Mirror Descent and Constrained Online Optimization Problems

Alexander A. Titov[1(✉)], Fedor S. Stonyakin[1,2], Alexander V. Gasnikov[1,3], and Mohammad S. Alkousa[1]

[1] Moscow Institute of Physics and Technologies, Moscow, Russia
{a.a.titov,mohammad.alkousa}@phystech.edu, gasnikov@yandex.ru
[2] V. I. Vernadsky Crimean Federal University, Simferopol, Russia
fedyor@mail.ru
[3] Adyghe State University, Caucasus Mathematical Center, Maykop, Russia

Abstract. We consider the following class of online optimization problems with functional constraints. Assume, that a finite set of convex Lipschitz-continuous non-smooth functionals are given on a closed set of n-dimensional vector space. The problem is to minimize the arithmetic mean of functionals with a convex Lipschitz-continuous non-smooth constraint. In addition, it is allowed to calculate the (sub)gradient of each functional only once. Using some recently proposed adaptive methods of Mirror Descent the method is suggested to solve the mentioned constrained online optimization problem with an optimal estimate of accuracy. For the corresponding non-Euclidean prox-structure, the case of a set of n-dimensional vectors lying on the standard n-dimensional simplex is considered.

Keywords: Online convex optimization
Non-smooth constrained optimization
Adaptive mirror descent · Non-euclidean prox-structure · Unit simplex

1 Introduction

Online convex optimization plays a key role in solving the problems, where statistical information is being updated [12,13]. There are a lot of examples of such problems, concerning internet network, consumer data sets or financial market. Quite a few branches of science also face the above-mentioned problems, for example, machine learning applications [14]. The important example is the decision-making problem [13,15]. Suppose, we are given N experts and range of admissible solutions lie on the unit simplex. Every expert gives his estimates of losses with the possible solution and the problem is to minimize total losses

The research by Alexander A. Titov (Sect. 5) and Fedor S. Stonyakin (Sect. 4) was supported by the Russian Science Foundation according to the research project 18-71-00048. The research by Alexander V. Gasnikov (Sect. 3) was supported by the Russian Foundation for Basic Research according to the research project 18-29- 03071 mk.

Y. Evtushenko et al. (Eds.): OPTIMA 2018, CCIS 974, pp. 64–78, 2019.
https://doi.org/10.1007/978-3-030-10934-9_5

from the point view of all experts (the arithmetic mean). Therefore, in recent years, methods for solving online optimization problems have been actively developed [8–14,16].

In problems of online convex optimization, it is required to minimize the sum (or the arithmetic mean) of several convex Lipschitz functionals f_i $(i = \overline{1,N})$ given on some closed set $Q \subset \mathbb{R}^n$. It should be noted that it is possible to calculate the (sub)gradient $\nabla f_i(x)$ of each functional f_i only once. Our paper is devoted to some optimal methods for the following type of problems

$$\begin{cases} \frac{1}{N}\sum\limits_{i=1}^{N} f_i(x) \to \min\limits_{x\in Q} \\ s.t.\ \ g(x) \le 0 \end{cases} \tag{1}$$

We assume that the functionals f_i and g satisfy the Lipschitz property, i.e. there exists a number $M > 0$, such that

$$|g(x) - g(y)| \le M\|x - y\|, \tag{2}$$

$$|f_i(x) - f_i(y)| \le M\|x - y\| \quad \forall i = \overline{1,N}. \tag{3}$$

We can explain the meaning of such formulation of the problem in the following situation. Suppose that we are engaged in some kind of activity during the fixed number of days. Each day can be productive or non-productive. We want to live out N productive days (not necessarily in a row, there can be some non-productive days within this period) so that the total nerve costs (characterized by $f_i(x)$) would be minimal. Note that we pay nervous expenses only in productive days when we try to do something. In non-productive days we do nothing, our aim is to return to the productive state, but we do not pay any costs. The productivity of the day is determined by the condition $g(x^k) \le \varepsilon$. Let's define index i as the number of the productive day. This day we receive feedback from the outside world in the next form: $\nabla f_i(x^k)$ and using this information we build a strategy for the next day x^{k+1}. In non-productive days, we get information about how far have we gone out of the functional constraint and we try to return to this framework. There is no point in arranging unnecessary non-productive days. Therefore, it is also desirable to minimize the number of non-productive days for a given N. The proposed algorithm provides a small amount of costs simultaneously, ensuring that the number of non-productive days will be no more than $O(N)$.

The optimization problems of non-smooth functionals with constraints attract widespread interest in large-scale optimization and its applications [6,23]. There are various methods of solving this kind of optimization problems. Some examples of these methods are: bundle-level method [19], penalty method [24], Lagrange multipliers method [7]. Among them, Mirror Descent (MD) [1,4,18] is viewed as a simple method for non-smooth convex optimization.

Note that a functional constraint, generally, can be non-smooth. That is why we consider subgradient methods. These methods have a long history starting from the method for deterministic unconstrained problems and Euclidean setting

in [21] and the generalization for constrained problems in [20], where the idea of steps switching between the direction of subgradient of the objective and the direction of subgradient of the constraint was suggested. Non-Euclidean extension usually referred to as Mirror Descent, originated in [17,18] and was later analyzed in [4]. An extension for constrained problems was proposed in [18], see also a recent version in [3].

Usually, the step size and stopping rule for Mirror Descent requires to know the Lipschitz constant of the objective function and constraint, if any. Adaptive stepsizes, which do not require this information, are considered in [5] unconstrained problems, and in [3] for constrained problems. Recently, in [2] optimal algorithms of Mirror Descent for convex programming problems with Lipschitz functional constraints with both adaptive step selection and adaptive stopping criteria were proposed for a number of classes of problems. Also, there were considered some modifications of these methods for the case of problems with many functional constraints in [22]. In [14] authors considered adaptive algorithms for online convex optimization problem with constraints, but with only standard Euclidean prox-structure.

In this paper, we propose adaptive and non-adaptive algorithms for solving the problem (1). Note that we consider arbitrary proximal structure, which seems essential for the problem of experts [10–13]. The paper consists of an Introduction and five main sections. In Sect. 2 we give some basic notation concerning convex optimization problems with functional constraints and online optimization problems. In Sect. 3 we propose a non-adaptive algorithm of Mirror Descent for the considered online optimization problem (1). Section 4 is devoted to an adaptive analog of this method (Algorithm 2).

Also in Sect. 4, by analogy with [22], we propose a modification of Algorithm 2 for problems with several functional constraints (Algorithm 3). It is shown that Algorithms 1, 2 and 3 are optimal accurate to multiplication by constants under the condition of nonnegativity of the regret (see Theorems 1 and 2). In Sect. 5 the condition of negative regret is considered. In this case, we get the optimal quality of estimation by the objective function, but the estimation of the number of non-productive steps is worse than (19). In the last section, we consider some numerical experiments that allow us to compare the work of Algorithms 1, 2, and 3 for certain examples.

Summing up, contributions of this paper are as follows:

- two methods (adaptive and non-adaptive) were proposed to solve the online optimization problem for an arbitrary prox-structure;
- the number of non-productive steps is $O(N)$ in the case of nonnegative regret;
- the number of non-productive steps is $O(N^2)$, but the accuracy by regret is better.

2 Problem Statement and Standard Mirror Descent Basics

Let $(E, ||\cdot||)$ be a normed finite-dimensional vector space and E^* be the conjugate space of E with the norm:

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

where $\langle y, x\rangle$ is the value of the continuous linear functional y at $x \in E$.

Let $Q \subset E$ be a (simple) closed convex set, $d : Q \to \mathbb{R}$ be a distance generating function (d.g.f) which is continuously differentiable and 1-strongly convex w.r.t. the norm $\|\cdot\|$, i.e.

$$\forall x, y \in Q \quad \langle \nabla d(x) - \nabla d(y), x - y\rangle \geq \|x - y\|^2,$$

and assume that $\min\limits_{x\in Q} d(x) = d(0)$. Suppose, we have a constant Θ_0 such that $d(x_*) \leq \Theta_0^2$, where x_* is a solution of (1).

Note that if there is a set of optimal points for (1) $X_* \subset Q$, we may assume that

$$\min_{x_*\in X_*} d(x_*) \leq \Theta_0^2.$$

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

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

Standard proximal setups, i.e. Euclidean, entropy, ℓ_1/ℓ_2, simplex, nuclear norm, spectahedron can be found, e.g. in [5]. Let us define the proximal mapping operator standardly

$$\mathrm{Mirr}_x(p) = \arg\min_{u\in Q}\left\{\langle p, u\rangle + V(x, u)\right\} \text{ for each } x \in Q \text{ and } p \in E^*.$$

We make the simplicity assumption, which means that $\mathrm{Mirr}_x(p)$ is easily computable. There are well-known examples of distance generating function, let us denote ℓ_p norm by $\|x\|_p$, and the unit simplex in $\mathbb{R}^n$ by

$$S_n(1) = \left\{x \in \mathbb{R}^n_+ \mid \sum_{i=1}^n x_i = 1\right\}.$$

Consider two cases:

– if $p = 1$, then

$$d(x) = \ln n + \sum_{k=1}^n x_k \ln x_k, \quad V(x, y) = \sum_{k=1}^n x_k \ln\left(\frac{x_k}{y_k}\right); \tag{4}$$

– if $p = 2$, then $d(x) = \frac{1}{2}\|x\|_2^2$, $V(x, y) = \frac{1}{2}\|x - y\|_2^2$.

Let $Q = \mathbb{B}_p^n(1) = \{x \in \mathbb{R}^n; \|x\|_p \leq 1\}$ be the unit ball with l_p norm. One can note the following: if $p \geq 2$, then it is optimal to choose the l_2-norm and the Euclidean prox-structure.

Define q by $\frac{1}{p} + \frac{1}{q} = 1$ and consider $1 \leq p \leq 2$, then $q \geq 2$. If in this case $q = O(\ln n)$, then it is optimal to choose l_p-norm and prox-structure with distance generating function

$$d(x) = \frac{1}{2(p-1)}\|x\|_p^2.$$

In all these cases $R^2 = \max\limits_{x \in Q} d(x) \geqslant \Theta_0^2$.

For $q > \Omega(\ln n)$, we choose l_a-norm, where

$$a = \frac{2\ln n}{2\ln n - 1}$$

and prox-structure with distance generating function

$$d(x) = \frac{1}{2(a-1)}\|x\|_a^2.$$

In this case

$$R^2 = O(\ln n) \geqslant \Theta_0^2 \quad \text{and} \quad \Theta_0 \leqslant O(\sqrt{\ln n}). \tag{5}$$

Let us remind one well-known statement (see, e.g. [5]).

Lemma 1. *Let $f : Q \to \mathbb{R}$ be a convex subdifferentiable function over the convex set Q and $z = Mirr_y(h\nabla f(y))$ for some $h > 0$, $y, z \in Q$. Then for each $x \in Q$*

$$h\langle \nabla f(y), y - x\rangle \leq \frac{h^2}{2}||\nabla f(y)||_*^2 + V(y,x) - V(z,x). \tag{6}$$

3 Online Optimization for the Case of Non-negative Regret: Non-adaptive Algorithm

Assume that the method produces N productive steps and each step the (sub)gradient of exactly one functional of the objectives is calculated. Denote the number of non-productive steps by N_J. Let's consider the non-adaptive method for the problem (1) with a constant step, which depends on the Lipschitz constant M. As a result, we get a sequence $\{x^k\}_{k \in I}$ (on productive steps), which can be considered as a solution to the problem (1) with accuracy δ (see (7)).

By Lemma 1

$$f_i(x^k) - f_i(x) \leq \frac{h}{2}M^2 + \frac{V(x^k,x)}{h} - \frac{V(x^{k+1},x)}{h} = \frac{\varepsilon}{2} + \frac{V(x^k,x)}{h} - \frac{V(x^{k+1},x)}{h}$$

$$g(x^k) - g(x) \leq \frac{h}{2}M^2 + \frac{V(x^k,x)}{h} - \frac{V(x^{k+1},x)}{h} = \frac{\varepsilon}{2} + \frac{V(x^k,x)}{h} - \frac{V(x^{k+1},x)}{h}$$

Algorithm 1. Constrained Online Optimization: Non-Adaptive Mirror Descent Algorithm

Require: $\varepsilon, N, \Theta_0^2, Q, d(\cdot), x^0$

$i := 1,\ k := 0;$
repeat
  if $g(x^k) \leqslant \varepsilon$ **then**
    $h = \frac{\varepsilon}{M^2};$
    $x^{k+1} := Mirr[x^k](h\nabla f_i(x^k));$
    $i := i + 1;$
    $k := k + 1;$
  else
    $h = \frac{\varepsilon}{M^2};$
    $x^{k+1} := Mirr[x^k](h\nabla g(x^k));$
    $k := k + 1;$
  end if
until $i = N + 1$
Guaranteed accuracy:

$$\delta := \frac{\varepsilon}{2} + \frac{M^2\Theta_0^2}{\varepsilon N} - \frac{\varepsilon N_J}{2N} \tag{7}$$

Taking summation over productive and non-productive steps, we get

$$\sum_{i=1}^{N}(f_i(x^k) - f_i(x^*)) + \sum_{k\in J}(g(x^k) - g(x^*))$$

$$\leq \frac{\varepsilon}{2}(N + N_J) + \frac{1}{h}\sum_{k=0}^{N+N_J-1}\left(V(x^k, x^*) - V(x^{k+1}, x^*)\right)$$

$$= \frac{\varepsilon}{2}(N + N_J) + \frac{M^2}{\varepsilon}\sum_{k=0}^{N+N_J-1}\left(V(x^k, x^*) - V(x^{k+1}, x^*)\right),$$

then

$$\sum_{i=1}^{N}(f_i(x^k) - f_i(x^*)) \leq \frac{\varepsilon}{2}N + \frac{M^2\Theta_0^2}{\varepsilon} - \frac{\varepsilon}{2}N_J \tag{8}$$

and by virtue of (7)

$$\frac{1}{N}\sum_{i=1}^{N} f_i(x^k) - \min_{x\in Q}\frac{1}{N}\sum_{i=1}^{N} f_i(x) \leq \delta. \tag{9}$$

If we assume the nonnegativity of the regret (i.e. the left side in (8)) and

$$\delta \leq \varepsilon = \frac{C}{\sqrt{N}} \text{ for some } C > 0, \tag{10}$$

then we get

$$0 \leq N + \frac{2M^2\Theta_0^2}{\varepsilon^2} - N_J = N + \frac{2M^2\Theta_0^2}{C^2}N - N_J,$$

then

$$N_J \leq N \cdot \left(1 + \frac{2M^2\Theta_0^2}{C^2}\right) \sim O(N).$$

Thus, we have the following result

Theorem 1. *Suppose Algorithm 1 works exactly N productive steps. After the stopping of the Algorithm 1, the following inequality holds:*

$$\frac{1}{N}\sum_{i=1}^{N} f_i(x^k) - \min_{x \in Q} \frac{1}{N}\sum_{i=1}^{N} f_i(x) \leq \delta.$$

For the case (10) *and*

$$\frac{1}{N}\sum_{i=1}^{N} f_i(x^k) - \min_{x \in Q} \frac{1}{N}\sum_{i=1}^{N} f_i(x) \geq 0$$

there will be no more than

$$N \cdot \left(1 + \frac{2M^2\Theta_0^2}{C^2}\right) \sim O(N). \tag{11}$$

non-productive steps.

Remark 1. The estimate (11) is optimal for the considered class of problems [12].

Corollary 1. *If $Q = S_n(1)$ and the corresponding prox-structure is chosen as* (4)*, then by* (5) *the estimate* (11) *modifies into*

$$N_J \leqslant N \cdot \left(1 + \frac{2M^2 \ln n}{C^2}\right).$$

4 Adaptive Mirror Descent for the Case of Non-negative Regret

Now, let us consider the adaptive analog of Algorithm 1 for problem (1). The main feature is a nondecreasing stepsize with consideration of the norm of (sub)gradient of the objective function or the constraints in a particular step. Therefore, the proposed algorithm will work until there are exactly N productive steps. As a result, we get a sequence $\{x^k\}_{k \in I}$ on productive steps, which can be considered as a solution to the problem (1) with accuracy δ (see (12)).

By Lemma 1

$$f_i(x^k) - f_i(x) \leq \frac{h_k}{2}\|\nabla f_i(x^k)\|_*^2 + \frac{V(x^k, x)}{h_k} - \frac{V(x^{k+1}, x)}{h_k}$$

Algorithm 2. Constrained Online Optimization: Adaptive Mirror Descent Algorithm

Require: $\varepsilon, N, \Theta_0^2, Q, d(\cdot), x^0$

$i := 1,\ k := 0;$
repeat
  if $g(x^k) \leqslant \varepsilon$ **then**
    $M_k := \|\nabla f_i(x^k)\|_*;$
    $h_k = \Theta_0 \left(\sum_{t=0}^{k} M_t^2 \right)^{-1/2};$
    $x^{k+1} := Mirr[x^k](h_k \nabla f_i(x^k));$
    $i := i + 1;$
    $k := k + 1;$
  else
    $M_k := \|\nabla g(x^k)\|_*;$
    $h_k = \Theta_0 \left(\sum_{t=0}^{k} M_t^2 \right)^{-1/2};$
    $x^{k+1} := Mirr[x^k](h_k \nabla g(x^k));$
    $k := k + 1;$
  end if
until $i = N + 1$
Guaranteed accuracy:

$$\delta := \frac{2\Theta_0}{N} \left(\sum_{i=0}^{N+N_J-1} M_i^2 \right)^{1/2} - \varepsilon \cdot \frac{N_J}{N}. \tag{12}$$

$$g(x^k) - g(x) \leq \frac{h_k}{2} \|\nabla g(x^k)\|_*^2 + \frac{V(x^k, x)}{h_k} - \frac{V(x^{k+1}, x)}{h_k}$$

Dividing each inequality by h_k and summing up for k from 0 to $N + N_J - 1$, and by using the definition of h_k, we obtain

$$\sum_{k \in I} \big(f(x^k) - f(x_*)\big) + \sum_{k \in J} \big(g(x^k) - g(x_*)\big) \leq \sum_{k=0}^{N+N_J-1} \frac{h_k M_k^2}{2}$$

$$+ \sum_{k=0}^{N+N_J-1} \frac{1}{h_k} \big(V(x^k, x_*) - V(x^{k+1}, x_*)\big) \text{ and}$$

$$\sum_{k=0}^{N+N_J-1} \frac{1}{h_k} \big(V(x^k, x_*) - V(x^{k+1}, x_*)\big) = \frac{1}{h_0} V(x^0, x_*) + \sum_{k=0}^{N+N_J-2} \Big(\frac{1}{h_{k+1}} - \frac{1}{h_k}\Big) V(x^{k+1}, x_*)$$

$$-\frac{1}{h_{N+N_J-1}} V(x^k, x_*) \leq \frac{\Theta_0^2}{h_0} + \Theta_0^2 \sum_{k=0}^{N+N_J-2} \Big(\frac{1}{h_{k+1}} - \frac{1}{h_k}\Big) = \frac{\Theta_0^2}{h_{N+N_J-1}}.$$

Whence, by the definition of step sizes h_k,

$$\sum_{i=1}^{N}\left(f_i(x^k)-f(x_*)\right)+\sum_{k\in J}\left(g(x^k)-g(x_*)\right)\le\sum_{k=0}^{N+N_J-1}\frac{h_kM_k^2}{2}+\frac{\Theta_0^2}{h_{N+N_J-1}}$$
$$\le\sum_{k=0}^{N+N_J-1}\frac{\Theta_0}{2}\frac{M_k^2}{\left(\sum_{j=0}^{k}M_j^2\right)^{1/2}}+\Theta_0\left(\sum_{k=0}^{N+N_J-1}M_k^2\right)^{1/2}\le 2\Theta_0\left(\sum_{k=0}^{N+N_J-1}M_k^2\right)^{1/2} \tag{13}$$

where we used inequality

$$\sum_{i=0}^{N+N_J-1}\frac{M_i^2}{\left(\sum_{j=0}^{i}M_j^2\right)^{1/2}}\le 2\left(\sum_{i=0}^{N+N_J-1}M_i^2\right)^{1/2},$$

which can be proved by induction. Since, for $k\in J$, $g(x^k)-g(x_*)\ge g(x^k)>\varepsilon$, we get

$$\sum_{i=1}^{N}(f_i(x^k)-f_i(x^*))<\varepsilon N-\varepsilon(N+N_J)+2\Theta_0\left(\sum_{i=0}^{N+N_J-1}M_i^2\right)^{1/2}. \tag{14}$$

and by (12)

$$\frac{1}{N}\sum_{i=1}^{N}f_i(x^k)-\min_{x\in Q}\frac{1}{N}\sum_{i=1}^{N}f_i(x)\le\delta. \tag{15}$$

If we assume the nonnegativity of the regret (i.e. the left side in (14)) and the accuracy is given by (10), one can get

$$\varepsilon(N+N_J)\le\varepsilon N+2\Theta_0\left(\sum_{i=0}^{N+N_J-1}M_i^2\right)^{1/2}\le\varepsilon N+2M\Theta_0\cdot\sqrt{N+N_J},$$

$$N_J^2\le\frac{4M^2\Theta_0^2(N+N_J)}{\varepsilon^2}=\frac{4M^2\Theta_0^2(N+N_J)N}{C^2}$$

Further,

$$\frac{N_J^2}{N^2+NN_J}=\frac{\left(\frac{N_J}{N}\right)^2}{1+\frac{N_J}{N}}\le\frac{4M^2\Theta_0^2}{C^2}$$

and $N_J=O(N)$. Thus, we have come to the following result.

Theorem 2. *Suppose Algorithm 2 works exactly N productive steps. After the stopping of the Algorithm 2, the following inequality holds:*

$$\frac{1}{N}\sum_{i=1}^{N}f_i(x^k)-\min_{x\in Q}\frac{1}{N}\sum_{i=1}^{N}f_i(x)\le\delta.$$

For the case of (10) *and*

$$\frac{1}{N}\sum_{i=1}^{N} f_i(x^k) - \min_{x\in Q}\frac{1}{N}\sum_{i=1}^{N} f_i(x) \geq 0$$

there will be no more than $O(N)$ *non-productive steps.*

Remark 2. Algorithm 2 is optimal for the considered class of problems [12].

Remark 3. Let's consider a modification of the proposed Algorithm 2 for the case of a set of functional constraints $g_m : Q \to \mathbb{R}$ $(m = \overline{1,K})$. We assume, that all the functionals g_m satisfy the Lipschitz condition:

$$|g_m(x) - g_m(y)| \leq M||x-y|| \;\forall x, y \in Q,\; m = \overline{1,K}. \tag{16}$$

In this case, instead of a set of convex functional constraints $\{g_m(\cdot)\}_{m=1}^{K}$ we can consider one constraint, given as $g : Q \to \mathbb{R}$, where

$$g(x) = \max_{m=\overline{1,K}} g_m(x), \quad |g(x) - g(y)| \leq M||x-y|| \;\forall x, y \in Q.$$

This method will be also optimal, but in practice it can give better accuracy (see Remark 4 below).

5 The Case of Negative Regret

Now we consider the situation when after the stopping of any of the above algorithms, it turns out that the regret is negative. In this case the following inequality

$$\frac{1}{N}\sum_{i=1}^{N} f_i(x^k) - \min_{x\in Q}\frac{1}{N}\sum_{i=1}^{N} f_i(x) \leq 0 \tag{18}$$

holds. It is already impossible to justify the optimality of the number of non-productive steps in view of the right-hand side of inequality (18).

Note that the set of productive steps is not empty, because for arbitrary p steps when the inequality

$$\sum_{k=1}^{p}\frac{1}{M_k^2} \geq \frac{2\Theta_0^2}{\varepsilon^2}$$

is satisfied, one of these p steps will necessarily be productive (see [2,22]). If all the other $p-1$ steps are non-productive (without loss of generality let the last step be productive), then

$$\sum_{k=1}^{p-1}\frac{1}{M_k^2} < \frac{2\Theta_0^2}{\varepsilon^2}$$

and

$$p - 1 < \frac{2M^2\Theta_0^2}{\varepsilon^2}.$$

Algorithm 3. Online Optimization: Adaptive Mirror Descent Algorithm Modification for the Case of Many Constraints

Require: $\varepsilon, N, \Theta_0^2, Q, d(\cdot), x^0$

$i := 1,\ k := 0;$
repeat
  if $g(x^k) \leqslant \varepsilon$ **then**
    $M_k := \|\nabla f_i(x^k)\|_*;$
    $h_k = \Theta_0 \left(\sum\limits_{t=0}^{k} M_t^2 \right)^{-1/2};$
    $x^{k+1} := Mirr[x^k](h_k \nabla f_i(x^k));$
    $i := i + 1;$
    $k := k + 1;$
  else
    $M_k := \|\nabla g_{m(k)}(x^k)\|_*$ for some $g_{m(k)}(\cdot)$: $g_{m(k)}(x^k) > \varepsilon$
    $h_k = \Theta_0 \left(\sum\limits_{t=0}^{k} M_t^2 \right)^{-1/2};$
    $x^{k+1} := Mirr[x^k](h_k \nabla g_{m(k)}(x^k));$
    $k := k + 1;$
  end if
until $i = N + 1$
Guaranteed accuracy:

$$\delta := \frac{2\Theta_0}{N} \left(\sum_{i=0}^{N+N_J-1} M_i^2 \right)^{1/2} - \varepsilon \cdot \frac{N_J}{N}. \tag{17}$$

It is clear, that running the method for a sufficiently long time, it is possible to achieve N productive steps. At the same time between each two successive productive steps there will be no more than $\frac{2M^2\Theta_0^2}{\varepsilon^2}$ non-productive steps, i.e. the number of all non-productive steps will be no more than

$$\frac{2M^2\Theta_0^2}{\varepsilon^2} N.$$

In comparison with the previous items, for $\varepsilon = \frac{C}{\sqrt{N}}$ there will be no more than

$$\frac{2M^2\Theta_0^2}{\varepsilon^2} N = O(N^2) \tag{19}$$

non-productive steps.

6 Numerical Experiments

To compare of Algorithms 1, 2 and 3, some numerical tests were carried out. Consider four different examples with objective function

$$f(x) = \frac{1}{N} \sum_{i=1}^{N} |\langle a_i, x \rangle - b_i| .$$

For the coefficients a_i and constants b_i for $i = 1, \ldots, N$, with different values of N. Let $A \in \mathbb{R}^{N\times 11}$ be a matrix with entries drawn from different random distributions. Then a_i^T are rows in the matrix $A' \in \mathbb{R}^{N\times 10}$, which is introduced from A, by eliminating the last column, and b_i are the entries of the last column in the matrix A. In details, entries of A drawn

- In example 1, from a normal distribution with mean (center) equalling 0 and standard deviation (width) equalling 1.
- In example 2, from a uniform distribution over $[0, 1)$.
- In example 3, from the standard exponential distribution with a scale parameter of 1.
- In example 4, from a Gumbel distribution with the location of the mode equalling 1 and the scale parameter equalling 2.

For the function of constraints $g(x) = \max\limits_{i\in\overline{1,m}} g_i(x)$, we take $m = 3$ and the functionals $g_i(x) = \langle \alpha_i, x\rangle$, where α_i^T are the rows of the matrix

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\ 1 & 2 & 4 & 6 & 8 & 10 & 12 & 14 & 16 & 18 \end{pmatrix}$$

We choose standard Euclidean proximal setup as a prox-function, starting point $x^0 = \dfrac{(1, 1, \ldots, 1)}{\sqrt{10}}$, $\Theta_0 = 3$, $\varepsilon = \frac{1}{\sqrt{N}}$ and

$$Q = \{x = (x_1, x_2, \ldots, x_{10}) \in \mathbb{R}^{10} \,|\, x_1^2 + x_2^2 + \ldots + x_{10}^2 \leq 1\}.$$

The results of the work of Algorithms 1, 2 and 3 are represented in Tables 1, 2 and 3 below, respectively, demonstrate the comparison between these algorithms. The number of non-productive steps are denoted by *nonprod.*, time is given in seconds and parts of the second, δ is guaranteed accuracy of the solution approximation found (sequence $\{x^k\}_{k\in I}$ on productive steps).

All experiments were implemented in Python 3.4, on computer fitted with Intel(R) Core(TM) i7-8550U CPU @ 1.80 GHz, 1992 Mhz, 4 Core(s), 8 Logical Processor(s). RAM of the computer is 8 GB.

From Tables 1 and 2 one can see, that the adaptive Algorithm 2 always works better than non-adaptive Algorithm 1. It is clearly shown in all the examples by

Table 1. Results of Algorithm 1.

	nonprod.	time	δ
ex. 1, $N = 3000$	7041	00.444	187.473
ex. 2, $N = 6000$	12645	00.812	132.565
ex. 3, $N = 7000$	15814	00.958	122.730
ex. 4, $N = 10000$	24971	01.523	102.682

Table 2. Results of Algorithm 2.

	nonprod.	time	δ
ex. 1, $N = 3000$	39	00.149	0.426
ex. 2, $N = 6000$	2821	00.404	0.223
ex. 3, $N = 7000$	5543	00.586	0.405
ex. 4, $N = 10000$	12576	01.104	0.692

the number of non-productive steps, running time of the algorithms and guaranteed accuracy δ. Where the number of non-productive steps and δ produced by Algorithm 2 is very small compared to the Algorithm 1.

From Table 3, we can see, that there is a difference between the number of non-productive steps produced by Algorithms 2 and 3, but the guaranteed accuracy δ and the running time produced by Algorithm 3 is smaller compared to Algorithm 2.

Table 3. Results of Algorithm 3.

	nonprod.	time	δ
ex. 1, $N = 3000$	47	00.121	0.414
ex. 2, $N = 6000$	2835	00.333	0.220
ex. 3, $N = 7000$	5563	00.454	0.394
ex. 4, $N = 10000$	12885	00.807	0.680

Remark 4. To show the advantages of Algorithm 3, as compared to Algorithm 2, one additional numerical test was carried out. Let's now take the functionals of constraints g_i, $i = 1, 2, 3$ as follows

$$g_1(x) = \sum_{i=1}^{10} i \cdot x_i + 1, \quad g_2(x) = \sum_{i=1}^{10} 10i \cdot x_i, \quad g_3(x) = \sum_{i=1}^{10} 50i \cdot x_i.$$

with the same all previous parameters: starting point $x^0 = \dfrac{(1, 1, \ldots, 1)}{\sqrt{10}}$, $\Theta_0 = 3$,

$$Q = \{x = (x_1, x_2, \ldots, x_{10}) \in \mathbb{R}^{10} \,|\, x_1^2 + x_2^2 + \ldots + x_{10}^2 \leq 1\},$$

but with $\varepsilon = 0.5$. Table 4 below demonstrate the comparison between Algorithms 2 and 3, for the objective function $f(x) = \frac{1}{3} \sum_{i=1}^{3} f_i(x)$, where

$$f_1(x) = \sqrt{\sum_{i=1}^{9} (x_i + x_{i+1})^2}, \; f_2(x) = \sqrt{0.1 \left(\sum_{i=1}^{10} x_i^2 + \sum_{i=1}^{9} x_i x_{i+1} \right)}, \; f_3(x) = \sqrt{\sum_{i=1}^{10} x_i^2}.$$

Table 4. Results of Algorithms 2 and 3.

ex. 5, $N = 3$	nonprod.	time	δ
Algorithm 2	1	00.044	1961.954
Algorithm 3	2	00.030	9.608

From Table 4, one can see, that Algorithm 3 works better than Algorithm 2, since the difference between the non-productive steps is very small, equalling only one, and the guaranteed accuracy δ produced by Algorithm 3 is very small compared to the precision produced by Algorithm 2.

References

1. Bayandina, A., Gasnikov, A., Gasnikova, E., Matsievsky, S.: Primal-dual mirror descent for the stochastic programming problems with functional constraints. Comput. Math. Math. Phys. (2018, accepted). https://arxiv.org/pdf/1604.08194.pdf. (in Russian)
2. Bayandina, A., Dvurechensky, P., Gasnikov, A., Stonyakin, F., Titov, A.: Mirror descent and convex optimization problems with non-smooth inequality constraints. In: Large-Scale and Distributed Optimization, pp. 181–213. Springer, Cham (2018)
3. Beck, A., Ben-Tal, A., Guttmann-Beck, N., Tetruashvili, L.: The comirror algorithm for solving nonsmooth constrained convex problems. Oper. Res. Lett. **38**(6), 493–498 (2010)
4. Beck, A., Teboulle, M.: Mirror descent and nonlinear projected subgradient methods for convex optimization. Oper. Res. Lett. **31**(3), 167–175 (2003)
5. Ben-Tal, A., Nemirovski, A.: Lectures on Modern Convex Optimization. Society for Industrial and Applied Mathematics, Philadelphia (2001)
6. Ben-Tal, A., Nemirovski, A.: Robust Truss Topology Design via semidefinite programming. SIAM J. Optim. **7**(4), 991–1016 (1997)
7. Boyd, S., Vandenberghe, L.: Convex Optimization. Cambridge University Press, New York (2004)
8. Bubeck, S., Eldan, R.: Multi-Scale Exploration of Convex Functions and Bandit Convex Optimization. e-print (2015). http://research.microsoft.com/en-us/um/people/sebubeck/ConvexBandits.pdf
9. Bubeck, S., Cesa-Bianchi, N.: Regret analysis of stochastic and nonstochastic multi-armed bandit problems. Found. Trends Mach. Learn. **5**(1), 1–122 (2012)
10. Gasnikov, A.V., Lagunovskaya, A.A., Morozova, L.E.: On the relationship between simulation logit dynamics in the population game theory and a mirror descent method in online optimization using the example of the shortest path problem. Proc. MIPT **7**(4), 104–113 (2015). (in Russian)
11. Gasnikov, A.V., Lagunovskaya, A.A., Usmanova, I.N., Fedorenko, F.A., Krymova, E.A.: Stochastic online optimization. Single-point and multi-point non-linear multi-armed bandits. Convex and strongly-convex case. Autom. Remote Control **78**(2), 224–234 (2017)
12. Hazan, E., Kale, S.: Beyond the regret minimization barrier: optimal algorithms for stochastic strongly-convex optimization. JMLR **15**, 2489–2512 (2014)

13. Hazan, E.: Introduction to online convex optimization. Found. Trends Optim. **2**(3–4), 157–325 (2015)
14. Jenatton, R., Huang, J., Archambeau, C.: Adaptive Algorithms for Online Convex Optimization with Long-term Constraints (2015). https://arxiv.org/abs/1512.07422
15. Kalai, A., Vempala, S.: Efficient algorithms for online decision problems. J. Comput. Syst. Sci. **71**, 291–307 (2005)
16. Lugosi, G., Cesa-Bianchi, N.: Prediction, Learning and Games. Cambridge University Press, New York (2006)
17. Nemirovskii, A.: Efficient methods for large-scale convex optimization problems. Ekonomika i Matematicheskie Metody (1979). (in Russian)
18. Nemirovsky, A., Yudin, D.: Problem Complexity and Method Efficiency in Optimization. Wiley, New York (1983)
19. Nesterov, Y.: Introductory Lectures on Convex Optimization: A Basic Course. Kluwer Academic Publishers, Massachusetts (2004)
20. Polyak, B.: A general method of solving extremum problems. Sov. Math. Dokl. **8**(3), 593–597 (1967). (in Russian)
21. Shor, N.Z.: Generalized gradient descent with application to block programming. Kibernetika **3**(3), 53–55 (1967). (in Russian)
22. Stonyakin, F.S., Alkousa, M.S., Stepanov, A.N., Barinov, M.A.: Adaptive mirror descent algorithms in convex programming problems with Lipschitz constraints. Trudy Instituta Matematiki i Mekhaniki URO RAN **24**(2), 266–279 (2018)
23. Shpirko, S., Nesterov, Y.: Primal-dual subgradient methods for huge-scale linear conic problem. SIAM J. Optim. **24**(3), 1444–1457 (2014)
24. Vasilyev, F.: Optimization Methods. Fizmatlit, Moscow (2002). (in Russian)