ADAPTIVE ALGORITHMS FOR RELATIVELY LIPSCHITZ CONTINUOUS CONVEX OPTIMIZATION PROBLEMS

FEDOR STONYAKIN, MOHAMMAD ALKOUSA, ALEXANDER TITOV, OLEG SAVCHUK, AND ALEXANDER GASNIKOV

ABSTRACT. Recently, there were proposed some innovative convex optimization concepts, namely, relative smoothness [3] and relative strong convexity [6, 8]. These approaches have significantly expanded the class of applicability of gradient-type methods with optimal estimates of the convergence rate. Later Yu. Nesterov and H. Lu [6, 8] introduced some modifications of the Mirror Descent method for convex minimization problems with the corresponding analogue of the Lipschitz condition (the so-called relative continuity or Lipschitz continuity). In this paper, we cover both the concept of relative smoothness and relative Lipschitz continuity and introduce some adaptive and universal methods which have optimal estimates of the convergence rate for the corresponding class of problems. We consider the relative boundedness condition for the variational inequality problem and propose some adaptive optimal methods for this class of problems. Some results of the conducted numerical experiments are presented, which demonstrate the effectiveness of the proposed methods.

1. INTRODUCTION

The recent dramatic growth of various branches of science has led to the necessity of the development of numerical optimization methods in spaces of large and extra-large dimensions. A special place in modern optimization theory is given to gradient methods. Recently, there was introduced a new direction for the research, associated with the development of gradient-type methods for optimization problems with relatively smooth [3] and relatively strongly convex [6] functions. Such methods are in high demand and urgent due to numerous theoretical and applied problems. For example, the D-optimal design problem turned out to be relatively smooth [6]. It is also quite interesting that in recent years there have appeared applications of these approaches (conditions of relative smoothness and strong convexity) to auxiliary problems for tensor methods for convex minimization problems of the second and higher orders [9, 10]. It is worth noting that tensor methods make it possible to

obtain optimal estimates of the rate of convergence of high-order methods for convex optimization problems [7].

A few years ago there was introduced a generalization of the Lipschitz condition for nonsmooth problems, namely, relative Lipschitz continuity [8,11]. The concept of relative Lipschitz continuity essentially generalizes the classical Lipschitz condition and covers quite important applied problems, including the problem of finding the common point of ellipsoids (IEP), as well as the support vector machine (SVM) for the binary classification problem.

The concepts of relative smoothness, relative Lipschitz continuity, and relative strong convexity made it possible to significantly expand the limits of applicability of gradient type methods while preserving the optimal convergence rate $O(\frac{1}{\varepsilon^2})$ for relatively Lipschitz problems and $O(\frac{1}{\varepsilon})$ for relatively smooth problems (ε, as usual, denotes the accuracy of the solution for functional residual). The authors of [5] have shown that for the class of relatively smooth problems, such an estimate for the rate of convergence cannot be improved in the general case.

In this paper we consider the class of (α, L, δ)-relatively smooth objective functions (see Definition 2.1), which covers both the concept of relative smoothness and relative Lipschitz continuity. Let Q be a closed convex subset of some finite-dimensional vector space. For the classical optimization problem

$$(1.1) \qquad \min_{x \in Q} f(x)$$

we propose some analogues of the universal gradient method which automatically adjusts to the "degree of relative smoothness" of the (α, L, δ)-relatively smooth problem (Sect. 5). We also mention that the proposed algorithms are applicable to solve the problem of minimizing the relatively strongly convex functions, see [15] for more details.

In addition to the classical optimization problem, we consider the problem of solving Minty variational inequality with (M-)relatively bounded operator. For a given relatively bounded and monotone operator $g : Q \longrightarrow \mathbb{R}^n$, we need to find a vector $x_* \in Q$, such that

$$(1.2) \qquad \langle g(x), x_* - x \rangle \leqslant 0 \quad \forall x \in Q.$$

Relative boundedness can be understood as an analogue of relative Lipschitz continuity for variational inequalities. It should be noted that the subgradient of a relatively Lipschitz continuous function satisfies the relative boundedness condition. This fact plays an important role in considering relatively Lipschitz continuous Lagrange saddle point problems and their reduction to corresponding variational inequalities with the relatively bounded operator. Recently, in [12] the authors proposed an adaptive version of the Mirror Prox method (extragradient type method) for variational inequalities with a condition similar to relative smoothness. It should be noted that variational inequalities with relatively smooth operators are applicable to the resource sharing problem [2]. Also, in [16] there were introduced some non-adaptive switching subgradient algorithms for convex programming problems with relatively Lipschitz continuous functions. Recently, there was proposed a non-adaptive method for solving variational inequalities with the relatively bounded

operator [17]. In this paper, we propose an adaptive algorithm for the corresponding class of problems.

The paper consists of the introduction and 6 main sections. In Sect. 2 we give some basic notations and definitions. In Sect. 3 we consider the Minty variational inequality with a relatively bounded operator and propose an adaptive algorithm for solving it. Sect. 4 is devoted to adaptive algorithms for relatively smooth optimization problems. In Sect. 5 we propose some universal algorithms for minimizing relatively smooth and relatively Lipschitz continuous functions. Sect. 6 is devoted to the numerical experiments which demonstrate the effectiveness of the proposed methods.

To sum it up, the contributions of the paper can be formulated as follows.

- We consider the variational inequality with the relatively bounded operator and propose some adaptive first-order methods to solve such a class of problems with optimal complexity estimates $O(\frac{1}{\varepsilon^2})$.
- We introduce adaptive and universal algorithms for minimizing relatively smooth and relatively Lipschitz continuous functions and provide their theoretical justification. The stopping criteria of the introduced adaptive algorithms are simple (which is especially important in terms of numerical experiments), but universal algorithms are guaranteed to be applicable to a wide class of problems. Our approach allows us to minimize the sum of relatively smooth and relatively Lipschitz continuous functions, even though such a sum does not satisfy neither relatively smoothness condition nor relatively Lipschitz one. Theoretical estimates of the proposed methods are optimal both for convex relatively Lipschitz minimization problems $O(\frac{1}{\varepsilon^2})$ and convex relatively smooth minimization problems $O(\frac{1}{\varepsilon})$.
- We provide the numerical experiments for the Intersection of Ellipsoids Problem (IEP) and the Lagrange saddle point problem for the Support Vector Machine (SVM) with inequality-type function constraints. We also, compare numerically, for (IEP), one of the proposed algorithms with the AdaMirr algorithm, which was recently proposed in [1]. The conducted experiments demonstrate that the proposed algorithms work better than AdaMirr and they can work faster than the obtained theoretical estimates in practice.

2. Basic definitions and notations

Let us give some basic definitions and notations concerning Bregman divergence and the prox structure, which will be used throughout the paper.

Let $(E, \|\cdot\|)$ be some normed finite-dimensional real vector space and E^* be its dual space with the norm

$$\|y\|_* = \max_x \{\langle y, x \rangle, \|x\| \leqslant 1\},$$

where $\langle y, x \rangle$ is the value of the linear function y at $x \in E$. Assume that $Q \subset E$ is a closed convex set (for variational inequalities in Sect. 3 we consider a convex compact set $Q \subset E$).

Let $d : Q \longrightarrow \mathbb{R}$ be a distance-generating function (d.g.f) which is continuously differentiable and convex.

For all $x, y \in Q \subset E$, we consider the corresponding Bregman divergence

$$V(y, x) = d(y) - d(x) - \langle \nabla d(x), y - x \rangle.$$

Now we introduce the following concept of (α, L, δ)-relative smoothness which covers both the concept of relative smoothness and relative Lipschitz continuity. Further, we denote by ∇f an arbitrary subgradient of f.

Definition 2.1. Let us call a convex function $f : Q \longrightarrow \mathbb{R}$ (α, L, δ)-*relatively smooth* for some $\alpha \in [0; 1]$, $L > 0$ and $\delta > 0$, if the following inequalities hold

$$(2.1) \quad f(y) \leqslant f(x) + \langle \nabla f(x), y - x \rangle + LV(y, x) + L\alpha V(x, y) + \delta, \quad \forall x, y \in Q,$$

$$(2.2) \quad \alpha \left(\langle \nabla f(x), y - x \rangle + LV(y, x) + \delta \right) \geqslant 0 \quad \forall x, y \in Q,$$

for each subgradient $\nabla f(x)$ of $f(x)$.

It is obvious that for $\alpha = 0$, $L > 0$, and $\delta = 0$ one gets the well-known relative smoothness condition (often defined as L-relative smoothness, see [3] for $\delta = 0$ and [12] for the case of $\delta > 0$). For $\alpha = 1$, $L = \frac{2M^2}{\varepsilon}$, and $\delta = \frac{\varepsilon}{4} > 0$, where ε is arbitrary, the inequalities (2.1) and (2.2) follow from the condition of the relative Lipschitz continuity (also known as relative continuity or M-relative Lipschitz continuity), proposed recently in [8, 11]

Definition 2.2. Convex function $f : Q \longrightarrow \mathbb{R}$ is called M-*relatively Lipschitz continuous* for some $M > 0$, if the following inequality holds

$$\langle \nabla f(x), y - x \rangle + M\sqrt{2V(y, x)} \geqslant 0 \quad \forall x, y \in Q.$$

Indeed, for each $x, y \in Q$ we have

$$\langle \nabla f(x), x - y \rangle \leqslant M\sqrt{2V(y, x)} \leqslant \frac{2M^2}{\varepsilon}V(y, x) + \frac{\varepsilon}{4}.$$

Further,

$$f(y) - f(x) \leqslant \langle \nabla f(y), y - x \rangle \leqslant M\sqrt{2V(x, y)} \leqslant \frac{2M^2}{\varepsilon}V(x, y) + \frac{\varepsilon}{4}$$

and

$$f(y) \leqslant f(x) + \langle \nabla f(x), y - x \rangle + \frac{2M^2}{\varepsilon}V(y, x) + \frac{2M^2}{\varepsilon}V(x, y) + \frac{\varepsilon}{2}.$$

So, each relatively Lipschitz continuous function f satisfies (2.1) for large enough $L > 0$ and $\delta > 0$.

It is worth mentioning that the sum of the relatively smooth function f_1 and relatively Lipschitz continuous convex function f_2 satisfies the (α, L, δ)-relative smoothness condition, if

$$f_1(y) \geqslant f_1(x) - rV(y, x) - q \quad \forall x, y \in Q,$$

for some fixed $r, q > 0$, and the corresponding values $\alpha, L, \delta > 0$ (this assumption can be understood as limiting the fast growth of f_1 and takes place, for example, when a function defined on a bounded set is bounded from below). Generally, such a sum is neither relatively smooth nor relatively Lipschitz continuous function.

Let us note the following important fact, which obviously follows from Lemma 3.2 from [12] and plays a key role in first inequalities in the following proofs. According to this fact for each operator $g : Q \to \mathbb{R}^n$

$$y = \arg \min_{x \in Q} \left\{ \langle g(z), x \rangle + \beta V(x, z) \right\}, \quad \beta \geq 0, z \in Q$$

we have

(2.3) $\qquad \langle g(z), x \rangle + \beta V(x, z) \geq \langle g(z), y \rangle + \beta V(y, z) + \beta V(x, y), \quad \forall x, z \in Q.$

3. Adaptive method for variational inequalities with relatively bounded operators

In this section we consider the Minty variational inequality problem (1.2) with relatively bounded (3.1) and monotone (3.2) operator g, i.e.

(3.1) $\qquad \langle g(x), x - y \rangle \leqslant M \sqrt{2V(y, x)} \quad \forall x, y \in Q,$

for some $M > 0$ and

(3.2) $\qquad \langle g(y) - g(x), y - x \rangle \geqslant 0 \quad \forall x, y \in Q,$

where Q is a convex compact set. In order to solve such a class of problems, we propose an adaptive algorithm, listed as Algorithm 1, below.

Algorithm 1 Adaptive Algorithm for Variational Inequalities with Relatively Bounded Operators.

Require: $\varepsilon > 0, x_0 \in Q, L_0 > 0, R > 0$ s.t. $\max_{x \in Q} V(x, x_0) \leqslant R^2, k = 0.$

1: Set $k = k + 1, L_{k+1} = \frac{L_k}{2}$.
2: Find
$$x_{k+1} = \arg \min_{x \in Q}\{\langle g(x_k), x \rangle + L_{k+1} V(x, x_k)\}.$$

3: **if**
$$\frac{\varepsilon}{2} + \langle g(x_k), x_{k+1} - x_k \rangle + L_{k+1} V(x_{k+1}, x_k) \geqslant 0,$$
    then go to the next iteration (item 1).
4: **else**
$$\text{set } L_{k+1} = 2L_{k+1}, \text{ and go to item 2.}$$

5: **end if**
6: Stopping criterion
$$S_N := \sum_{k=0}^{N-1} \frac{1}{L_{k+1}} \geqslant \frac{2R^2}{\varepsilon}.$$

Ensure: $\widehat{x} = \frac{1}{S_N} \sum_{k=0}^{N-1} \frac{x_{k+1}}{L_{k+1}}.$

Theorem 3.1. *Let $g : Q \longrightarrow \mathbb{R}^n$ be a relatively bounded and monotone operator, i.e. (3.1) and (3.2) hold, $L_0 \leqslant \frac{2M^2}{\varepsilon}$. Then after the stopping of Algorithm 1, the following inequality holds*

$$\max_{x \in Q} \langle g(x), \widehat{x} - x \rangle \leqslant \frac{1}{S_N} \sum_{k=0}^{N-1} \frac{1}{L_{k+1}} \langle g(x), x_k - x \rangle \leqslant \varepsilon.$$

Moreover, the total number of iterations will not exceed $N = \left\lceil \dfrac{4M^2R^2}{\varepsilon^2} \right\rceil$.

Proof. The proof is given in Appendix A. $\qquad\qquad\square$

Remark 3.2. Let us note, that defining

$$\Delta_N := \frac{1}{S_N} \max_{x \in Q} \sum_{k=0}^{N-1} \frac{1}{L_{k+1}} \langle g(x_k), x_k - x \rangle,$$

one can get that the convergence for the function's residuals

$$\min_{0 \leqslant k \leqslant N-1} f(x_k) - f^*,$$

for minimization problems with $g(x)$, defined as $g(x) = \nabla f(x)$, which also covers the primal-dual gap for saddle-point problems.

Let us consider the following modification of Algorithm 1 with adaptation both to the parameters $L = \frac{M^2}{\varepsilon}$ and $\delta = \frac{\varepsilon}{2}$.

Algorithm 2 Adaptation to Inexactness for Variational Inequalities with Relatively Bounded Operators.

Require: $\varepsilon > 0, x_0 \in Q, L_0 > 0, \delta_0 > 0, R$ s.t. $\max\limits_{x \in Q} V(x, x_0) \leqslant R^2, k = 0$.

 1: Set $k = k + 1, L_{k+1} = \frac{L_k}{2}, \delta_{k+1} = \frac{\delta_k}{2}$.
 2: Find

(3.3) $x_{k+1} = \arg \min\limits_{x \in Q} \{ \langle g(x_k), x \rangle + L_{k+1} V(x, x_k) \}.$

 3: **if**

(3.4) $0 \leqslant \langle g(x_k), x_{k+1} - x_k \rangle + L_{k+1} V(x_{k+1}, x_k) + \delta_{k+1},$

   then go to the next iteration (item 1).
 4: **else**

           set $L_{k+1} = 2 \cdot L_{k+1}, \delta_{k+1} = 2 \cdot \delta_{k+1}$ and go to item 2.

 5: **end if**
Ensure: $\widehat{x} = \frac{1}{S_N} \sum\limits_{k=0}^{N-1} \frac{x_{k+1}}{L_{k+1}}.$

Theorem 3.3. *Let* $g : Q \longrightarrow \mathbb{R}^n$ *be a relatively bounded and monotone operator, i.e. (3.1) and (3.2) hold,* $L_0 \leqslant 2L = \frac{2M^2}{\varepsilon}$. *Then after* N *steps of Algorithm 2 the following inequality holds*

$$(3.5) \qquad \max_{x \in Q} \langle g(x), \widehat{x} - x \rangle \leqslant \frac{R^2}{S_N} + \frac{1}{S_N} \sum_{k=0}^{N-1} \frac{\delta_{k+1}}{L_{k+1}}.$$

Moreover, if $L_0 \leqslant 2L$ *and* $\delta_0 \leqslant \varepsilon$, *the auxiliary problem (3.3) in Algorithm 2 is solved no more than* $2N + \log_2 \frac{2L}{L_0}$ *times.*

Proof. The proof is given in Appendix B. $\qquad \square$

Remark 3.4. Note, that Remark 3.2 also takes place for Theorem 3.3.

Remark 3.5. The condition of the relative boundedness is essential only for justifying (3.4). For $L_{k+1} \geqslant L = \frac{M^2}{\varepsilon}$ and $\delta_{k+1} \geqslant \frac{\varepsilon}{2}$, (3.4) certainly holds. So, if $L_0 \leqslant C_1 L$ for $C = \max \left\{ C_1; \frac{\varepsilon}{\delta_0} \right\}$, $L_{k+1} \leqslant CL$ and $\delta_{k+1} \leqslant \frac{C\varepsilon}{2} \; \forall k \geqslant 0$. Thus, $\max_{x \in Q} \langle g(x), \widehat{x} - x \rangle \leqslant \varepsilon$ after $N = O(\varepsilon^{-2})$ iterations of Algorithm 2. This fact, in essence, constitutes the optimality of the proposed method for the class of variational inequality problems with monotone M-relatively bounded operators.

4. ADAPTIVE ALGORITHMS FOR RELATIVELY LIPSCHITZ CONTINUOUS CONVEX OPTIMIZATION PROBLEMS

Now we consider the classical optimization problem (1.1) under the assumption of M-relative Lipschitz continuity of the objective function f. For solving such a type of problems we propose two adaptive algorithms, listed as Algorithm 3 and Algorithm 4, below.

Theorem 4.1. *Let* $f : Q \longrightarrow \mathbb{R}$ *be a convex and* M-*relatively Lipschitz continuous function, i.e. (2.1) and (2.2) take place with* $\alpha = 1, \delta = \frac{\varepsilon}{2}, L \geqslant \frac{M^2}{\varepsilon}$. *Then after the stopping of Algorithm 3, the following inequality holds* $f(\widehat{x}) - f(x_*) \leqslant \varepsilon$. *Moreover, the total number of iterations will not exceed* $N = \left\lceil \frac{4M^2 R^2}{\varepsilon^2} \right\rceil$.

Proof. The proof is given in Appendix C. $\qquad \square$

Theorem 4.2. *Let* $f : Q \longrightarrow \mathbb{R}$ *be a convex and* M-*relatively Lipschitz continuous function, i.e. (2.1) and (2.2) take place with* $\alpha = 1, \delta = \frac{\varepsilon}{2}, L = \frac{M^2}{\varepsilon}$. *Then after* N *steps of Algorithm 4, the following inequality holds*

$$(4.3) \qquad f(\widehat{x}) - f(x_*) \leqslant \frac{R^2}{S_N} + \frac{1}{S_N} \sum_{k=0}^{N-1} \frac{\delta_{k+1}}{L_{k+1}}.$$

Proof. The proof is similar to the proof of Theorem 4.1 with $\frac{\varepsilon}{2} \longrightarrow \delta_{k+1}$. $\qquad \square$

The optimality of Algorithm 4 for the class of convex and M-relatively Lipschitz continuous problems can be proved similar to Remark 3.5.

Algorithm 3 Adaptive Algorithm for Relatively Lipschitz Continuous Optimization Problems.

Require: $\varepsilon > 0, x_0 \in Q, L_0 > 0, R$ s.t. $V(x_*, x_0) \leqslant R^2, k = 0$.

1: Set $k = k + 1, L_{k+1} = \frac{L_k}{2}$.

2: Find
$$x_{k+1} = \arg\min_{x \in Q}\{\langle \nabla f(x_k), x \rangle + L_{k+1}V(x, x_k)\}.$$

3: **if**

$$(4.1) \qquad 0 \leqslant \langle \nabla f(x_k), x_{k+1} - x_k \rangle + L_{k+1}V(x_{k+1}, x_k) + \frac{\varepsilon}{2},$$

　　then go to the next iteration (item 1).

4: **else**
$$\text{set } L_{k+1} = 2 \cdot L_{k+1} \text{ and go to item 2.}$$

5: **end if**

6: Stopping criterion
$$S_N = \sum_{k=0}^{N-1} \frac{1}{L_{k+1}} \geqslant \frac{2R^2}{\varepsilon}.$$

Ensure: $\widehat{x} = \frac{1}{S_N} \sum_{k=0}^{N-1} \frac{x_{k+1}}{L_{k+1}}$.

Algorithm 4 Adaptation to Inexactness for Relatively Lipschitz Continuous Optimization Problems.

Require: $\varepsilon > 0, x_0 \in Q, L_0 > 0, \delta_0 > 0, R$ s.t. $V(x_*, x_0) \leqslant R^2, k = 0$.

1: Set $k = k + 1, L_{k+1} = \frac{L_k}{2}, \delta_{k+1} = \frac{\delta_k}{2}$.

2: Find
$$(4.2) \qquad x_{k+1} = \arg\min_{x \in Q}\{\langle \nabla f(x_k), x \rangle + L_{k+1}V(x, x_k)\}.$$

3: **if**
$$0 \leqslant \langle \nabla f(x_k), x_{k+1} - x_k \rangle + L_{k+1}V(x_{k+1}, x_k) + \delta_{k+1},$$
　　then go to the next iteration (item 1).

4: **else**
$$\text{set } L_{k+1} = 2 \cdot L_{k+1}, \delta_{k+1} = 2 \cdot \delta_{k+1} \text{ and go to item 2.}$$

5: **end if**

Ensure: $\widehat{x} = \frac{1}{S_N} \sum_{k=0}^{N-1} \frac{x_{k+1}}{L_{k+1}}$.

5. UNIVERSAL ALGORITHMS FOR RELATIVELY SMOOTH AND RELATIVELY LIPSCHITZ CONTINUOUS CONVEX OPTIMIZATION PROBLEMS

In this section, we introduce some analogues of Algorithms 3 and 4, which adjust to the "degree of relative smoothness" of the considered (α, L, δ)-relatively smooth problem. This approach allows the construction of adaptive gradient-type methods that are applicable to both relatively Lipschitz continuous and relatively smooth problems with optimal complexity estimates.

Algorithm 5 Universal Method for Relatively Smooth and Relatively Lipschitz Continuous Convex Optimization Problems with Adaptation to Inexactness.

Require: $\varepsilon > 0, x_0 \in Q, L_0 > 0, \delta_0 > 0, R$ s.t. $V(x_*, x_0) \leqslant R^2, k = 0$.

1: Set $k = k + 1, L_{k+1} = \frac{L_k}{2}, \delta_{k+1} = \frac{\delta_k}{2}$.
2: Find

$$(5.1) \qquad x_{k+1} = \arg \min_{x \in Q}\{\langle \nabla f(x_k), x \rangle + L_{k+1}V(x, x_k)\}.$$

3: **If**

$$f(x_{k+1}) \leqslant f(x_k) + \langle \nabla f(x_k), x_{k+1} - x_k \rangle + L_{k+1}V(x_{k+1}, x_k) + \delta_{k+1},$$

   then go to the next iteration (item 1).
4: **else**

$$\text{set } L_{k+1} = 2 \cdot L_{k+1}, \delta_{k+1} = 2 \cdot \delta_{k+1} \text{ and go to item 2.}$$

5: **end if**
Ensure: $\widehat{x} = \frac{1}{S_N} \sum_{k=0}^{N-1} \frac{x_{k+1}}{L_{k+1}}$.

Theorem 5.1. *Let $f : Q \longrightarrow \mathbb{R}$ be a convex and (α, L, δ)-relatively smooth function, i.e. (2.1), (2.2) hold. Then after N iterations of Algorithm 5, the following inequality holds*

$$f(\widehat{x}) - f(x_*) \leqslant \frac{R^2}{S_N} + \frac{1}{S_N}\sum_{k=0}^{N-1}\frac{\delta_{k+1}}{L_{k+1}},$$

where $S_N = \sum_{k=0}^{N-1}\frac{1}{L_{k+1}}$. Note that for $L_0 \leqslant 2L$ and $\delta_0 \leqslant 2\delta$ the auxiliary problem (5.1) in Algorithm 5 is solved no more than $2N + \log_2 \frac{2L}{L_0}$ times.

Proof. The proof is given in Appendix D. $\qquad\square$

The optimality of Algorithm 5 for the class of convex and M-relatively Lipschitz continuous problems can be proved similar to Remark 3.5. The optimal rate of convergence $O(\varepsilon^{-1})$ for the class of L-relatively smooth problems also takes place for Algorithm 5. For more details see the conclusion of proof in Appendix E, the proof of these facts for Algorithm 5 can be obtained analogously.

Let us now formulate a variant of the universal method for relatively Lipschitz continuous and relatively smooth problems which makes it possible to prove the

guaranteed preservation of the optimal complexity estimates. This method is listed as Algorithm 6, below.

Algorithm 6 Universal Method for Relatively Smooth and Relatively Lipschitz Continuous Convex Optimization Problems.

Require: $\varepsilon > 0, x_0 \in Q, L_0 > 0, R$ s.t. $V(x_*, x_0) \leqslant R^2, k = 0$.

1: Set $k = k + 1, L_{k+1} = \frac{L_k}{2}$.
2: Find
$$x_{k+1} = \arg\min_{x \in Q}\{\langle \nabla f(x_k), x\rangle + L_{k+1}V(x, x_k)\}.$$

3: **If**
$$f(x_{k+1}) \leqslant f(x_k) + \langle \nabla f(x_k), x_{k+1} - x_k\rangle + L_{k+1}V(x_{k+1}, x_k) + \frac{3\varepsilon}{4},$$

   then go to the next iteration (item 1).
4: **else**
$$\text{set } L_{k+1} = 2 \cdot L_{k+1} \text{ and go to item 2}.$$

5: **end if**
6: Stopping criterion
$$S_N := \sum_{k=0}^{N-1} \frac{1}{L_{k+1}} \geqslant \frac{4R^2}{\varepsilon}.$$

Ensure: $\widehat{x} = \frac{1}{S_N}\sum_{k=0}^{N-1}\frac{x_{k+1}}{L_{k+1}}$.

Theorem 5.2. *Let $f : Q \longrightarrow \mathbb{R}$ be a convex and (α, L, δ)-relatively smooth function, i.e. (2.1) and (2.2) hold with $\delta \leqslant \frac{3\varepsilon}{4}$, $L_0 \leqslant 2L$. Then after the stopping of Algorithm 6, the following inequality holds $f(\widehat{x}) - f(x_*) \leqslant \varepsilon$. The number of iterations of Algorithm 6 does not exceed $\left\lceil \frac{8LR^2}{\varepsilon} \right\rceil$. If f is $\left(1, \frac{2M^2}{\varepsilon}, \frac{\varepsilon}{2}\right)$-relatively smooth function (for example, M-relatively Lipschitz continuous function) then the number of iterations of Algorithm 6 does not exceed $\left\lceil \frac{16M^2R^2}{\varepsilon^2} \right\rceil$.*

Proof. The proof is given in Appendix E. □

Remark 5.3. It is worth noting that, generally speaking, for Algorithms 3–6 it is acceptable to use the following output point
$$\widehat{x} = \arg\min_{i\in\{0,\dots,N+1\}} f(x_i).$$

At the same time for various applied problems, such a modification can both improve and degrade the practical quality of the algorithms.

6. NUMERICAL EXPERIMENTS

In this section, in order to demonstrate the performance of the proposed Algorithms, we firstly consider some numerical experiments concerning the Intersection of Ellipsoids Problem (IEP). Secondly, we compare the proposed Algorithm 4 with AdaMirr algorithm, which was recently proposed in [1]. We also, consider some numerical experiments concerning the Support Vector Machine (SVM) [8, 13].

All experiments were implemented in Python 3.4, on a computer fitted with Intel(R) Core(TM) i7-8550U CPU @ 1.80GHz, 4 Core(s), 8 Logical Processor(s). The RAM of the computer is 8 GB.

6.1. **The Intersection of Ellipsoids Problem (IEP).** For the Intersection of Ellipsoids Problem, supposing, that the intersection is nonempty, we compute a point $x \in \mathbb{R}^n$ in the intersection of m ellipsoids, i.e.

$$x \in \mathcal{E} = \mathcal{E}_1 \cap \mathcal{E}_2 \cap \ldots \cap \mathcal{E}_m,$$

where $\mathcal{E}_i = \left\{ x \in \mathbb{R}^n : \frac{1}{2}x^T A_i x + b_i^T x + c_i \leqslant 0 \right\}$, $A_i \in \mathbb{R}^{n \times n}$ is a given symmetric positive semi-definite matrix, $b_i \in \mathbb{R}^n, c_i \in \mathbb{R}$ are given, for every $i = 1, \ldots, m$. We note that the Intersection of Ellipsoids Problem is equivalent to the following unconstrained optimization problem

$$(6.1) \qquad \min_{x \in \mathbb{R}^n} \left\{ f(x) := \max_{1 \leqslant i \leqslant m} \left[\frac{1}{2} x^T A_i x + b_i^T x + c_i \right] \right\}.$$

The objective function f in (6.1) is both non-differentiable and non-Lipschitz [8]. So the traditional first-order methods are not applicable to such types of problems. We will demonstrate, how the proposed Algorithm 4 can be applied to solve such a problem (here we take more attention to the Algorithm 4, because it works better than Algorithms 3 and 5, see Fig. 2).

Let $\sigma := \max_{1 \leqslant i \leqslant m} \|A_i\|_2^2$ where $\|A_i\|_2$ is the spectral norm of A_i,

$\rho := \max_{1 \leqslant i \leqslant m} \|A_i b_i\|_2$ and $\gamma := \max_{1 \leqslant i \leqslant m} \|b_i\|_2^2$. We run Algorithm 4 with the following prox function

$$(6.2) \qquad d(x) := \frac{a_2}{4}\|x\|_2^4 + \frac{a_1}{3}\|x\|_2^3 + \frac{a_0}{2}\|x\|_2^2,$$

where $a_0 = \gamma, a_1 = \rho, a_2 = \sigma$ (see [8] for more details). The objective function f (6.1) is 1-relatively Lipschitz continuous with respect to the prox function $d(\cdot)$, defined in (6.2) [8]. The Bregman divergence $V(\cdot, \cdot)$ for the corresponding prox function $d(\cdot)$ is defined as follows

$$(6.3) \qquad V(y, x) = a_0 V_{d_0}(y, x) + a_1 V_{d_1}(y, x) + a_2 V_{d_2}(y, x).$$

where $d_i(x) = \frac{1}{i+2}\|x\|_2^{i+2}$ $(i = 0, 1, 2)$, and

$$V_{d_i}(y, x) = \frac{1}{i+2} \left(\|y\|_2^{i+2} + (i+1)\|x\|_2^{i+2} - (i+2)\|x\|_2^i \langle x, y \rangle \right) \ (i = 0, 1, 2).$$

Note that each iteration of Algorithm 4 requires the capability to solve the subproblem (4.2), which is equivalent to the following linearized problem

$$(6.4) \qquad x_{k+1} = \arg\min_{x \in \mathbb{R}^n} \{ \langle c_k, x \rangle + d(x) \},$$

where $c_k = \frac{1}{L_{k+1}}\nabla f(x_k) - \nabla d(x_k)$ and $d(x)$ is given in (6.2). The solution of the problem (6.4) can be found explicitly

$$x_{k+1} = -\theta_k c_k,$$

for some $\theta_k \geqslant 0$, where θ_k is a positive real root of the following cubic equation

$$\gamma\theta + \rho\|c_k\|_2\theta^2 + \sigma\|c_k\|_2^2\theta^3 - 1 = 0.$$

We run Algorithm 4 with different values of n and prox-function (6.2) and the starting point $x_0 = (0.2, \dots, 0.2) \in \mathbb{R}^n$ and not in $\mathcal{E}$. The matrices A_i, for every $i = 1, \dots, m$, are diagonal matrices with entries chosen randomly from the uniform distribution over $(0, 1)$, the vectors b_i and the constants c_i, are also chosen randomly from a normal (Gaussian) distribution with mean (center) equaling 0 and standard deviation (width) equaling 0.1. We generated the random data 5 times and averaged the results of algorithms that we received each time, such that the $\mathbf{0} \in \mathbb{R}^n \in \mathcal{E}$. We considered $L_0 = \frac{\|\nabla f(1,0,\dots,0) - \nabla f(0,1,0,\dots,0)\|_2}{\sqrt{2}}, \delta_0 = 0.5$, and $R^2 = \frac{3\sigma}{4}\|x_0\|_2^3 + \frac{2\rho}{3}\|x_0\|_2^3 + \frac{\gamma}{2}\|x_0\|_2^2$ (see the proof of the Proposition 5.4 in [8]).

The results of the work of Algorithm 4 for IEP are presented in Fig 1, below. These results demonstrate the running time of the algorithm in seconds as a function of the number of iterations, and the quality of the solution "Estimate", which is in fact the right side of inequality (4.3).

We note that the quality of the solution of the problem, which is produced by Algorithm 4, grows sharply at the beginning of the work of the algorithm for $k = 100$ to $10\,000$. We improve the quality of the initial solution by two orders of magnitude on average. Nevertheless, the rate of convergence significantly decreases when the number of iterations goes from $k = 15\,000$ to $100\,000$.
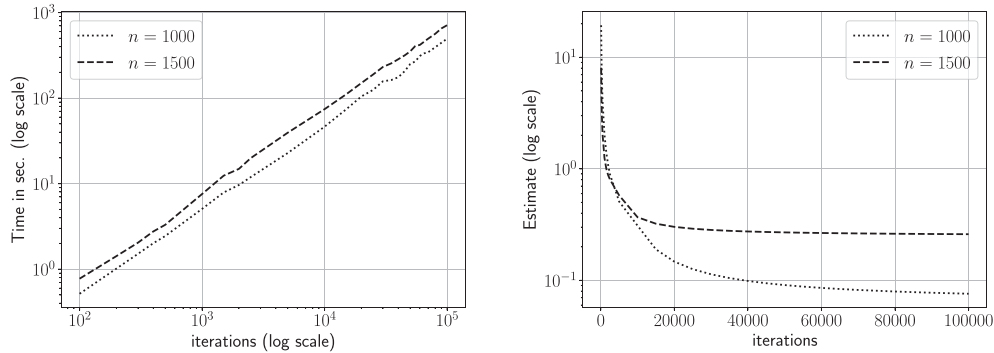

Figure 1. The results of Algorithm 4 for IEP with different values of n and $m = 10$.

6.2. **Comparison with AdaMirr.** Recently in [1], there was proposed an adaptive first-order method, called AdaMirr, in order to solve the relatively continuous and relatively smooth optimization problems. AdaMirr briefly can be stated as

$$x_{k+1} = \arg\min_{x \in Q}\{\langle -\gamma_k \nabla f(x_k), x_k - x\rangle + V(x, x_k)\}, \quad k = 1, 2, \dots.$$
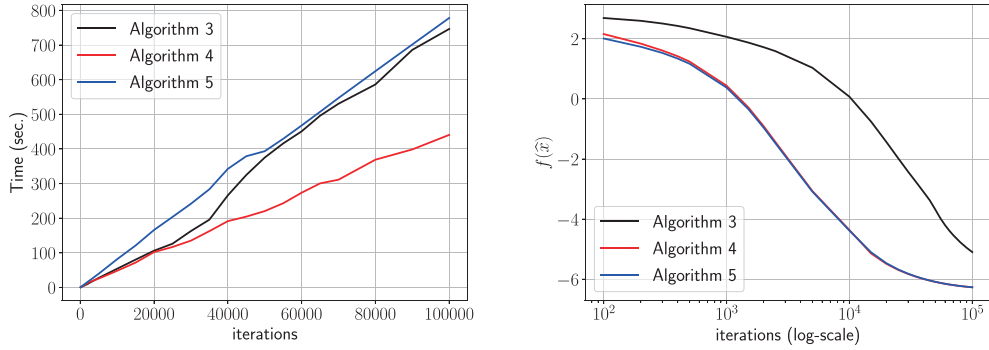
Figure 2. The results of Algorithms 3, 4 and 5 for IEP with $n = 1000$ and $m = 10$.

with γ_k defined as

$$\gamma_k = \frac{1}{\sqrt{\sum_{s=0}^{k-1} \delta_s^2}} \quad \text{with} \quad \delta_s^2 = \frac{V(x_s, x_{s+1}) + V(x_{s+1}, x_s)}{\gamma_s^2}, \quad k = 1, 2, \ldots$$

and $\delta_0 = \sqrt{V(x_0, x_1) + V(x_1, x_0)}$. In [1], it was proved that for M-relatively Lipschitz continuous convex function, then after N steps of AdaMirr, the following inequality holds

$$(6.5) \qquad f(\overline{x}_N) - f^* \leqslant \frac{\sqrt{2}M\left[D_1 + \frac{8M^2}{\delta_0^2} + 2\ln\left(1 + \frac{2M^2N}{\delta_0^2}\right)\right]}{\sqrt{N}} + \frac{3\sqrt{2}M + \frac{4M^2}{\delta_0^2}}{N},$$

where $\overline{x}_N = \frac{1}{N}\sum_{k=1}^{N} x_k$ and $D_1 = V(x_*, x_1)$.

In this subsection, we compare the proposed Algorithm 4 with AdaMirr, for the Intersection of Ellipsoids Problem (see Subsec. 6.1). We run the compared algorithms for the same parameters and setting as in the Subsec. 6.1. The results of the comparison are presented in Fig. 3, which illustrates the value of the objective function at the output point of each algorithm, the estimates of the quality of the solution for Algorithm 4 (see the right side of inequality (4.3)) and AdaMirr (see the right side of inequality (6.5)) and the running time of algorithms in seconds.

From the results in Fig. 3, we can see that the proposed Algorithm 4 works better than AdaMirr, except for the running time, where AdaMirr works faster because, in the proposed Algorithm 4, there is an adaptive procedure for the Lipschitz continuity parameter, which needs more time. Note that AdaMirr does not converge to the solution of the problem for all taken iterations from 100 to 10^4.

6.3. **Support Vector Machine (SVM) and Inequality-Type Function Constraints.** The Support Vector Machine (SVM) is an important supervised learning model for binary classification problem [13]. The SVM optimization problem can be formulated as follows

$$(6.6) \qquad \min_{x \in \widetilde{Q}} \left\{ f(x) := \left(\frac{1}{n}\sum_{i=1}^{n} \max\{0, 1 - y_i x^\top w_i\}\right) + \frac{\tau}{2}\|x\|_2^2 \right\},$$
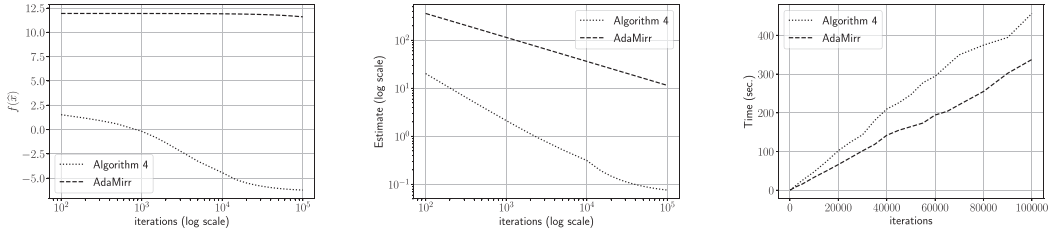
Figure 3. The results of comparison of Algorithm 4 and AdaMirr for IEP with $n = 1000$ and $m = 10$.

where w_i is the input feature vector of sample i and $y_i \in \{-1, 1\}$ is the label of sample i, $\tau > 0$ is the regularization parameter, and $\widetilde{Q}$ is a compact convex set. The objective function in (6.6) is non-differentiable and because of the existence of the ℓ_2-norm regularization the value of the Lipschitz constant of such a function can be extremely large. Thus, we cannot always directly use typical subgradient or gradient schemes to solve the problem (6.6). The problem of constrained (inequality-type constraints) minimization of convex functions attracts widespread interest in many areas of modern large-scale optimization and its applications [4, 14]. Therefore, we demonstrate the performance of the proposed Algorithm 1 for such class of problems. We consider an example of the Lagrange saddle point problem induced by the function f in the problem (6.6), with some inequality-type function constraints. This problem has the following form

$$(6.7) \qquad \min_{x \in \widetilde{Q}} \left\{ f(x) \; \middle| \; \varphi_p(x) := \sum_{i=1}^{n} \alpha_{pi} x_i^2 - \beta_p \leqslant 0 \; , \; p = 1, ..., m \right\},$$

where $\alpha_{pi} > 0, \beta_p \in \mathbb{R}, \forall i = 1, \ldots, n$ and $\forall p = 1, \ldots, m$. The corresponding Lagrange saddle point problem is defined as follows

$$\min_{x \in \widetilde{Q}} \max_{\boldsymbol{\lambda} = (\lambda_1, \lambda_2, \ldots, \lambda_m)^\top \in \widehat{Q} \subset \mathbb{R}_+^m} \left\{ L(x, \boldsymbol{\lambda}) := f(x) + \sum_{p=1}^{m} \lambda_p \varphi_p(x) \right\},$$

where $\widehat{Q}$ is a compact convex set. This problem is equivalent to the variational inequality with the following monotone bounded operator

$$G(x, \boldsymbol{\lambda}) = \begin{pmatrix} \nabla f(x) + \sum_{p=1}^{m} \lambda_p \nabla \varphi_p(x), \\ (-\varphi_1(x), -\varphi_2(x), \ldots, -\varphi_m(x))^\top \end{pmatrix},$$

where ∇f and $\nabla \varphi_p$ are subgradients of f and φ_p.

  We run Algorithm 1 with the following prox function

$$d(x, \boldsymbol{\lambda}) := \frac{a_2}{4} \|x\|_2^4 + \frac{a_1}{3} \|x\|_2^3 + \frac{a_0}{2} \|x\|_2^2 + \frac{1}{2} \|\boldsymbol{\lambda}\|_2^2, \quad \forall x \in \mathbb{R}^n, \boldsymbol{\lambda} \in \mathbb{R}_+^m$$

with

$$(6.8) \qquad a_0 = \frac{1}{n} \sum_{i=1}^{n} \|w_i\|_2^2, \quad a_1 = \frac{2\tau}{n} \sum_{i=1}^{n} \|w_i\|_2, \quad a_2 = \tau^2,$$

and the following Bregman divergence

$$(6.9) \qquad V_{\text{new}}\left((y, \boldsymbol{\lambda}), (x, \boldsymbol{\lambda}')\right) = V(y, x) + \frac{1}{2}\|\boldsymbol{\lambda} - \boldsymbol{\lambda}'\|_2^2,$$

for every $x, y \in \mathbb{R}^n, \boldsymbol{\lambda}, \boldsymbol{\lambda}' \in \mathbb{R}_+^m$, where $V(y, x)$ is given in (6.3) with coefficients defined in (6.8). We consider the ball $\widetilde{Q} \subset \mathbb{R}^n$ at the center $\mathbf{0} \in \mathbb{R}^n$ and the radius $r = \min\left\{\frac{1}{n\tau}\sum_{i=1}^{n}\|w_i\|_2, \sqrt{\frac{2}{\tau}}\right\}$ (see [8]). We take the initial point $(x_0, \boldsymbol{\lambda}_0) \in \mathbb{R}^{n+m}$, with all coordinates equaling 0.01. The coefficients α_{pi} in (6.7), and the vectors w_i for $i = 1, \ldots, n$ are chosen randomly from the uniform distribution over $[0, 1)$, and $\beta_p = r \, (\forall p = 1, \ldots, m)$. We also consider $\widehat{Q} = \{\boldsymbol{\lambda} \in \mathbb{R}_+^m : \|\boldsymbol{\lambda}\|_2^2 \leqslant r^2\}$, $L_0 = \frac{\|G((1,0,\ldots,0),\mathbf{0}) - G((0,1,0,\ldots,0),\mathbf{0})\|_2}{\sqrt{2}}$, where $\mathbf{0} \in \mathbb{R}^m$, $\delta_0 = 0.5$, and $\tau = 0.5$ in (6.6). In order to estimate the parameter R, for the Bregman divergence (6.9), we have (see [8])

$$\begin{aligned} V_{\text{new}}\left((x, \boldsymbol{\lambda}), (x_0, \boldsymbol{\lambda}_0)\right) \leqslant{} & \frac{a_2}{4}\|x - x_0\|_2^2 \left(\|x + x_0\|_2^2 + 2\|x_0\|_2^2\right) \\ & + \frac{a_1}{3}\|x - x_0\|_2^2 \left(\|x\|_2^2 + 2\|x_0\|_2^2\right) + \frac{a_0}{2}\|x - x_0\|_2^2 \\ & + \frac{1}{2}\|\boldsymbol{\lambda} - \boldsymbol{\lambda}_0\|_2^2. \end{aligned}$$

Therefore in $Q := \widetilde{Q} \times \widehat{Q}$, we have

$$\begin{aligned} & V_{\text{new}}\left((x, \boldsymbol{\lambda}), (x_0, \boldsymbol{\lambda}_0)\right) \\ & \leqslant (r + \|x_0\|_2)^2 \left[\frac{a_2}{4}\left(r^2 + 2r\|x_0\|_2 + 3\|x_0\|_2^2\right) + \frac{a_1}{3}\left(r^2 + 2\|x_0\|_2^2\right) + \frac{a_0}{2}\right] \\ & \quad + \frac{1}{2}(r + \|\lambda_0\|_2)^2 := v. \end{aligned}$$

Thus we can take $R = \sqrt{v}$. In each iteration of Algorithm 1, solving the sub-problem (2), for the problem (6.7), will be automatically (not explicitly as was for IEP in the previous subsections).

The results of the work of Algorithm 1, for $n = 25, m = 5$, are presented in Fig. 4. These results demonstrate the number of the iterations of Algorithm 1, as a function of $\varepsilon \in \{i^{-1}, i = 2, 4, 8, 12, 16, 20\}$. As it is known, for the variational inequality with a non-smooth operator, the theoretical complexity estimate $O\left(\varepsilon^{-2}\right)$ is optimal. But experimentally we can see from Fig. 4 that, the proposed Algorithm 1, has iteration complexity nearly to $O\left(\varepsilon^{-1}\right)$, which is an optimal estimate for the problems with smooth operators.

Conclusions

In this paper, we considered (α, L, δ)-relatively smooth optimization problems which provide for the possibility of minimizing both relatively smooth and relatively Lipschitz continuous functions. For such a type of problems, we introduced some adaptive and universal methods with optimal estimates of the convergence rate. We also considered the problem of solving the variational inequality with a relatively bounded operator. Finally, we presented the results of numerical experiments for the considered algorithms.
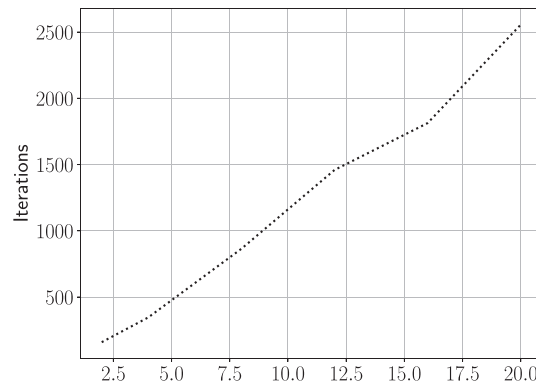
Figure 4. The results of Algorithm 1, for the problem (6.7).

The authors are very grateful to Dmitry Pasechnyuk for fruitful discussions. Also, the authors are very grateful to the unknown reviewers for their extremely valuable comments.

REFERENCES

[1] K. Antonakopoulos and P. Mertikopoulos, *Adaptive first-order methods revisited: Convex optimization without Lipschitz requirements*, NeurIPS (2021), `https:// arxiv.org/ pdf/2107.08011.pdf`

[2] K. Antonakopoulos, E. V. Belmega and P. Mertikopoulos, *An adaptive mirror-prox algorithm for variational inequalities with singular operators*, In NeurIPS (2019)

[3] H. H. Bauschke, J. Bolte and M. Teboulle, *A descent lemma beyond Lipschitz gradient continuity: first-order methods revisited and applications*, Mathematics of Operations Research **42** (2017), 330–348.

[4] A. Ben-Tal and A. Nemirovski, *Robust Truss Topology Design via Semidefinite Programming*, SIAM J. Optim. **7** (1997), 991–1016.

[5] R. A. Dragomir, A. B. Taylor, A. d'Aspremont and J. Bolte, *Optimal complexity and certification of Bregman first-order methods*, Mathematical Programming, Series A **194** (2022), 41–83.

[6] H. Lu, R. Freund and Yu. Nesterov, *Relatively smooth convex optimization by first-order methods, and applications*, SIOPT **28** (2018), 333–354.

[7] D. Kamzolov, A. Gasnikov, P. Dvurechensky, A. Agafonov and M. Takáč *Exploiting higher-order derivatives in convex optimization methods* (2022) `https://arxiv.org/ pdf/2208.13190.pdf`

[8] H. Lu, *Relative Continuity for Non-Lipschitz Nonsmooth Convex Optimization Using Stochastic (or Deterministic) Mirror Descent*, Informs Journal on Optimization **1** (2019), 288–303.

[9] Y. Nesterov, *Implementable tensor methods in unconstrained convex optimization*, Mathematical Programming, Series A **186** (2021), 157–183.

[10] Y. Nesterov, *Inexact accelerated high-order proximal-point methods*, (No. UCL-Université Catholique de Louvain) CORE. (2020)

[11] Y. Nesterov, *Relative Smoothness: New Paradigm in Convex Optimization*, Conference report, EUSIPCO-2019, A Coruna, Spain, September 4, 2019.

[12] F. Stonyakin, A. Tyurin, A. Gasnikov, P. Dvurechensky, A. Agafonov, D. Dvinskikh, M. Alkousa, D. Pasechnyuk, S. Artamonov and V. Piskunova, *Inexact relative smoothness and strong convexity for optimization and variational inequalities by inexact model*, Optim. Methods and Software **36** (2021), 1155–1201.

[13] S. S. Shwartz, Y. Singer, N. Srebro and A. Cotter, *Pegasos: primal estimated sub-gradient solver for SVM*, Mathematical Programming **127** (2011), 3–30.

[14] S. Shpirko and Yu. Nesterov, *Primal-dual subgradient methods for huge-scale linear conic problem*, SIAM J. Optim. **24** (2014), 1444–1457.

[15] O. S. Savchuk, A. A. Titov, F. S. Stonyakin and M. S. Alkousa, *Adaptive first-order methods for relatively strongly convex optimization problems*, Computer Research and Modeling **14** (2022), 445–472.

[16] A. A. Titov, F. S. Stonyakin, M. S. Alkousa, S. S. Ablaev and A. V. Gasnikov, *analogues of switching subgradient schemes for relatively Lipschitz-continuous convex programming problems*, in: International Conference on Mathematical Optimization Theory and Operations Research, Springer, Cham, 2020, pp. 133–149.

[17] A. Titov, F. Stonyakin, M. Alkousa and A. Gasnikov, *Algorithms for solving variational inequalities and saddle point problems with some generalizations of Lipschitz property for operators*, in: Communications in Computer and Information Science, vol. 1476, Springer, Cham, 2021, pp. 86–101.

APPENDIX A. THE PROOF OF THEOREM 3.1

Proof. Due to (2.3) and (3.3), for each $x \in Q$, we have

$$\langle g(x_k), x_{k+1} - x \rangle \leqslant L_{k+1}V(x, x_k) - L_{k+1}V(x, x_{k+1}) - L_{k+1}V(x_{k+1}, x_k).$$

Thus, taking into account (3.4) and monotonicity of g, we get

$$
\begin{aligned}
L_{k+1}V(x, x_k) - L_{k+1}V(x, x_{k+1}) &\geqslant \langle g(x_k), x_{k+1} - x \rangle + L_{k+1}V(x_{k+1}, x_k) \\
&= \langle g(x_k), x_{k+1} - x \rangle + \langle g(x_k), x_{k+1} - x_k \rangle \\
&\quad + L_{k+1}V(x_{k+1}, x_k) - \langle g(x_k), x_{k+1} - x_k \rangle. \\
&\geqslant \langle g(x_k), x_{k+1} - x \rangle - \langle g(x_k), x_{k+1} - x_k \rangle - \frac{\varepsilon}{2} \\
&= \langle g(x_k), x_k - x \rangle - \frac{\varepsilon}{2} \geqslant \langle g(x_k), x_k - x \rangle - \frac{\varepsilon}{2},
\end{aligned}
$$

whence we obtain that

(6.10) $$\langle g(x), x_k - x \rangle \leqslant L_{k+1}V(x, x_k) - L_{k+1}V(x, x_{k+1}) + \frac{\varepsilon}{2}, \quad \forall x \in Q.$$

Taking summation over both sides of (6.10), we have

$$\sum_{k=0}^{N-1} \frac{1}{L_{k+1}} \langle g(x), x_k - x \rangle \leqslant V(x, x_0) + \sum_{k=0}^{N-1} \frac{\varepsilon}{2L_{k+1}}, \quad \forall x \in Q,$$

which leads to

$$\langle g(x), \widehat{x} - x \rangle \leqslant \frac{1}{S_N} \sum_{k=0}^{N-1} \frac{1}{L_{k+1}} \langle g(x), x_k - x \rangle \leqslant \frac{R^2}{S_N} + \frac{\varepsilon}{2},$$

and

$$\max_{x \in Q} \langle g(x), \widehat{x} - x \rangle \leqslant \frac{R^2}{S_N} + \frac{\varepsilon}{2}.$$

As operator g is relatively bounded, i.e.

$$\langle g(x), x - y \rangle \leqslant M\sqrt{2V(y, x)} \leqslant \frac{M^2 V(y, x)}{\varepsilon} + \frac{\varepsilon}{2}, \quad \forall \varepsilon > 0,$$

the stopping criterion of Algorithm 1 is guaranteed to be satisfied for $L_{k+1} \geqslant \frac{M^2}{\varepsilon}$. Since the exit from the iteration will certainly happen for $L_{k+1} \leqslant \frac{2M^2}{\varepsilon}$, we have

$$\frac{R^2}{S_N} \leqslant \frac{2M^2 R^2}{\varepsilon N}.$$

Thus, the total number of iterations of Algorithm 1 will not exceed

$$N = \left\lceil \frac{4M^2 R^2}{\varepsilon^2} \right\rceil.$$

$\square$

Appendix B. The proof of Theorem 3.3

Proof. The proof of (3.5) is similar to the proof of Theorem 3.1, with $\frac{\varepsilon}{2} = \delta_{k+1}$. Let us assume that on the $(k+1)$-th iteration $(k = 0, 1, \ldots, N-1)$ of the Algorithm 2, the auxiliary problem (3.3) is solved i_{k+1} times. Then

$$2^{i_{k+1}-2} = \frac{L_{k+1}}{L_k} = \frac{\delta_{k+1}}{\delta_k},$$

since at the beginning of each iteration the parameters L_k, δ_k are divided by 2. Therefore,

$$\sum_{k=0}^{N-1} i_{k+1} = 2N + \log_2 \frac{L_N}{L_0}, \quad \log_2 \frac{L_N}{L_0} = \log_2 \frac{\delta_N}{\delta_0}.$$

It is clear that at least one of the inequalities $L_N \leqslant 2L, \delta_N \leqslant 2\delta$ holds, which ends the proof of the theorem. $\square$

Appendix C. The proof of Theorem 4.1

Proof. Let us use the reasoning in the proof of Theorem 3.1 for $g(x) = \nabla f(x)$. Taking into account (2.3), for any $x \in Q$, we have,

$$\langle \nabla f(x_k), x_{k+1} - x \rangle \leqslant L_{k+1} V(x, x_k) - L_{k+1} V(x, x_{k+1}) - L_{k+1} V(x_{k+1}, x_k).$$

Thus, taking into account (4.1), we get

$$\begin{aligned}
L_{k+1} V(x, x_k) - L_{k+1} V(x, x_{k+1}) &\geqslant \langle \nabla f(x_k), x_{k+1} - x \rangle + L_{k+1} V(x_{k+1}, x_k) \\
&= \langle \nabla f(x_k), x_{k+1} - x \rangle + \langle \nabla f(x_k), x_{k+1} - x_k \rangle \\
&\quad + L_{k+1} V(x_{k+1}, x_k) - \langle \nabla f(x_k), x_{k+1} - x_k \rangle \\
&\geqslant \langle \nabla f(x_k), x_{k+1} - x \rangle - \langle \nabla f(x_k), x_{k+1} - x_k \rangle - \frac{\varepsilon}{2} \\
&= \langle \nabla f(x_k), x_k - x \rangle - \frac{\varepsilon}{2}.
\end{aligned}$$

So, we have

$$(6.11) \qquad \langle \nabla f(x_k), x_k - x \rangle \leqslant L_{k+1} V(x, x_k) - L_{k+1} V(x, x_{k+1}) + \frac{\varepsilon}{2}, \quad \forall x \in Q.$$

Taking summation over both sides of (6.11), we obtain

$$\sum_{k=0}^{N-1} \frac{1}{L_{k+1}} \langle \nabla f(x_k), x_k - x \rangle \leqslant V(x, x_0) + \sum_{k=0}^{N-1} \frac{\varepsilon}{2L_{k+1}}, \quad \forall x \in Q.$$

Further, in view of the inequality

$$\langle \nabla f(x_k), x_k - x \rangle \geqslant f(x_k) - f(x),$$

we have

$$\sum_{k=0}^{N-1} \frac{1}{L_{k+1}} \left(f(x_k) - f(x) \right) \leqslant V(x, x_0) + \sum_{k=0}^{N-1} \frac{\varepsilon}{2L_{k+1}}.$$

Moreover, since f is convex, the following inequality holds

$$\sum_{k=0}^{N-1} \frac{1}{L_{k+1}} f(x_k) \geqslant S_N f(\widehat{x}),$$

where $S_N = \sum\limits_{k=0}^{N-1} \frac{1}{L_{k+1}}$. Then

(6.12) $$\sum_{k=0}^{N-1} \frac{1}{L_{k+1}} \left(f(x_k) - f(x) \right) \leqslant V(x, x_0) + \frac{\varepsilon}{2} S_N.$$

Since $V(x_*, x_0) \leqslant R^2$, we obtain, for $x = x_*$ in (6.12), that

$$f(\widehat{x}) - f(x_*) \leqslant \frac{R^2}{S_N} + \frac{\varepsilon}{2}.$$

□

APPENDIX D. THE PROOF OF THEOREM 5.1

Proof. 1) Taking into account the standard minimum condition for the subproblem (5.1) and (2.3), we have

$$\langle \nabla f(x_k) + L_{k+1} \nabla_{x=x_{k+1}} V(x, x_k), x - x_k \rangle \geqslant 0, \quad \forall x \in Q.$$

After the completion of the k-th iteration ($k = 0, 1, \ldots$) of the Algorithm 5, the following inequalities hold

$$\langle \nabla f(x_k), x_{k+1} - x \rangle \leqslant L_{k+1} V(x, x_k) - L_{k+1} V(x, x_{k+1}) - L_{k+1} V(x_{k+1}, x_k),$$

and

$$f(x_{k+1}) \leqslant f(x_k) + \langle \nabla f(x_k), x_{k+1} - x_k \rangle + L_{k+1} V(x, x_k) - L_{k+1} V(x, x_{k+1}) + \delta_{k+1}.$$

Therefore,

$$f(x_{k+1}) \leqslant f(x_k) + \langle \nabla f(x_k), x - x_k \rangle + L_{k+1} V(x, x_k) - L_{k+1} V(x, x_{k+1}) + \delta_{k+1}.$$

Further, taking into account the inequality $f(x_k) + \langle \nabla f(x_k), x - x_k \rangle \leqslant f(x)$, for $x = x_*$, we obtain

$$f(x_{k+1}) - f(x_*) \leqslant L_{k+1} V(x_*, x_k) - L_{k+1} V(x_*, x_{k+1}) + \delta_{k+1},$$

whence, after summation, in view of the convexity of f, we have

$$f(\widehat{x}) - f(x_*) \leqslant \frac{1}{S_N} \sum_{k=0}^{N-1} \frac{f(x_{k+1})}{L_{k+1}} - f(x_*) \leqslant \frac{V(x_*, x_0)}{S_N} + \frac{1}{S_N} \sum_{k=0}^{N-1} \frac{\delta_{k+1}}{L_{k+1}}.$$

2) Since f satisfies (2.1) and (2.2), for sufficiently large L_{k+1} and δ_{k+1} the iteration exit criterion will certainly be satisfied. According to (2.1), for some fixed $L > 0$ and $\delta > 0$, the following inequality holds

$$f(y) \leqslant f(x) + \langle \nabla f(x), y - x \rangle + LV(y,x) + \alpha LV(x,y) + \delta, \quad \forall x, y \in Q.$$

Therefore, for $L_{k+1} \geqslant L$ and taking into account (5.1) we obtain

$$\begin{aligned} f(x_{k+1}) &\leqslant f(x_k) + \langle \nabla f(x_k), x_{k+1} - x_k \rangle + L_{k+1}(V(x_{k+1}, x_k) \\ &\quad + \alpha V(x_k, x_{k+1})) + \delta \\ &\leqslant f(x_k) - L_{k+1}(1 - \alpha)V(x_k, x_{k+1}) + \delta, \end{aligned}$$

whence

$$(6.13) \qquad \alpha f(x_{k+1}) \leqslant \alpha f(x_k) - L_{k+1}\alpha(1 - \alpha)V(x_k, x_{k+1}) + \alpha\delta.$$

Now, in view of (2.1) and taking into account $1 - \alpha \geqslant 0$, the following inequality holds

$$(6.14) \quad \begin{aligned} (1 - \alpha)f(x_{k+1}) &\leqslant (1 - \alpha)f(x_k) + (1 - \alpha)\langle \nabla f(x_k), x_{k+1} - x_k \rangle + \\ &\quad + L_{k+1}(1 - \alpha)(V(x_{k+1}, x_k) + \alpha V(x_k, x_{k+1})) + (1 - \alpha)\delta \end{aligned}$$

and for $\alpha = 0$ we have

$$(6.15) \qquad f(x_{k+1}) \leqslant f(x_k) + \langle \nabla f(x_k), x_{k+1} - x_k \rangle + L_{k+1}V(x_{k+1}, x_k) + \delta.$$

Taking into account (2.2) for $\alpha > 0$, after summing the inequalities (6.13) and (6.14), we have

$$\begin{aligned} f(x_{k+1}) &\leqslant f(x_k) + (1 - \alpha)\langle \nabla f(x_k), x_{k+1} - x_k \rangle + L_{k+1}(1 - \alpha)V(x_{k+1}, x_k) + \delta \\ &\leqslant f(x_k) + \langle \nabla f(x_k), x_{k+1} - x_k \rangle + L_{k+1}V(x_{k+1}, x_k) + \alpha\delta \\ &\leqslant f(x_k) + \langle \nabla f(x_k), x_{k+1} - x_k \rangle + L_{k+1}V(x_{k+1}, x_k) + \delta, \end{aligned}$$

i.e. (6.15) holds for each $\alpha \in [0; 1]$. It means that the iteration exit criterion of the Algorithm 5 will certainly be satisfied for $L_{k+1} \geqslant L$ and $\delta_{k+1} \geqslant \delta$.

3) Let us assume that on the $(k + 1)$-th iteration ($k = 0, 1, \ldots, N - 1$) of the Algorithm 5, the auxiliary problem (5.1) is solved i_{k+1} times. Then

$$2^{i_{k+1}-2} = \frac{L_{k+1}}{L_k} = \frac{\delta_{k+1}}{\delta_k},$$

since at the beginning of each iteration the parameters L_k, δ_k are divided by 2. Therefore,

$$\sum_{k=0}^{N-1} i_{k+1} = 2N + \log_2 \frac{L_N}{L_0}, \quad \log_2 \frac{L_N}{L_0} = \log_2 \frac{\delta_N}{\delta_0}.$$

It is clear that at least one of the inequalities $L_N \leqslant 2L, \delta_N \leqslant 2\delta$ holds, which ends the proof. $\qquad \square$

APPENDIX E. THE PROOF OF THEOREM 5.2

Proof. 1) Analogously of the point 1) of the Theorem's 5.1 proof, we have

$$(6.16) \qquad f(\widehat{x}) - f(x_*) \leqslant \frac{1}{S_N} \sum_{k=0}^{N-1} \frac{f(x_{k+1})}{L_{k+1}} - f(x_*) \leqslant \frac{V(x_*, x_0)}{S_N} + \frac{3\varepsilon}{4}.$$

2) Analogously of the point 2) of the Theorem's 5.1 proof, we conclude that for each (α, L, δ)-relatively smooth function f the criterion for the exit from the iteration is certainly fulfilled for $L_{k+1} \geqslant L$.

3) Due to (6.16) for each $k \geqslant 0$ we have

$$f(\widehat{x}) - f(x_*) \leqslant \frac{R^2}{S_N} + \frac{3\varepsilon}{4}.$$

So, for each (α, L, δ)-relatively smooth function f the exit from the iteration will certainly happen for $L_{k+1} \leqslant 2L$, whence

$$S_N \geqslant \frac{N}{2L} \quad \text{and} \quad \frac{R^2}{S_N} \leqslant \frac{2LR^2}{N}.$$

If we require the condition $\frac{2LR^2}{N} \leqslant \frac{\varepsilon}{4}$, we have that $f(\widehat{x}) - f(x_*) \leqslant \varepsilon$ certainly holds for

$$N \geqslant \frac{8LR^2}{\varepsilon}.$$

If we require that f is a $\left(1, \frac{2M^2}{\varepsilon}, \frac{\varepsilon}{2}\right)$-relatively smooth function, we have that

$$N \geqslant \frac{8LR^2}{\varepsilon} = \frac{16M^2R^2}{\varepsilon^2}.$$

Note, that if f is relatively Lipschitz continuous function then f is a $\left(1, \frac{2M^2}{\varepsilon}, \frac{\varepsilon}{2}\right)$-relatively smooth function. Indeed, we have

$$f(y) - f(x) \leqslant \langle \nabla f(y), y - x \rangle \leqslant M\sqrt{2V(x, y)},$$

i.e.

$$(6.17) \qquad f(y) \leqslant f(x) + M\sqrt{2V(x, y)},$$

and

$$(6.18) \qquad \langle \nabla f(x), x - y \rangle \leqslant M\sqrt{2V(y, x)}.$$

After summing inequalities (6.17) and (6.18), we get, that the following inequalities hold for any $x, y \in Q$:

$$(6.19) \qquad \begin{aligned} f(y) &\leqslant f(x) + \langle \nabla f(x), y - x \rangle + M\left(\sqrt{2V(y, x)} + \sqrt{2V(x, y)}\right) \\ &\leqslant f(x) + \langle \nabla f(x), y - x \rangle + \frac{2M^2}{\varepsilon}\left(V(y, x) + V(x, y)\right) + \frac{\varepsilon}{2}. \end{aligned}$$

$\square$

F. STONYAKIN
Moscow Institute of Physics and Technology and V. I. Vernadsky Crimean Federal University, Russia
   E-mail address: `fedyor@mail.ru`

M. ALKOUSA
Moscow Institute of Physics and Technology and HSE University, Russia
   E-mail address: `mohammad.alkousa@phystech.edu`

A. TITOV
Moscow Institute of Physics and Technology and HSE University, Russia
   E-mail address: `a.a.titov@phystech.edu`

O. SAVCHUK
V. I. Vernadsky Crimean Federal University, Russia
   E-mail address: `oleg.savchuk19@mail.ru`

A. GASNIKOV
Moscow Institute of Physics and Technology and IITP RAS, Moscow and Caucasus Mathematical Center, Adyghe State University, Maikop, Russia
   E-mail address: `avgasnikov@gmail.com`