma_0^{(2)}(\varepsilon_2, \sigma_2) = \mathrm{poly}(\varepsilon_2, \sigma_2)$, which holds for the dependences in (31) and (32), we can choose $\delta^{(2)}(\varepsilon_2)$ and $\sigma_0^{(2)}(\varepsilon_2, \sigma_2)$ such that (31) and (32) hold. So, the first main assumption of Theorem 4 holds. At the same time, according to Assumptions 1 and 2, constructing inexact oracle for φ requires $\mathcal{N}_G^y(\tau_G, H_1)\mathcal{K}_G^y(\varepsilon_2, \sigma_2)$ calls of the basic oracle for $G(x, \cdot)$, τ_G calls of the basic oracle for $G(\cdot, y)$, $\mathcal{N}_h(\tau_h, H_1)\mathcal{K}_h(\varepsilon_2, \sigma_2)$ calls of the basic oracle for h, and constructing exact oracle for $\psi = f$ requires τ_f calls of the basic oracle for f.

Let us discuss the second main assumption of Theorem 4. To ensure that this assumption holds, we need in each iteration of Algorithm 1, used as a building block in Algorithm 2, to find the $\left(\widetilde{\varepsilon}_f^{(2)}(\varepsilon_2), \widetilde{\sigma}^{(2)}(\varepsilon_2, \sigma_2)\right)$-solution to the auxiliary problem (15), where $\widetilde{\sigma}^{(2)}(\varepsilon_2, \sigma_2)$, $\widetilde{\varepsilon}_f^{(2)}(\varepsilon_2)$ satisfy inequalities (34) and (35). For the particular definitions of φ, ψ (62) in this Loop, this problem has the following form:

$$x_{l+1}^t = \operatorname*{argmin}_{u\in\mathbb{R}^{d_x}}\left\{\left\langle\nabla\varphi_{\delta^{(2)},2L_\varphi}\left(x_l^{md}\right), x - x_l^{md}\right\rangle + \psi(x) + \frac{H_2}{2}\left\|x - x_l^{md}\right\|_2^2\right\} = \quad (63)$$

$$= \operatorname*{argmin}_{x\in\mathbb{R}^{d_x}}\left\{\left\langle\nabla g_{\delta^{(2)},2L_g}\left(x_l^{md}\right), x - x_l^{md}\right\rangle + f(x) + \frac{H_2}{2}\left\|x - x_l^{md}\right\|^2\right\}, \quad (64)$$

where $g(x) = \max_{y\in\mathbb{R}^{d_y}}\left\{G(x, y) + h(y) - \frac{H_1}{2}\left\|y - y_k^{md}\right\|^2\right\}$, $L_g = L_G + 2\frac{L_G^2}{H_1+\mu_y}$. Below, in the next section "Loop 3", we explain how to solve this auxiliary problem to obtain its $\left(\widetilde{\varepsilon}_f^{(2)}(\varepsilon_2), \widetilde{\sigma}^{(2)}(\varepsilon_2, \sigma_2)\right)$-solution.

To summarize Loop 2, both main assumptions of Theorem 4 hold and we can use it to guarantee that we obtain an $(\varepsilon_2', \sigma_2')$-solution of the auxiliary problem (52). This requires us one

time to solve the problem (54), which, by Assumption 2 has the same cost as evaluating inexact oracle for the function φ. Further, we need $O\left(\left(1+\left(\frac{H_2}{\mu_\varphi+\mu_\psi}\right)^{1/2}\right)\log\varepsilon_2^{-1}\right) = O\left(\left(1+\left(\frac{H_2}{\mu_x}\right)^{1/2}\right)\log\varepsilon_2^{-1}\right)$ calls to the inexact oracles for φ and for ψ, and the same number of times solving the auxiliary problem (63). Combining this oracle complexity with the cost of calculating inexact oracles for φ and for ψ, we find that solving the problem (53) requires $O\left(\left(1+\left(\frac{H_2}{\mu_x}\right)^{1/2}\right)\log\varepsilon_2^{-1}\right)\tau_f$ calls of the basic oracle for f, $O\left(\left(1+\left(\frac{H_2}{\mu_x}\right)^{1/2}\right)\log\varepsilon_2^{-1}\right)\mathcal{N}_G^y(\tau_G, H_1)\mathcal{K}_G^y(\varepsilon_2, \sigma_2)$ calls of the basic oracle for $G(x, \cdot)$, $O\left(\left(1+\left(\frac{H_2}{\mu_x}\right)^{1/2}\right)\log\varepsilon_2^{-1}\right)\tau_G$ calls of the basic oracle for $G(\cdot, y)$, $O\left(\left(1+\left(\frac{H_2}{\mu_x}\right)^{1/2}\right)\log\varepsilon_2^{-1}\right)\mathcal{N}_h(\tau_h, H_1)\mathcal{K}_h(\varepsilon_2, \sigma_2)$ calls of the basic oracle for h. The only remaining thing is to provide an inexact solution to problem (63) and, next, we move to Loop 3 to explain how to guarantee this. Note that we need to solve problem (63) $O\left(\left(1+\left(\frac{H_2}{\mu_x}\right)^{1/2}\right)\log\varepsilon_2^{-1}\right)$ times.

Loop 3

As mentioned in the previous Loop 2, in each iteration of Algorithm 2 in Loop 2 we need to find many times an $(\varepsilon_3, \sigma_3)$-solution of the auxiliary problem (63), where we denoted for simplicity $\sigma_3 = \widetilde{\sigma}^{(2)}(\varepsilon_2, \sigma_2)$ and $\varepsilon_3 = \widetilde{\varepsilon}_f^{(2)}(\varepsilon_2)$. To solve problem (63), we would like to apply Algorithm 2 with

$$\varphi = f(x), \quad \psi = \left\langle \nabla g_{\delta^{(2)},2L_g}\left(x_l^{md}\right), x - x_l^{md}\right\rangle + \frac{H_2}{2}\left\|x - x_l^{md}\right\|^2, \tag{65}$$

where $g(x) = \max\limits_{y\in\mathbb{R}^{d_y}}\left\{G(x, y) + h(y) - \frac{H_1}{2}\left\|y - y_k^{md}\right\|^2\right\}$, $L_g = L_G + 2\frac{L_G^2}{H_1+\mu_y}$.

The function φ is μ_x-strongly convex, L_f-smooth and its exact gradient is available. The function ψ is, clearly, H_2-strongly convex, H_2-smooth and its exact gradient is available. Also, we can obtain the exact gradient for the function $\varphi + \psi$.

Thus, we can apply Algorithm 2 with parameter $H = H_3 \geqslant L_f$, which will be chosen later, to solve problem (63). The first main assumption of Theorem 4, clearly, holds. At the same time, constructing an exact oracle for $\varphi = f$ requires τ_f calls of the basic oracle for f. At the same time, no calls to the oracle for $G(\cdot, y)$, $G(x, \cdot)$, h are needed.

Let us discuss the second main assumption of Theorem 4. To ensure that this assumption holds, we need in each iteration of Algorithm 1, used as a building block in Algorithm 2, to find $\left(\widetilde{\varepsilon}_f^{(3)}(\varepsilon_3), \widetilde{\sigma}^{(3)}(\varepsilon_3, \sigma_3)\right)$-solution to the auxiliary problem (15), where $\widetilde{\sigma}^{(3)}(\varepsilon_3, \sigma_3)$, $\widetilde{\varepsilon}_f^{(3)}(\varepsilon_3)$ satisfy inequalities (34) and (35). For the particular definitions of φ, ψ in (65) in this loop, this problem has the following form:

$$u_{m+1}^t = \operatorname*{argmin}_{u\in\mathbb{R}^{d_x}}\left\{\left\langle\nabla\varphi\left(u_m^{md}\right), u - u_m^{md}\right\rangle + \psi(u) + \frac{H_3}{2}\left\|u - u_m^{md}\right\|_2^2\right\} =$$

$$= \operatorname*{argmin}_{u\in\mathbb{R}^{d_x}}\left\{\left\langle\nabla f\left(u_m^{md}\right), u - u_m^{md}\right\rangle + \left\langle\nabla g_{\delta^{(2)},2L_g}\left(x_l^{md}\right), u - x_l^{md}\right\rangle + \frac{H_2}{2}\left\|u - x_l^{md}\right\|^2 + \frac{H_3}{2}\left\|u - u_m^{md}\right\|_2^2\right\}, \tag{66}$$

where $g(x) = \max\limits_{y\in\mathbb{R}^{d_y}}\left\{G(x, y) + h(y) - \frac{H_1}{2}\left\|y - y_k^{md}\right\|^2\right\}$, $L_g = L_G + 2\frac{L_G^2}{H_1+\mu_y}$.

This quadratic auxiliary problem (66) can be solved explicitly and exactly since at the point it needs to be solved, $\nabla g_{\delta^{(2)},2L_g}(x_l^{md})$ is already calculated. Thus, the second main assumption of Theorem 4 is satisfied with $\widetilde{\sigma}^{(3)}(\varepsilon_3, \sigma_3) = 0$ and $\widetilde{\varepsilon}_f^{(3)}(\varepsilon_3) = 0$, which clearly satisfy (31) and (32).

To summarize Loop 3, both main assumptions of Theorem 4 hold and we can use it to guarantee that we obtain an $(\varepsilon_3, \sigma_3)$-solution of the auxiliary problem (63). This requires $O\left(\left(1 + \left(\frac{H_3}{\mu_\varphi + \mu_\psi}\right)^{1/2}\right)\log \varepsilon_3^{-1}\right) = O\left(\left(1 + \left(\frac{H_3}{H_2}\right)^{1/2}\right)\log \varepsilon_3^{-1}\right)$ calls to the inexact oracles for φ and for ψ, and the same number of times solving the auxiliary problem (66). Combining this oracle complexity with the cost of calculating inexact oracles for φ and for ψ, we find that solving the problem (63) requires $O\left(\left(1 + \left(\frac{H_3}{H_2}\right)^{1/2}\right)\log \varepsilon_3^{-1}\right)\tau_f$ calls of the basic oracle for f.

Table 4. Summary of the three loops of the general framework described above

	Goal	φ, ψ	μ in Th. 4	**Iteration number of Algorithm 1** (Th. 4)	**Each iteration requires**
Loop 1	(ε, σ)-solution of problem (39)	(51)	μ_y	$\widetilde{O}\left(1 + \sqrt{\frac{H_1}{\mu_y}}\right)$	Find $(\varepsilon_1, \sigma_1)$-solution of (52) and calculate $\left(\delta^{(1)}, L_\psi\right)$-oracle of $\psi(y)$
Loop 2	$(\varepsilon_1, \sigma_1)$-solution of problem (53)	(62)	μ_x	$\widetilde{O}\left(1 + \sqrt{\frac{H_2}{\mu_x}}\right)$	Find $(\varepsilon_2, \sigma_2)$-solution of (63) and calculate $\left(\delta^{(2)}, L_\varphi\right)$-oracle of $\varphi(x)$
Loop 3	$(\varepsilon_2, \sigma_2)$-solution of problem (63)	(65)	H_2	$\widetilde{O}\left(1 + \sqrt{\frac{H_3}{H_2}}\right)$	Find $(\varepsilon_3, \sigma_3)$-solution of (66)

Complexity of the general framework

Below we formally finalize in Theorem 5 the analysis of the general framework by carefully combining the bounds from loops to obtain the final bounds for the total number of oracle calls for each part f, G, h of the objective in problem (37). We will use Theorem 5 in the following sections to obtain complexity results for problems with structure as in (1) and (2).

Theorem 5. *Let Assumptions* 1, 2, 3 *hold. Then, execution of the general optimization framework described in sections "Loop* 1*" – "Loop* 3*" with*

$$H_1 = 2L_G, \quad H_2 = 2\left(L_G + \frac{2L_G^2}{\mu_y + H_1}\right), \quad H_3 = 2L_f$$

generates an (ε, σ)*-solution to the problem* (37)*, i. e. satisfies* (10)*. Moreover, for the number of basic oracle calls it holds that*

Number of calls of basic oracle O_f *for* f *is*

$$\widetilde{O}\left(\left(1 + \sqrt{\frac{L_G}{\mu_y}}\right)\left(\mathcal{N}_f(\tau_f) + \left(1 + \sqrt{\frac{L_G}{\mu_x}}\right)\left(1 + \sqrt{\frac{L_f}{L_G}}\right)\cdot \tau_f\right)\right), \tag{67}$$

Number of calls of basic oracle O_h *for* h *is*

$$\widetilde{O}\left(\left(1 + \sqrt{\frac{L_G}{\mu_y}}\right)\left(\tau_h + \left(1 + \sqrt{\frac{L_G}{\mu_x}}\right)\mathcal{N}_h\left(\tau_h, 2L_G\right)\right)\right), \tag{68}$$

Number of calls of basic oracle O_G^x for $G(\cdot, y)$ is

$$\widetilde{O}\left(\left(1+\sqrt{\frac{L_G}{\mu_y}}\right)\left(\mathcal{N}_G^x(\tau_G)+\left(1+\sqrt{\frac{L_G}{\mu_x}}\right)\tau_G\right)\right), \tag{69}$$

Number of calls of basic oracle O_G^y for $G(x, \cdot)$ is

$$\widetilde{O}\left(\left(1+\sqrt{\frac{L_G}{\mu_y}}\right)\left(\tau_G+\left(1+\sqrt{\frac{L_G}{\mu_x}}\right)\mathcal{N}_G^y\left(\tau_G, 2L_G\right)\right)\right). \tag{70}$$

Accelerated Method for Saddle-Point Problems

In this section, we consider problem (1) which is problem (37) with a specific finite-sum structure of the function h and our goal is to obtain its (ε, σ)-solution. To get the final estimates for the number of oracles calls, we need to satisfy Assumptions 1, 2, 3 which are formulated in Section "Accelerated Framework for Saddle-Point Problems" where we construct our general framework. So, the plan of this section is first to prove Lemma 4 and Corollary 2, which guarantee that Assumptions 2, 3 hold. To satisfy Assumption 2 we use a two-loop procedure with Algorithm 2 and the stochastic variance reduction method to solve problem (49) in order to use the finite-sum structure of the function h and avoid expensive calculation of the gradient of the whole sum in each iteration. As a corollary, we also show how to satisfy Assumption 3. Then, we obtain final estimates for the setting of this section by combining the complexities to satisfy Assumptions 2, 3 with the estimates in Theorem 5.

Problem statement

In this section we consider the optimization problem of the form (1):

$$\min_{x\in\mathbb{R}^{d_x}}\max_{y\in\mathbb{R}^{d_y}}\{f(x)+G(x, y)-h(y)\}, \quad h(y):=\frac{1}{m_h}\sum_{i=1}^{m_h}h_i(y) \tag{71}$$

and develop accelerated optimization methods for its solution under the following assumptions.

Condition 4.

1. *Function $f(x)$ is L_f-smooth and μ_x-strongly convex.*
2. *Function $G(x, y)$ is L_G-smooth, i. e. for each $(x_1, x_2), (y_1, y_2) \in \mathbb{R}^{d_x}\times\mathbb{R}^{d_y}$*

$$\|\nabla G(x_1, x_2)-\nabla G(y_1, y_2)\| \leqslant L_G\|(x_1, x_2)-(y_1, y_2)\|. \tag{72}$$

3. *$m_h \geqslant 1$ and each function $h_i(x)$, $i \in 1, \ldots, m_h$ is L_h^i-smooth and convex, function $h(y)$ is μ_y-strongly convex. We also define $L_h = \frac{1}{m_h}\sum\limits_{i=1}^{m_h}L_h^i$ in this case.*

To fit Condition 1 we consider the full gradient oracles $\nabla_x G(x, y)$, $\nabla_y G(x, y)$, $\nabla f(x)$ as the basic oracles O_G^x, O_G^y, O_f, respectively, and the stochastic gradient oracle $\nabla h_i(y)$ as the basic oracle O_h. Then Condition 4 guarantees that Condition 1 holds with

$$\tau_f = \tau_G = 1, \quad \tau_h = m_h. \tag{73}$$

Preliminaries

We start with two auxiliary results, which show how Assumptions 2 and 3 can be satisfied in the setting of this section. The first lemma provides complexity for inexact solution of the maximization problem (49) and the complexity of finding an inexact oracle for function g defined in the same equation.

Lemma 4. *Let the function g be defined via the maximization problem in* (49)*, i. e.*

$$g(x) = \max_{y \in \mathbb{R}^{d_y}} \left\{ G(x, y) - h(y) - \frac{H}{2}\|y - y_0\|^2 \right\}, \tag{74}$$

where $G(x, y)$, $h(y)$ correspond to (71) *and satisfy Condition* 4*, $y_0 \in \mathbb{R}^{d_y}$. Assume also that $m_h(H + 2L_G + \mu_y) \leqslant L_h$ and $H + \mu_y \leqslant 4L_G$. Then, organizing computations in two loops and applying Algorithm* 2 *in the outer loop and accelerated variance reduction method L-SVRG from* [Morin, Giselsson, 2020] *in the inner loop, we guarantee Condition* 2 *with $\tau_G = 1$ basic oracle calls for $G(\cdot, y)$ and the following estimates for the number of basic oracle calls for $G(x, \cdot)$ and h, respectively*:

$$\mathcal{N}^y_G(\tau_G, H) = O\left(1 + \sqrt{\frac{L_G}{H + \mu_y}}\right), \tag{75}$$

$$\mathcal{N}_h(\tau_h, H) = O\left(\sqrt{\frac{\tau_h L_h}{H + \mu_y}}\right). \tag{76}$$

Proof. To satisfy Condition 2, we need to provide an $\left(\frac{\delta(\varepsilon)}{2}, \sigma_0(\varepsilon, \sigma)\right)$-solution to the problem (74) and $(\delta(\varepsilon), \sigma_0(\varepsilon, \sigma), 2L_g)$-oracle of g in (74), where $L_g = L_G + \frac{2L_G^2}{\mu_y + H}$.

By Lemma 2 with $F(x, y) = G(x, y)$, $w(y) = h(y) + \frac{H}{2}\|y - y_0\|^2$, $\delta = \delta(\varepsilon)$ and $\sigma_0 = \sigma_0(\varepsilon, \sigma)$ applied to the problem (74), if we find a $\left(\frac{\delta}{2}, \sigma_0\right)$-solution $\widetilde{y}_{\delta/2}(x)$ of the problem (74), then $\nabla_x G\left(x, \widetilde{y}_{\delta/2}(x)\right)$ is $(\delta, \sigma_0, 2L_g)$-oracle of g and its calculation requires $\tau_G = 1$ calls of the oracle $\nabla_x G(\cdot, y)$. To finish the proof, we now focus on obtaining a $\left(\frac{\delta}{2}, \sigma_0\right)$-solution $\widetilde{y}_{\delta/2}(x)$ of the problem (74), for which we construct a two-loop procedure described below.

Loop 1

The goal of Loop 1 is to find an $\left(\frac{\delta(\varepsilon)}{2}, \sigma_0(\varepsilon, \sigma)\right)$-solution of problem (74) as a maximization problem in y.

To obtain such an approximate solution, we change the sign of this optimization problem and apply Algorithm 2 with

$$\varphi = -G(x, y), \quad \psi = h(y) + \frac{H}{2}\|y - y_0\|^2. \tag{77}$$

Function φ is convex and has L_G-Lipschitz continuous gradient, function ψ is $H + \mu_y$-strongly convex and has $L_h + H$-Lipschitz continuous gradient. Thus, we can apply Algorithm 2 with exact oracles and parameter $H_1 \geqslant 2L_G$, which will be chosen later, to solve problem (74). To satisfy the conditions of Theorem 4, which gives the complexity of Algorithm 2, we, first, observe that the oracles of φ and ψ are exact and, second, observe that we need in each iteration of Algorithm 1, used as a building block

in Algorithm 2, to find an $\left(\widetilde{\varepsilon}_f^{(1)}\left(\frac{\delta}{2}\right), \widetilde{\sigma}^{(1)}\left(\frac{\delta}{2}, \sigma_0\right)\right)$-solution to the auxiliary problem (15), which in this case has the following form:

$$z_{k+1}^t = \underset{z\in\mathbb{R}^{d_y}}{\operatorname{argmin}}\{\left\langle \nabla\varphi\left(z_k^{md}\right), z - z_k^{md}\right\rangle + \psi(z) + \frac{H_1}{2}\left\|z - z_k^{md}\right\|_2^2\} =$$

$$= \underset{z\in\mathbb{R}^{d_y}}{\operatorname{argmin}}\left\{-\left\langle \nabla_z G\left(x, z_k^{md}\right), z - z_k^{md}\right\rangle + h(z) + \frac{H}{2}\|z - y_0\|^2 + \frac{H_1}{2}\left\|z - z_k^{md}\right\|_2^2\right\}, \quad (78)$$

where $\widetilde{\sigma}^{(1)}\left(\frac{\delta}{2}, \sigma_0\right)$, $\widetilde{\varepsilon}_f^{(1)}\left(\frac{\delta}{2}\right)$ need to satisfy inequalities (34) and (35). Below, in the section "Loop 2" we explain how to solve this auxiliary problem by a variance reduction method in such a way that these inequalities hold.

To summarize Loop 1, both main assumptions of Theorem 4 hold and we can use it to guarantee that we obtain an $\left(\frac{\delta}{2}, \sigma_0\right)$-solution of problem (74). Due to polynomial dependences $\delta(\varepsilon) = \text{poly}(\varepsilon)$, $\sigma_0(\varepsilon, \sigma) = \text{poly}(\varepsilon, \sigma)$ this requires $\widetilde{O}\left(1 + \left(\frac{H_1}{\mu_\varphi + \mu_\psi}\right)^{1/2}\right) = \widetilde{O}\left(1 + \left(\frac{H_1}{\mu_y + H}\right)^{1/2}\right)$ calls to the (exact) oracles for φ and for ψ, and the same number of times solving the auxiliary problem (78). Combining this oracle complexity with the cost of calculating (exact) oracles for φ and for ψ, we find that solving the problem (74) requires $\widetilde{O}\left(1 + \left(\frac{H_1}{\mu_y + H}\right)^{1/2}\right)$ calls of the basic oracle for $G(x, \cdot)$ and $\widetilde{O}\left(m_h + m_h\left(\frac{H_1}{\mu_y + H}\right)^{1/2}\right)$ of the basic oracles for h, i. e. stochastic gradients ∇h_i. The only remaining thing is to provide an inexact solution to problem (78) and, next, we move to Loop 2 to explain how to guarantee this. Note that we need to solve problem (78) $\widetilde{O}\left(1 + \left(\frac{H_1}{\mu_y + H}\right)^{1/2}\right)$ times.

Loop 2

We solve problem (78) by the algorithm L-SVRG proposed in [Morin, Giselsson, 2020], whose complexity is stated in Lemma 13, see Supplementary materials.

As mentioned in the previous Loop 1, in each iteration of Algorithm 2 in Loop 1 we need many times to find an $(\varepsilon_2, \sigma_2)$-solution of the auxiliary problem (78), where for simplicity we denote $\sigma_2 = \widetilde{\sigma}^{(1)}\left(\frac{\delta}{2}, \sigma_0\right)$ and $\varepsilon_2 = \widetilde{\varepsilon}_f^{(1)}\left(\frac{\delta}{2}\right)$.

To obtain such an approximate solution, we apply L-SVRG from [Morin, Giselsson, 2020] with (see Lemma 13 in Supplementary materials)

$$\varphi = \frac{1}{m_h}\sum_{i=1}^{m_h}\underbrace{\left(h_i(z) + \frac{H}{2}\|z - y_0\|^2 + \frac{H_1}{2}\left\|z - z_k^{md}\right\|_2^2\right)}_{\varphi_i(z)}, \quad \psi = -\left\langle \nabla_z G\left(x, z_k^{md}\right), z - z_k^{md}\right\rangle. \quad (79)$$

Functions φ_i are convex and have $L_h^i + H + H_1$-Lipschitz continuous gradient for all $i = 1, \ldots, m_h$, function ψ is convex, 0-smooth and prox-friendly. Also, function φ is $\mu_y + H + H_1$-strongly convex. Thus, all the conditions of Lemma 13 in Supplementary Materials are satisfied and we can apply L-SVRG from [Morin, Giselsson, 2020] to solve problem (78). From this lemma we get an estimate $\widetilde{O}\left(m_h + \sqrt{\frac{m_h(L_h + H + H_1)}{\mu_y + H + H_1}}\right)$ for the number of calls of the basic oracle for h.

To summarize Loop 2, the assumptions of Lemma 13 in Supplementary Materials hold and we can use it to guarantee that we obtain an $(\varepsilon_2, \sigma_2)$-solution of problem (78). According to the polynomial dependences (34) and (35), we find that

$$\sigma_2 = \widetilde{\sigma}^{(1)}\left(\frac{\delta}{2}, \sigma_0\right) = \text{poly}\left(\frac{\delta}{2}, \sigma_0\right), \quad \varepsilon_2 = \widetilde{\varepsilon}_f^{(1)}\left(\frac{\delta}{2}, \sigma_0\right) = \text{poly}\left(\frac{\delta}{2}, \sigma_0\right).$$

Using conditions $\delta(\varepsilon) = \text{poly}(\varepsilon)$, $\sigma_0(\varepsilon, \sigma) = \text{poly}(\varepsilon, \sigma)$ in the formulation of Asumption 2, we find that the dependences

$$\sigma_2(\varepsilon, \sigma),\ \widetilde{\sigma}^{(1)}(\varepsilon, \sigma),\ \varepsilon_2(\varepsilon, \sigma),\ \widetilde{\varepsilon}_f^{(1)}(\varepsilon, \sigma)$$

are polynomial. Then, we can use notation $\widetilde{O}(\cdot)$ without specifying what precision we mean and implying that the logarithmic part depends on the initial ε, σ. Finally, according to Lemma 13 in Supplementary materials an $(\varepsilon_2, \sigma_2)$-solution of problem (78) requires $\widetilde{O}\left(m_h + \sqrt{\frac{m_h(L_h+H+H_1)}{\mu_y+H+H_1}}\right)$ calls of the basic oracle for h, i. e. stochastic gradients ∇h_i, and the same number of times solving the auxiliary problem of the form $\operatorname{argmin}_y\{\psi(y) + \frac{1}{2\alpha}\|y - \overline{y}\|_2^2\}$. This problem is solved explicitly since $\psi(y)$ is a linear function.

Combining the estimates of both loops

Combining the estimates of the above section "Loop 1" and paragraph "Loop 2" we see that, finding a point $\widetilde{y}_{\delta/2}(x)$ which is an $\left(\frac{\delta(\varepsilon)}{2}, \sigma_0(\varepsilon, \sigma)\right)$-solution to the problem (74) requires the following number of calls of the basic oracles of $G(x, \cdot)$ and h, respectively:

$$\widetilde{O}\left(1 + \sqrt{\frac{H_1}{H+\mu_y}}\right), \tag{80}$$

$$\text{\# of calls in Loop1} + (\text{\# of steps in Loop 1})\cdot(\text{\# of calls in Loop 2}) = \tag{81}$$

$$= \widetilde{O}\left(m_h + m_h\sqrt{\frac{H_1}{H+\mu_y}} + \left(1 + \sqrt{\frac{H_1}{H+\mu_y}}\right)\left(m_h + \sqrt{\frac{m_h(L_h+H+H_1)}{\mu_y+H+H_1}}\right)\right).$$

Finding $(\delta(\varepsilon), \sigma_0(\varepsilon, \sigma), 2L_g)$-oracle of g by calculating $\nabla_x G\left(x, \widetilde{y}_{\delta/2}(x)\right)$ requires additionally $\tau_G = 1$ calls of the basic oracle for $G(\cdot, y)$. Since in Condition 2 we denote the dependence on the target accuracy ε and confidence level σ by a separate quantities denoted by $\mathcal{K}(\varepsilon, \sigma)$ and in this case it is logarithmic, choosing $H_1 = 2L_G$ we get the final estimates for $\mathcal{N}_G^y$ and $\mathcal{N}_h$ to guarantee that Condition 2 holds:

$$\mathcal{N}_G^y = O\left(1 + \sqrt{\frac{L_G}{H+\mu_y}}\right), \tag{82}$$

$$\mathcal{N}_h = O\left(m_h + \left(1 + \sqrt{\frac{2L_G}{H+\mu_y}}\right)\left(m_h + \sqrt{\frac{m_h(L_h+H+2L_G)}{\mu_y+H+2L_G}}\right)\right) =$$

$$= O\left(m_h + \left(1 + \sqrt{\frac{2L_G}{H+\mu_y}}\right)\left(m_h + \sqrt{\frac{m_hL_h}{\mu_y+H+2L_G}} + \sqrt{\frac{m_h(H+2L_G)}{\mu_y+H+2L_G}}\right)\right) =$$

$$= O\left(m_h + \left(1 + \sqrt{\frac{2L_G}{H+\mu_y}}\right)\left(m_h + \sqrt{\frac{m_hL_h}{\mu_y+H+2L_G}}\right)\right) =$$

$$= O\left(m_h + \sqrt{\frac{2L_G}{H+\mu_y}}\sqrt{\frac{m_hL_h}{2L_G}}\right) = O\left(m_h + \sqrt{\frac{m_hL_h}{H+\mu_y}}\right) = O\left(\sqrt{\frac{m_hL_h}{H+\mu_y}}\right), \tag{83}$$

where we used that, by the assumptions of this lemma, $1 \leqslant \frac{4L_G}{H+\mu_y}$, $m_h(H + 2L_G + \mu_y) \leqslant L_h$ and $\forall a,\ b \geqslant 0$ $\sqrt{a+b} \leqslant \sqrt{a} + \sqrt{b}$. □

By changing the variables x and y in Lemma 4 and choosing $H = 0$ we obtain the simple Corollary 2 which ensures Assumption 3.

Corollary 2. *Let the function r be defined via the maximization problem in* (50)*, i. e.*

$$r(y) = \min_{x \in \mathbb{R}^{d_x}} \{G(x, y) + f(x)\}, \tag{84}$$

where $G(x, y)$, $f(y)$ are according to (71) *and satisfy Condition* 4*. Assume also that $2L_G + \mu_x \leqslant L_f$ and $\mu_x \leqslant 4L_G$. Then, organizing computations in two loops and applying Algorithm* 2 *in the outer loop and the accelerated variance reduction method L-SVRG from* [Morin, Giselsson, 2020] *in the inner loop, we guarantee Condition* 3 *with $\tau_G = 1$ basic oracle calls for $G(x, \cdot)$ and the following estimates for the number of basic oracle calls for $G(\cdot, y)$, f, respectively*:

$$\mathcal{N}_G^x(\tau_G) = O\left(1 + \sqrt{\frac{L_G}{\mu_x}}\right), \tag{85}$$

$$\mathcal{N}_f(\tau_f) = O\left(\sqrt{\frac{L_f}{\mu_x}}\right). \tag{86}$$

Final estimates

We are now in a position to state the final result of this section for the complexity estimates when solving problem (71). Assumption 4 with (73) guarantee that Condition 1 holds. Lemma 4 and Corollary 2 guarantee that Assumptions 2, 3 hold.

Thus, all the conditions of Theorem 5 are satisfied and we obtain the following result for solving the problem (71) with our system of inner-outer loops.

Theorem 6. *Assume that for the problem* (71) *Assumption* 4 *holds and additionally $m_h(4L_G + \mu_y) \leqslant L_h$, $2L_G + \mu_x \leqslant L_f$, $\mu_y \leqslant L_G$, $\mu_x \leqslant L_G$.*

Then the general framework, described in Section "Accelerated Framework for Saddle-Point Problems", combined with the algorithms described in the previous subsection, find an (ε, σ)-solution to problem (71) *with the following number of basic oracle calls*

$$\nabla f\text{-oracle calls}: \quad \widetilde{O}\left(\sqrt{\frac{L_G L_f}{\mu_x \mu_y}}\right), \tag{87}$$

$$\nabla h_i\text{-oracle calls}: \quad \widetilde{O}\left(\sqrt{\frac{m_h L_G L_h}{\mu_x \mu_y}}\right), \tag{88}$$

$$\nabla_x G\text{-oracle calls}: \quad \widetilde{O}\left(\sqrt{\frac{L_G^2}{\mu_x \mu_y}}\right), \tag{89}$$

$$\nabla_y G\text{-oracle calls}: \quad \widetilde{O}\left(\sqrt{\frac{L_G^2}{\mu_x \mu_y}}\right). \tag{90}$$

An important particular case for which we state the following corollary is when it does not have the finite-sum, i. e. $m_h = 1$.

Corollary 3 (particular case $m_h = 1$). *Let the assumptions of Theorem* 6 *hold and additionally $m_h = 1$. Then the general framework described in Section "Accelerated Framework for*

Saddle-Point Problems", combined with the algorithms described in the previous subsection, find an (ε, σ)*-solution to problem* (71) *with the following number of basic oracle calls*:

$$\nabla f\text{-oracle calls}: \quad \widetilde{O}\left(\sqrt{\frac{L_G L_f}{\mu_x \mu_y}}\right), \tag{91}$$

$$\nabla h\text{-oracle calls}: \quad \widetilde{O}\left(\sqrt{\frac{L_G L_h}{\mu_x \mu_y}}\right), \tag{92}$$

$$\nabla_x G\text{-oracle calls}: \quad \widetilde{O}\left(\sqrt{\frac{L_G^2}{\mu_x \mu_y}}\right), \tag{93}$$

$$\nabla_y G\text{-oracle calls}: \quad \widetilde{O}\left(\sqrt{\frac{L_G^2}{\mu_x \mu_y}}\right). \tag{94}$$

Accelerated Methods for Saddle-Point Problems with Finite-Sum Structure

In this section, we consider problem (2), which is problem (37) with a specific finite-sum structure of the function G. The algorithms in this section are, in fact, deterministic, i. e. correspond to confidence levels $\sigma = 0$. Thus, our goal is to obtain an ε-solution to problem (2). As in the previous section, we use the general framework described in Section "Accelerated Framework for Saddle-Point Problems", but in a simpler setting of all the confidence levels σ being equal to zero. To obtain the final estimates for the number of basic oracles calls, we need to satisfy Assumptions 1, 2, 3 which are formulated in Section "Accelerated Framework for Saddle-Point Problems", where we construct our general framework. The proof that these assumptions hold and the proof of the resulting complexity bounds follow mostly the same lines as for the case of problem (71) under Assumption 4 in the previous section, but are rather technical. Thus, in this section we only state the main results and the proofs are deferred to the Appendix.

Problem statement

In this section we consider the optimization problem of the form (2):

$$\min_{x \in \mathbb{R}^{d_x}} \max_{y \in \mathbb{R}^{d_y}} \{f(x) + G(x, y) - h(y)\}, \quad G(x, y) := \frac{1}{m_G} \sum_{i=1}^{m_G} G_i(x, y), \tag{95}$$

and develop accelerated optimization methods for its solution under the following assumptions.

Condition 5.

1. *Function* $f(x)$ *is* μ_x*-strongly convex, and function* $h(y)$ *is* μ_y*-strongly convex.*
2. $m_G \geqslant 1$ *and each function* $G_i(x, y)$, $i \in 1, \ldots, m_G$ *is convex in* x *and concave in* y, *and* L_G^i*-smooth, i. e. for each* $x = (x_1, x_2)$, $y = (y_1, y_2) \in \mathbb{R}^{d_x} \times \mathbb{R}^{d_y}$

$$\|\nabla G_i(x_1, x_2) - \nabla G_i(y_1, y_2)\| \leqslant L_G^i \|(x_1, x_2) - (y_1, y_2)\|. \tag{96}$$

We also define $L_G = \frac{1}{m_G} \sum\limits_{i=1}^{m_G} L_G^i$.

3. *One of the following statements holds for the functions* $f(x)$, $h(y)$:
   (a) *Function* $f(x)$ *is* L_f*-smooth and function* $h(y)$ *is* L_h*-smooth*;
   (b) *Function* $f(x)$ *is* L_f*-smooth, function* $h(y)$ *is* L_h*-smooth and prox-friendly.*

Under Condition 5.2 it is easy to see that the function $G(x, y)$ in problem (95) is L_G-smooth. Indeed,

$$\|\nabla G(x_1,\, x_2) - \nabla G(y_1,\, y_2)\| \leqslant \frac{1}{m_G}\sum_{i=1}^{m_G}\|\nabla G_i(x_1,\, x_2) - \nabla G_i(y_1,\, y_2)\| \leqslant$$

$$\leqslant \frac{1}{m_G}\sum_{i=1}^{m_G} L_G^i\|(x_1,\, x_2) - (y_1,\, y_2)\| = L_G\|(x_1,\, x_2) - (y_1,\, y_2)\|,$$

where $x = (x_1,\, x_2)$, $y = (y_1,\, y_2) \in \mathbb{R}^{d_x} \times \mathbb{R}^{d_y}$.

To further fit Condition 1, we consider the full gradient oracles $\nabla h(y)$, $\nabla f(x)$ as the basic oracles O_h, O_f, respectively, and the stochastic gradient oracle $\nabla_x G_i(x, y)$, $\nabla_y G_i(x, y)$ as the basic oracles O_G^x, O_G^y, respectively. Then Condition 5 guarantees that Condition 1 holds with

$$\tau_f = \tau_h = 1, \quad \tau_G = m_G. \tag{97}$$

Complexity estimates

In this section we consider problem (95) under one of the two different Assumptions 5.3(a) or (b) and mostly follow the lines of derivations described in Section "Accelerated Method for Saddle-Point Problems" with appropriate changes caused by the different problem statement. In particular, we change the order of the loops in the general framework described in Section "Accelerated Framework for Saddle-Point Problems" as well as in the proof of Lemma 4 and Corollary 2 depending on which is larger: L_h or L_G and L_f or L_G. This eventually allows us to avoid assumptions of the form $4L_G + \mu_y \leqslant L_h$, $2L_G + \mu_x \leqslant L_f$, which are used in Theorem 6. The proof of the resulting complexity bounds follows mostly the same ideas as for the case of problem (71) under Assumption 4, but is rather technical. Thus, in this section we only state the result and the proofs are deferred to the appendices. In Supplementary Materials we propose a variation of the general framework described in Section "Accelerated Framework for Saddle-Point Problems", but with the change of the order of Loop 2 and Loop 3. As a result, we prove Theorem 10 which is a counterpart of Theorem 5. In Supplementary Materials we prove Lemma 14 and Corollary 4, which generalize Lemma 4 and Corollary 2 in two aspects. First, we consider the function G given in (95). Second, we do not use the assumption $m_h(H + 2L_G + \mu_y) \leqslant L_h$ of Lemma 4 and $2L_G + \mu_x \leqslant L_f$ of Corollary 2.

We start by considering problem (95) under Assumption 5.1,2,3(a). This assumption, combined with (97), guarantees that Condition 1 holds. Lemma 14 and Corollary 4 in Supplementary Materials guarantee that Assumptions 2, 3 hold. This allows us to combine Lemma 14 and Corollary 4 with either Theorem 5 if $L_f \geqslant L_G$, or Theorem 10 in Supplementary Materials if $L_f \leqslant L_G$. The resulting complexity estimates for solving problem (95) with our system of inner-outer loops are given in the next theorem which is proved in Supplementary Materials. Notice that in this case the algorithm is fully deterministic and we find an ε-solution to problem (95).

Theorem 7. *Assume that for problem* (95) *Assumption* 5.1,2,3(a) *holds and additionally* $\mu_x \leqslant L_G$, $\mu_x \leqslant L_f$ *and* $\mu_y \leqslant L_G$. *Then using the general framework from Section "Accelerated Framework for Saddle-Point Problems", the general framework in Supplementary Materials, Lemma* 14 *and Corollary* 4 *for each relation between* L_h, L_G *and* L_f, L_G, *respectively, we provide an algorithm which finds an* ε*-solution to problem* (95) *with the following number of basic*

oracle calls:

$$\nabla f\text{-oracle calls}: \quad \widetilde{O}\left(\sqrt{\frac{L_G L_f}{\mu_x \mu_y}}\right), \tag{98}$$

$$\nabla h\text{-oracle calls}: \quad \widetilde{O}\left(\max\left\{\sqrt{\frac{L_G L_h}{\mu_x \mu_y}},\ \sqrt{\frac{L_G^2}{\mu_x \mu_y}}\right\}\right), \tag{99}$$

$$\nabla_x G_i\text{-oracle calls}: \quad \widetilde{O}\left(m_G \sqrt{\frac{L_G^2}{\mu_x \mu_y}}\right), \tag{100}$$

$$\nabla_y G_i\text{-oracle calls}: \quad \widetilde{O}\left(m_G \sqrt{\frac{L_G^2}{\mu_x \mu_y}}\right). \tag{101}$$

We prove this theorem in the Appendix.

We would like to emphasize that, even though we do not use variance reduction techniques in the algorithm described in Theorem 7, under assumption 5.1,2,3(a) our bounds are better than the bounds obtained by the variance reduction method proposed in [Palaniappan, Bach, 2016]. To solve the problem (95) by the algorithm of [Palaniappan, Bach, 2016], we need to restate this problem as

$$\min_{x \in \mathbb{R}^{d_x}} \max_{y \in \mathbb{R}^{d_y}} \left\{ \frac{1}{m_G} \sum_{i=1}^{m_G} \left(\widetilde{G}_i(x, y) := f(x) + G_i(x, y) - h(y) \right) \right\}$$

with the objective being $L_{\widetilde{G}} = \max\{L_G + L_f, L_G + L_h\}$-smooth. The algorithm in [Palaniappan, Bach, 2016] does not propose a way to separate the complexities for different parts of the objective and the resulting number of oracle calls for each part is the same

$$\nabla f,\ \nabla h,\ \nabla_x G_i,\ \nabla_y G_i\text{-oracle calls}: \widetilde{O}\left(\sqrt{m_G} \frac{L_{\widetilde{G}}}{\min\{\mu_x, \mu_y\}}\right). \tag{102}$$

Comparing these estimates with the estimates of Theorem 7, we make two important observations.

- Due to our approach with complexity separation the estimates from Theorem 7 on the number of oracle calls for f and h are always better than the corresponding estimates in (102) at least by a factor $\sqrt{m_G}$.
- At first sight, the estimates on the number of calls of $\nabla_x G_i$ and $\nabla_y G_i$ from Theorem 7 seem worse than the corresponding estimates in (102) due to the additional factor $\sqrt{m_G}$. However, this is not the case, for example, when L_f or L_h are large enough, leading to $L_{\widetilde{G}} \gg L_G$. This can be demonstrated by taking $m_G L_G \leqslant L_f$, then the estimates on the number of calls of $\nabla_x G_i$ and $\nabla_y G_i$ in Theorem 7 become $\sqrt{\frac{L_f^2}{\mu_x \mu_y}}$, which is smaller than the estimates in (102).

An interesting open question is whether we can improve the complexity bounds in Theorem 7 by applying variance reduction methods to ensure Assumptions 2 and 3. We conjecture that it is possible to improve the bounds (100) and (101) to $\widetilde{O}\left(\sqrt{\frac{m_G L_G^2}{\mu_x \mu_y}}\right)$.

As a particular case of problem (95) we can consider problem (71) with $m_h = 1$. This allows us to relax the assumptions $m_h(4L_G + \mu_y) \leqslant L_h$, $2L_G + \mu_x \leqslant L_f$, $\mu_y \leqslant L_G$ made in Corollary 3 and obtain the following corollary of the previous theorem. Notice that again in this case the algorithm is fully deterministic and we find an ε-solution to problem (71).

Corollary 4. *Assume that for problem* (71) *Assumption* 4 *holds and additionally* $m_h = 1$, $\mu_x \leqslant L_G$, $\mu_x \leqslant L_f$ *and* $\mu_y \leqslant L_G$. *Then, using the general framework from Section "Accelerated Framework for Saddle-Point Problems", the general framework in Supplementary Materials and Lemma* 14 *with Corollary* 4 *for each relation between* L_h, L_G *and* L_f, L_G, *respectively, we provide an algorithm which finds an* ε*-solution to problem* (71) *with the following number of basic oracle calls*:

$$\nabla f\text{-oracle calls}: \quad \widetilde{O}\left(\sqrt{\frac{L_G L_f}{\mu_x \mu_y}}\right), \tag{103}$$

$$\nabla h\text{-oracle calls}: \quad \widetilde{O}\left(\max\left\{\sqrt{\frac{L_G L_h}{\mu_x \mu_y}}, \sqrt{\frac{L_G^2}{\mu_x \mu_y}}\right\}\right), \tag{104}$$

$$\nabla_x G\text{-oracle calls}: \quad \widetilde{O}\left(\sqrt{\frac{L_G^2}{\mu_x \mu_y}}\right), \tag{105}$$

$$\nabla_y G\text{-oracle calls}: \quad \widetilde{O}\left(\sqrt{\frac{L_G^2}{\mu_x \mu_y}}\right). \tag{106}$$

We now turn to the problem (95) under Assumption 5.1,2,3(b). This assumption, combined with (97), guarantees that Condition 1 holds. Part 3(b) allows a simple construction, which is given in the proof of Lemma 15 in Supplementary Materials, to guarantee Assumption 2. The main difference from Lemma 14 is that due to the prox-friendliness of h the second loop is not needed and it is sufficient to apply just Algorithm 2 to solve problem (49) in Assumption 2. Corollary 4 in Supplementary Materials guarantees that Assumption 3 holds. This allows us to combine Lemma 14 and Corollary 4 with either Theorem 5 if $L_f \geqslant L_G$, or Theorem 10 in Supplementary Materials if $L_f \leqslant L_G$. The resulting complexity estimates for solving problem (95) with our system of inner-outer loops are given in the next theorem which is proved in Supplementary Materials. Notice that in this case the algorithm is fully deterministic and we find an ε-solution to problem (95).

Theorem 8. *Assume that for problem* (95) *Assumption* 5.1,2,3(b) *holds and additionally* $\mu_x \leqslant L_G$, $\mu_x \leqslant L_f$ *and* $\mu_y \leqslant L_G$. *Then, using the general framework from Section "Accelerated Framework for Saddle-Point Problems", the general framework in Supplementary Materials and Lemma* 15 *with Corollary* 4 *for each relation between* L_h, L_G *and* L_f, L_G, *respectively, we provide an algorithm which finds an* ε*-solution to problem* (95) *with the following number of basic oracle calls*:

$$\nabla f\text{-oracle calls}: \quad \widetilde{O}\left(\sqrt{\frac{L_G L_f}{\mu_x \mu_y}}\right), \tag{107}$$

$$\nabla h\text{-oracle calls}: \quad \widetilde{O}\left(\sqrt{\frac{L_G}{\mu_y}}\right), \tag{108}$$

$$\nabla_x G_i\text{-oracle calls}: \quad \widetilde{O}\left(m_G \sqrt{\frac{L_G^2}{\mu_x \mu_y}}\right), \tag{109}$$

$$\nabla_y G_i\text{-oracle calls}: \quad \widetilde{O}\left(m_G \sqrt{\frac{L_G^2}{\mu_x \mu_y}}\right). \tag{110}$$

We prove this theorem in Supplementary Materials.

REMARK 2. In this remark, using the results from [Song, Wright, Diakonikolas, 2021], we show how we can utilize our approach to solve the problems of structured nonsmooth convex finite-sum optimization that appears widely in machine learning applications, including support vector machines and least absolute deviation.

We consider large-scale regularized nonsmooth convex empirical risk minimization (ERM) of linear predictors in machine learning. Let $b_i \in \mathbb{R}^n_x$, $i = 1, 2, \ldots, n$, be sample vectors with n typically large; $f_i \colon \mathbb{R} \to \mathbb{R}$, $i = 1, 2, \ldots, n$, be possibly nonsmooth convex loss functions associated with the linear predictor $\langle b_i, x\rangle$. The problem we study is

$$\min_{x\in\mathbb{R}^{d_x}} \max_{y\in\mathbb{R}^{d_y}} \left\{ \frac{1}{n}\sum_{i=1}^{n} f_i(\langle b_i, x\rangle + G(x, y) - h(y) \right\}. \tag{111}$$

We require that the convex conjugates of the functions f_i, defined by $f_i^*(z_i) := \max\limits_{\xi_i}(\xi_i z_i - f_i(\xi_i))$, admit efficiently computable proximal operators. Thus, we can rewrite the function $\frac{1}{n}\sum\limits_{i=1}^{n} f_i(\langle b_i, x\rangle)$ in the following way:

$$\frac{1}{n}\sum_{i=1}^{n} f_i(\langle b_i, x\rangle) = \frac{1}{n}\sum_{i=1}^{n} \max_{z_i} \left(z_i\langle b_i, x\rangle - f_i^*(z_i)\right) = \max_{z\in\mathbb{R}^n} \left\{ \langle z, Bx\rangle - \frac{1}{n}\sum_{i=1}^{n} f_i^*(z_i) \right\}, \tag{112}$$

where $y = (y_1, \ldots, y_n)$, $B = \frac{1}{n}[b_1, \ldots, b_n]^T$. Then by substitution of the equation (112) into the problem (111), we obtain

$$\min_{x\in\mathbb{R}^{d_x}} \left\{ \max_{y\in\mathbb{R}^{d_y}} \{G(x, y) - h(y)\} + \max_{z\in\mathbb{R}^n} \left\{ \langle z, Bx\rangle - \frac{1}{n}\sum_{i=1}^{n} f_i^*(z_i) \right\} \right\}. \tag{113}$$

We can use another notation $\eta = (y, z)$ and rewrite the problem (113) as follows:

$$\min_{x\in\mathbb{R}^{d_x}} \left\{ \max_{\eta=(y,z)\in\mathbb{R}^{d_y+n}} \left\{ G(x, y) - h(y) + \langle z, Bx\rangle - \frac{1}{n}\sum_{i=1}^{n} f_i^*(z_i) \right\} \right\}, \tag{114}$$

which we can solve using the general framework from Section "Accelerated Framework for Saddle-Point Problems" under the differences assumptions. It is worth mentioning that the function $f^*(z) = \frac{1}{n}\sum\limits_{i=1}^{n} f_i^*(z_i)$ is separable and admits an efficiently computable proximal operator. This the primal-dual problem (112) has significantly lower complexity than the saddle-point problem (111). That means we can use the primal-dual approach with no care that the saddle-problem (114) becomes more complex.

References

Alacaoglu A., Malitsky Y. Stochastic Variance Reduction for Variational Inequality Methods // Conference on Learning Theory. — 2021. — Vol. 178. — P. 778–816.

Alkousa M., Dvinskikh D., Stonyakin F., Gasnikov A., Kovalev D. Accelerated methods for composite non-bilinear saddle point problem // arxiv.org. — 2019. — https://arxiv.org/abs/1906.03620 (date of access: 09.06.2019).

Alkousa M., Gasnikov A., Dvinskikh D., Kovalev D., Stonyakin F. Accelerated Methods for Saddle-Point Problem // Computational Mathematics and Mathematical Physics. — 2020. — Vol. 60, No. 11. — P. 1787–1809.

Bubeck S., Jiang Q., Lee Y., Li Y., Sidford A. Near-optimal method for highly smooth convex optimization // Conference on Learning Theory. — 2019. — P. 492–507.

Carmon Y., Jin Y., Sidford A., Tian K. Variance reduction for matrix games // Advances in Neural Information Processing Systems. — 2019. — P. 11381–11392.

Chambolle A., Pock T. A First-Order Primal-Dual Algorithm for Convex Problems with Applications to Imaging // Journal of Mathematical Imaging and Vision. — 2011. — Vol. 40, No. 1. — P. 120–145.

Chen Y., Lan G., Ouyang Y. Accelerated schemes for a class of variational inequalities // Mathematical Programming. — 2017. — Vol. 165, No. 1. — P. 113–149.

d'Aspremont A., Scieur D., Taylor A. Acceleration Methods // arxiv.org. — 2021. — https://arxiv.org/abs/2101.09545 (date of access: 23.01.2021).

Devolder O. Exactness, inexactness and stochasticity in first-order methods for large-scale convex optimization. — PhD thesis. — ICTEAM and CORE, Université Catholique de Louvain, 2013.

Devolder O., Glineur F., Nesterov Y. First-order methods of smooth convex optimization with inexact oracle // Mathematical Programming. — 2014. — Vol. 146, No. 1. — P. 37–75.

Dvinskikh D., Kamzolov D., Gasnikov A., Dvurechensky P., Pasechnyk D., Matykhin V., Chernov A. Accelerated meta-algorithm for convex optimization problems // Computational Mathematics and Mathematical Physics. — 2021. — Vol. 61, No. 1. — P. 17–28.

Dvurechensky P., Nesterov Y., Spokoiny V. Primal-Dual Methods for Solving Infinite-Dimensional Games // Journal of Optimization Theory and Applications. — 2015. — Vol. 166, No. 1. — P. 23–51.

Gasnikov A. Searching equillibriums in large transport networks // arxiv.org. — 2016. — https://arxiv.org/abs/1607.03142 (date of access: 11.07.2016).

Gasnikov A., Dvurechensky P., Nesterov Y. Stochastic gradient methods with inexact oracle // Proceedings of Moscow Institute of Physics and Technology. — 2016. — Vol. 8, No. 1. — P. 41–91.

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. — https://arxiv.org/abs/2010.02280 (date of access: 05.10.2020).

Gladin E., Sadiev A., Gasnikov A., Dvurechensky P., Beznosikov A., Alkousa M. Solving smooth min-min and min-max problems by mixed oracle algorithms // Communications in Computer and Information Science. — 2021. — Vol. 1476. — P. 19–40.

Grapiglia G., Nesterov Y. On inexact solution of auxiliary problems in tensor methods for convex optimization // Optimization Methods and Software. — Taylor & Francis, 2020. — P. 1–26.

Han Y., Xie G., Zhang Z. Lower Complexity Bounds of Finite-Sum Optimization Problems: The Results and Construction // arxiv.org. — 2021. — https://arxiv.org/abs/2103.08280 (date of access: 15.03.2021).

Hien L., Zhao R., Haskell W. An Inexact Primal-Dual Smoothing Framework for Large-Scale Non-Bilinear Saddle Point Problems // arxiv.org. — 2020. — https://arxiv.org/abs/1711.03669 (date of access: 10.11.2017).

Ibrahim A., Azizian W., Gidel G., Mitliagkas I. Linear lower bounds and conditioning of differentiable games // International Conference on Machine Learning. — 2020. — P. 4583–4593.

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

Lan G. Lectures on optimization. Methods for Machine Learning. — H. Milton Stewart School of Industrial and Systems Engineering, 2019.

Lan G. First-order and Stochastic Optimization Methods for Machine Learning. — Springer, 2020.

Lin T., Jin C., Jordan M. Near-Optimal Algorithms for Minimax Optimization booktitle // Proceedings of Thirty Third Conference on Learning Theory. — 2020. — Vol. 125. — P. 2738–2779.

Lin H., Mairal J., Harchaoui Z. A universal catalyst for first-order optimization // Conference Neural Information Processing Systems (NIPS). — 2015.

Monteiro R., Svaiter B. An Accelerated Hybrid Proximal Extragradient Method for Convex Optimization and Its Implications to Second-Order Methods // SIAM Journal on Optimization. — 2013. — Vol. 23, No. 2. — P. 1092–1125.

Morgenstern O., Von Neumann J. Theory of games and economic behavior. — Princeton university press, 1953.

Moreau J. Proximité et dualité dans un espace hilbertien // Bulletin de la Société Mathématique de France. — Société mathématique de France, 1965. — Vol. 93. — P. 273–299.

Morin M., Giselsson P. Sampling and Update Frequencies in Proximal Variance Reduced Stochastic Gradient Methods // arxiv.org. — 2020. — https://arxiv.org/abs/2002.05545 (date of access: 13.02.2020).

Nash Jr., John F. The bargaining problem // Econometrica: Journal of the econometric society. — 1950. — P. 155–162.

Nemirovski A. Prox-method with rate of convergence $O\left(\frac{1}{t}\right)$ 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.

Nemirovsky A., Yudin D. Problem Complexity and Method Efficiency in Optimization. — New York: J. Wiley & Sons, 1983.

Nesterov Y. Dual extrapolation and its applications to solving variational inequalities and related problems // Mathematical Programming. — Springer, 2007. — Vol. 109, No. 2–3. — P. 319–344.

Nesterov Y. Excessive gap technique in nonsmooth convex minimization // SIAM Journal on Optimization. — 2005a. — Vol. 16, No. 1. — P. 235–249.

Nesterov Y. Gradient methods for minimizing composite functions // Mathematical Programming. — Springer, 2013. — Vol. 140, No. 1. — P. 125–161.

Nesterov Y. Smooth minimization of non-smooth functions // Mathematical Programming. — 2005b. — Vol. 103, No. 1. — P. 127–152.

Nesterov Y., Scrimali L. Solving strongly monotone variational and quasi-variational inequalities // Discrete & Continuous Dynamical Systems. — 2011. — Vol. 31, No. 1078-0947-2011-4-1383. — P. 1383.

Ostrovskii D., Lowy A., Razaviyayn M. Efficient search of first-order nash equilibria in nonconvex-concave smooth min-max problems // SIAM Journal on Optimization. — 2021. — Vol. 31, No. 4. — P. 2508–2538.

Palaniappan B., Bach F. Stochastic Variance Reduction Methods for Saddle-Point Problems // Advances in Neural Information Processing Systems. — 2016. — Vol. 29.

Shalev-Shwartz S., Zhang T. Accelerated Proximal Stochastic Dual Coordinate Ascent for Regularized Loss Minimization // Proceedings of the 31st International Conference on Machine Learning. — 2014. — Vol. 32. — P. 64–72.

Song C., Wright S., Diakonikolas J. Variance Reduction via Primal-Dual Accelerated Dual Averaging for Nonsmooth Convex Finite-Sums // Conference on Machine Learning. — 2021. — Vol. 139. — P. 9824–9834.

Thekumparampil K., Jain P., Netrapalli P., Oh S. Efficient Algorithms for Smooth Minimax Optimization // Advances in Neural Information Processing Systems. — 2019. — Vol. 32.

Wang Y., Li J. Improved Algorithms for Convex-Concave Minimax Optimization // Conference on Neural Information Processing Systems (NeurIPS). — 2020. — No. 403. — P. 4800–4810.

Xie G., Han Y., Zhang Z. DIPPA: An improved Method for Bilinear Saddle Point Problems // arxiv.org. — 2021. — https://arxiv.org/abs/2103.08270 (date of access: 15.03.2021).

Xu Z., Zhang H., Xu Y., Lan G. A unified single-loop alternating gradient projection algorithm for nonconvex-concave and convex-nonconcave minimax problems // Mathematical Programming. — 2023.

Yang J., Zhang S., Kiyavash N., He N. A Catalyst Framework for Minimax Optimization // Advances in Neural Information Processing Systems. — 2020. — Vol. 33.

Zhu Y., Liu D., Tran-Dinh Q. A New Primal-Dual Algorithm for a Class of Nonlinear Compositional Convex Optimization Problems // arxiv.org. — 2020. — https://arxiv.org/abs/2006.09263v2 (date of access: 16.06.2020).