Weierstraß-Institut
für Angewandte Analysis und Stochastik
Leibniz-Institut im Forschungsverbund Berlin e. V.

Preprint ISSN 2198-5855

Advances in low-memory subgradient optimization

Pavel Dvurechensky[1], Alexander Gasnikov[2], Evgeni Nurminski[3], Fedor Stonyakin [2]

submitted: January 20, 2020

[1] Weierstrass Institute
Mohrenstr. 39
10117 Berlin
Germany
E-Mail: pavel.dvurechensky@wias-berlin.de

[2] Moscow Institute of Physics and Technology
Institutskiy Pereulok, 9
Dolgoprudny, Moscow Region
141701 Russian Federation
E-Mail: gasnikov@yandex.ru
fedyor@mail.ru

[3] Far Eastern Federal University
8 Sukhanova St.
Russky ostrov, Vladivostok
690090 Russian Federation
E-Mail: nurminskiy.ea@dvfu.ru

No. 2676
Berlin 2020

2010 *Mathematics Subject Classification.* 90C30, 90C25, 68Q25, 65K15.

Key words and phrases. Convex optimization, composite optimization, universal method, mirror prox, acceleration, nonsmooth optimization.

Edited by
Weierstraß-Institut für Angewandte Analysis und Stochastik (WIAS)
Leibniz-Institut im Forschungsverbund Berlin e. V.
Mohrenstraße 39
10117 Berlin
Germany

Fax: +49 30 20372-303
E-Mail: preprint@wias-berlin.de
World Wide Web: http://www.wias-berlin.de/

Advances in low-memory subgradient optimization

Pavel Dvurechensky, Alexander Gasnikov, Evgeni Nurminski, Fedor Stonyakin

Abstract

One of the main goals in the development of non-smooth optimization is to cope with high dimensional problems by decomposition, duality or Lagrangian relaxation which greatly reduces the number of variables at the cost of worsening differentiability of objective or constraints. Small or medium dimensionality of resulting non-smooth problems allows to use bundle-type algorithms to achieve higher rates of convergence and obtain higher accuracy, which of course came at the cost of additional memory requirements, typically of the order of n^2, where n is the number of variables of non-smooth problem. However with the rapid development of more and more sophisticated models in industry, economy, finance, et all such memory requirements are becoming too hard to satisfy. It raised the interest in subgradient-based low-memory algorithms and later developments in this area significantly improved over their early variants still preserving $O(n)$ memory requirements. To review these developments this chapter is devoted to the black-box subgradient algorithms with the minimal requirements for the storage of auxiliary results, which are necessary to execute these algorithms. To provide historical perspective this survey starts with the original result of N.Z. Shor which opened this field with the application to the classical transportation problem. The theoretical complexity bounds for smooth and non-smooth convex and quasi-convex optimization problems are briefly exposed in what follows to introduce to the relevant fundamentals of non-smooth optimization. Special attention in this section is given to the adaptive step-size policy which aims to attain lowest complexity bounds. Unfortunately the non-differentiability of objective function in convex optimization essentially slows down the theoretical low bounds for the rate of convergence in subgradient optimization compared to the smooth case but there are different modern techniques that allow to solve non-smooth convex optimization problems faster then dictate lower complexity bounds. In this work the particular attention is given to Nesterov smoothing technique, Nesterov Universal approach, and Legendre (saddle point) representation approach. The new results on Universal Mirror Prox algorithms represent the original parts of the survey. To demonstrate application of non-smooth convex optimization algorithms for solution of huge-scale extremal problems we consider convex optimization problems with non-smooth functional constraints and propose two adaptive Mirror Descent methods. The first method is of primal-dual variety and proved to be optimal in terms of lower oracle bounds for the class of Lipschitz-continuous convex objective and constraints. The advantages of application of this method to sparse Truss Topology Design problem are discussed in certain details. The second method can be applied for solution of convex and quasi-convex optimization problems and is optimal in a sense of complexity bounds. The conclusion part of the survey contains the important references that characterize recent developments of non-smooth convex optimization.

Introduction

We consider a finite-dimensional non-differentiable convex optimization problem (COP)

$$\min_{x\in E} f(x) = f_\star = f(\vec{x}^\star), \vec{x}^\star \in X_\star, \tag{1}$$

where E denotes a finite-dimensional space of primal variables and $f : E \to \mathbb{R}$ is a finite convex function, not necessarily differentiable. For a given point $\vec{x}$ the subgradient oracul returns value of objective function at that point $f(\vec{x})$ and subgradient $g \in \partial f(\vec{x})$. We do not make any assumption about the choice of g from $\partial f(\vec{x})$. As we are interested in computational issues related to solving (1) mainly we assume that this problem is solvable and has nonempty and bounded set of solutions $X_\star$.

This problem enjoys a considerable popularity due to its important theoretical properties and numerous applications in large-scale structured optimization, discrete optimization, exact penalization in constrained optimization, and others. Non-smooth optimization theory made it possible to solve in an efficient way classical discrete min-max problems [23], l_1-approximation and others, at the same time opening new approaches in bi-level, monotropic programming, two-stage stochastic optimization, to name a few.

As a major steps in:the development of different algorithmic ideas we can start with the subgradient algorithm due to Shor (see [71] for the overview and references to earliest works).

1 Example Application: Transportation Problem and The First Subgradient Algorithm

From utilitarian point of view the development of non-smooth (convex) optimization started with the classical transportation problem

$$\begin{gathered} \min \textstyle\sum_{i=1}^m \sum_{j=1}^n c_{ij}x_{ij} \\ \textstyle\sum_{i=1}^m x_{ij} = a_j,\ j = 1,2,\dots,n; \\ \textstyle\sum_{j=1}^n x_{ij} = b_i,\ i = 1,2,\dots,m \\ x_{ij} \geq 0, i = 1,2,\dots,m; j = 1,2,\dots,n \end{gathered} \tag{2}$$

which is widely used in many applications.

By dualizing this problem with respect to balancing constrains we can convert (2) into dual problem of the kind

$$\max\ \Phi(\vec{u},\vec{v}) \tag{3}$$

where $\vec{u} = (u_i, i = 1,2,\dots,m); \vec{v} = (v_j, j = 1,2,\dots,n)$ are dual variables associated with the balancing constraints in (2) and $\Phi(\vec{u},\vec{v})$ is the dual function defined as

$$\Phi(\vec{u},\vec{v}) = \inf_{\vec{x}\geq 0} L(\vec{x},\vec{u},\vec{v}) \tag{4}$$

and $L(\vec{x},\vec{u},\vec{v})$ is the Lagrange function of the problem:

$$L(\vec{x},\vec{u},\vec{v}) = \sum_{i=1}^m \sum_{j=1}^n c_{ij}x_{ij} + \sum_{j=1}^n u_j(\sum_{i=1}^m x_{ij} - a_j) + \sum_{i=1}^m v_i(\sum_{j=1}^n x_{ij} - b_i).$$

By rearranging terms in this expression we can obtain the following expression for the dual function

$$\begin{gathered} \Phi(\vec{u},\vec{v}) = -m\textstyle\sum_{j=1}^n u_j a_j - n\sum_{i=1}^m v_i b_i + \sum_{i=1}^m \sum_{j=1}^n \inf_{\vec{x}\geq 0} x_{ij}\{c_{ij} + u_j + v_i\} = \\ -m\textstyle\sum_{j=1}^n u_j a_j - n\sum_{i=1}^m v_i b_i - \mathrm{Ind}_D(\vec{u},\vec{v}), \end{gathered} \tag{5}$$

where

$$\mathrm{Ind}_D(\vec{u},\vec{v}) = \begin{cases} 0 & \text{when } c_{ij} + u_i + v_j \geq 0; \\ \infty & \text{otherwise.} \end{cases} \tag{6}$$

is the indicator function of the set $D = \{\vec{u}, \vec{v} : c_{ij} + u_j + v_i \geq 0, i = 1, 2, \ldots, m; j = 1, 2, \ldots, n\}$ which is the feasible set of the dual problem.

Of course, by explicitely writing feasibility constraints for (3) we obtain the linear dual transportation problem with a fewer variables but with much higher number of constraints. This is bad news for textbook simplex method so many specialized algorithms were developed, one of them was simple-minded method of generalized gradient which started the development of non-smooth optimization.

This method relies on subgradient of concave function $\Phi(\vec{u}, \vec{v})$ which we can transform into convex just by changing signs and replacing $\inf$ with $\sup$

$$\Phi(\vec{u}, \vec{v}) = m \sum_{j=1}^{n} u_j a_j + n \sum_{i=1}^{m} v_i b_i + \\ \sum_{i=1}^{m} \sum_{j=1}^{n} \sup_{\vec{x} \geq 0} x_{ij} \{c_{ij} + u_j + v_i\} = \\ = m \sum_{j=1}^{n} u_j a_j + n \sum_{i=1}^{m} v_i b_i + \mathrm{Ind}_D(\vec{u}, \vec{v}),$$

and ask for its *minimization*.

According to convex analysis [65] the subdifferential $\partial_c \Phi(\vec{u}, \vec{v})$ exists for any $\vec{v}, \vec{u} \in \mathrm{int}\,\mathrm{dom}(\mathrm{Ind}_D)$, and in this case just equals to the (constant) vector $g_L = (\vec{g}_{\vec{u}}, \vec{g}_{\vec{v}}) = (\vec{a}, \vec{b})$ of a linear objective in the interior of D. The situation becomes more complicated when $\vec{u}, bv$ happens to be at the boundary of D, the subdifferential set ceases to be a singleton and becomes even unbounded, roughly speaking certain linear manifolds are added to g_L but we will not go into details here. The difficulty is that if we mimic gradient method of the kind

$$\vec{u}^{k+1} = \vec{u}^k - \lambda g_L^u = \vec{u}^k - \lambda \vec{a}; \vec{v}^{k+1} = \vec{v}^k - \lambda g_L^v = \vec{v}^k - \lambda \vec{b}; k = 0, 1, \ldots \quad (7)$$

with a certain step-size $\lambda > 0$, we inevitably violate the dual feasibility constraints as $\vec{a}, \vec{b} > 0$. Than the dual function (7) becomes undefined and correspondently the subdifferential set becomes undefined as well.

There are at least two simple ways to overcome this difficulty. One is to incorporate in the gradient method certain operations which restore feasibility and the appropriate candidate for it is the orthogonal projection operation where one can make use of the special structure of constraints and sparsity. However it will still require computing projection operator of the kind $B^T(BBT)^{-1}B$ for basis matrices B with rather uncertain number of iteration and of matrices of the size around $(n+m) \times (n+m)$. Neither computers speed nor memory sizes at that time where not up to demands to solve problems of $n + m \approx 10^4$ which was required by GOSPLAN!

The second ingenious way was to take into account that if $\sum_{j=1}^{n} a_j = \sum_{i=1}^{m} b_i = V$, which is required anyway for solvability of transportation problem in a closed form. The flow variables may be uniformally bounded by V and the dual function can be redefined as

$$\Phi_V(\vec{u}, \vec{v}) = m \sum_{j=1}^{n} u_j a_j + n \sum_{i=1}^{m} v_i b_i - \\ \sum_{i=1}^{m} \sum_{j=1}^{n} \max_{0 \leq \vec{x} \leq V} x_{ij} \{c_{ij} + u_j + v_i\} = \\ = m \sum_{j=1}^{n} u_j a_j + n \sum_{i=1}^{m} v_i b_i + P_V(\vec{u}, \vec{v})$$

where the penalty function $P_V(\vec{u}, \vec{v})$ is easily computed by component-wise maximization:

$$P_V(\vec{u}, \vec{v}) = \sum_{i=1}^{m} \sum_{j=1}^{n} \max_{x_{ij} \in [0,V]} x_{ij} \{c_{ij} + u_j + v_i\} = \\ \sum_{i=1}^{m} \sum_{j=1}^{n} V \{c_{ij} + u_j + v_i\}_+$$

where $\{\cdot\}_+ = \max\{0, \cdot\}$. Than the dual objective function becomes finite, the optimization problem — unconstrained and we can use simple subgradient method with very low requirements for memory and computations.

Actually even tighter bounds $\vec{x}_{ij} \leq \min(a_i, b_j)$ can be imposed on the flow variables which may be advantageous for computational reasons.

In both cases there is a fundamental problem of recovering optimal primal $n \times m$ primal solution from $n+m$ dual. This problem was studied by many authors and recent advances in this area can be studied from the excellent paper by A. Nedic and A. Ozdoglar [46]. Theoretically speaking, nonzero positive values of $c_{ij} + u^\star_j + v^\star_i$, where $\vec{u}^\star, \vec{v}^\star$ are the *exact* optimal solutions of the dual problem (3) signal that the corresponding optimal primal flow $x^\star_{ij}$ is equal to zero. Hopefully after excluding these variables we obtain nondegenerate basis and can compute the remaining variables by simple and efficient linear algebra, especially taking into account the uni-modularity of basis.

However the theoretical gap between zeros and non-zeros is exponentially small even for modest length integer data therefore we need an accuracy unattainable by modern 64-128 bits hardware. Also the real life computations are always accompanied by numerical noise and we face the hard choice in fact guessing which dual constraints are active and which are not.

To connect the transportation problem with non-smooth optimization notice that the penalty function $P_V(\vec{u}, \vec{v})$ is finite with the subdifferential $\partial_c P_V(\vec{u}, \vec{v})$ which can be represented as a set of $n \times m$ matrices

$$\vec{g}_{ij} = \begin{cases} V & \text{if } c_{ij} + u_j + v_i > 0 \\ 0 & \text{if } c_{ij} + u_j + v_i < 0 \\ \mathrm{cone}(0, V) & \text{if } c_{ij} + u_j + v_i = 0 \end{cases}$$

so the subdifferential set is a convex hull of up to $2^{(n+m)}$ extreme points — enormous number even for a modest size transportation problem. Nevertheless it is easy to get at least single member of subdifferential and consider the simplest version of subgradient method:

$$\vec{x}^{k+1} = \vec{x}^k - \lambda \bar{\vec{g}}^k, k = 0, 1, \ldots$$

where $\vec{x}^0$ is a given starting point, $\lambda > 0$ — fixed step-size and $\bar{\vec{g}}^k = \vec{g}^k / \|\vec{g}^k\|$ is a normalized subgradient $\vec{g}^k \in \partial f(\vec{x}^k)$. Of course we assume that $\vec{g}^k \neq 0$ otherwise $\vec{x}^k$ is already a solution.

Of course, there is no hope of classical convergence result such that $\vec{x}^k \to \vec{x}^\star \in X_\star$, but the remarkable theorem of Shor [68] establishes that this simplest algorithm determines at least the approximate solution. As a major step in the development of different algorithmic ideas we can start with the subgradient algorithm due to Shor (see [71] for the overview and references to earliest works). Of course, there is no hope of classical convergence result such that $\vec{x}^k \to \vec{x}^\star \in X_\star$, but the remarkable theorem of Shor [68] establishes that this very simple algorithm provides an approximate solution of (1) at least theoretically.

Theorem 1. *Let f is a finite convex function with a subdifferential ∂f and the sequence $\{\vec{x}^k\}$ is obtained by the recursive rule*

$$x^{k+1} = \vec{x}^k - \lambda g^k_\nu, k = 0, 1, \ldots \tag{8}$$

with $\lambda > 0$ and $g^k_\nu = g^k / \|g^k\|, g^k \in \partial f(\vec{x}^k)$, $g^k \neq 0$ is a normalized subgradient at the point x^k. Then for any $\varepsilon > 0$ there is an infinite set $Z_\varepsilon \subset Z$ such that for any $k \in Z_\varepsilon$

$$f(\tilde{\vec{x}}^k) = f(\vec{x}^k) \text{ and } \mathrm{dist}(\tilde{\vec{x}}^k, X_\star) \leq \lambda(1+\varepsilon)/2.$$

The statement of the theorem is illustrated on Fig. 1 together with the idea of the proof. The detailed proof of the theorem goes like following: Let $\vec{x}^\star \in X_\star$ and estimate

$$\|\vec{x}^{k+1} - \vec{x}^\star\|^2 = \|\vec{x}^k - \vec{x}^\star - \lambda \vec{g}^k_\nu\|^2 = \|\vec{x}^k - \vec{x}^\star\|^2 + \lambda^2 - 2\lambda \bar{\vec{g}}^k(\vec{x}^k - \vec{x}^\star).$$

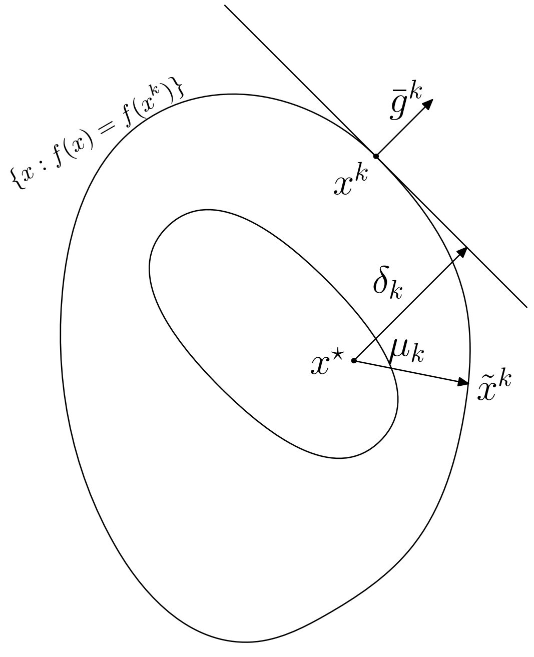

Figure 1: The statement and the idea of the proof of Shor theorem

The last term in fact equals

$$\min_{\vec{z} \in H_k} \|\vec{x}^\star - \vec{z}\|^2 = \|\vec{x}^\star - \vec{z}^k\|^2 = \delta_k,$$

where $H_k = \{\vec{z} : \vec{z} g_\nu^k = \vec{x}^k g_\nu^k$ is a hyperplane, orthogonal to g_ν^k and passing through the point $\vec{x}^k$, so

$$\|\vec{x}^{k+1} - \vec{x}^\star\|^2 = \|\vec{x}^k - \vec{x}^\star\|^2 + \lambda^2 - 2\lambda\delta_k,\ k = 0,1,2,\dots \tag{9}$$

If $\lambda^2 - 2\lambda\delta_k \leq -\lambda^2\varepsilon$ for any $k \in Z$ then

$$\|\vec{x}^{k+1} - \vec{x}^\star\|^2 \leq \|\vec{x}^k - \vec{x}^\star\|^2 - \lambda^2\varepsilon,\ k = 0,1,2,\dots \tag{10}$$

therefore

$$0 \leq \|\vec{x}^{k+1} - \vec{x}^\star\|^2 \leq \|\vec{x}^0 - \vec{x}^\star\|^2 \leq -k\lambda^2\varepsilon \to -\infty \tag{11}$$

when $k \to \infty$. This contradiction proves that there is k_0 such that $\lambda^2 - 2\lambda\delta_{k_0} > -\lambda^2\varepsilon$ or $\delta_{k_0} < \lambda(1+\varepsilon)/2$.

To complete the proof notice that by convexity $f(z^{k_0}) \geq f(\vec{x}_{k_0})$ and therefore

$$\min_{z: f(z) = f(\vec{x}^{k_0})} \|\vec{x}^\star - z\|^2 = \|\vec{x}^\star - \vec{z}^{k_0}\|^2 = \min_{z: f(z) \geq f(x^{k_0})} \|\vec{x}^\star - z\|^2 \leq \|\vec{x}^\star - z^{k_0}\|^2 = \delta_{k_0}. \tag{12}$$

By setting $\vec{\tilde{x}}^0 = z^{k_0}$ we obtain $\|\vec{x}^\star - \vec{\tilde{x}}^0\|^2 < \lambda(1+\varepsilon)/2$.

By replacing x^0 in (11) by $\vec{\tilde{x}}^0$ and repeating the reasoning above we obtain $\tilde{x}^1$ such that $\|\vec{x}^\star - \vec{\tilde{x}}^1\|^2 < \lambda(1+\varepsilon)/2$, then in the same manner $\vec{\tilde{x}}^2, \vec{\tilde{x}}^3$ and so on with $\|\vec{x}^\star - \vec{\tilde{x}}^k\|^2 < \lambda(1+\varepsilon)/2, k = 2,3,\dots$ which complete the proof. ■

2 Complexity Results for Convex Optimization

At this section we describe the complexity results for non-smooth convex optimization problems. Most of the results mentioned below can be found in books [50, 64, 60, 15, 9]. We start with the **'small dimensional problems'**, when

$$N \geq n = \dim \vec{x},$$

where N is a number of oracle calls (number of subgradient calculations or/and calculations of separation hyperplane to some simple set at a given point).

Let's consider convex optimization problem

$$f(\vec{x}) \to \min_{\vec{x} \in Q}, \tag{13}$$

where Q – is a compact and simple set (it's significant here!). Based on at least N subgradient calculations (in general, oracle calls) we would like to find such a point $\vec{x}^N$ that

$$f\left(\vec{x}^N\right) - f_* \leq \varepsilon,$$

where $f_* = f(\vec{x}_*)$ is an optimal value of function in (13), $\vec{x}_*$ – the solution of (13). The lower and the upper bounds for the oracle complexity is (up to a multiplier, which has logarithmic dependence on some characteristic of the set Q)

$$N \sim n \ln\left(\Delta f / \varepsilon\right),$$

where $\Delta f = \sup\limits_{\vec{x}, \vec{y} \in Q} \{f(\vec{y}) - f(\vec{x})\}$. The center of gravity method [45, 62] converges according to this estimate. The center of gravity method in $n = 1$ is a simple binary search method [12]. But in $n > 1$ this method is hard to implement. The complexity of iteration is too high, because we required center of gravity oracle [15]. Well known ellipsoid method [69, 50] requires[1] $N = \tilde{O}\left(n^2 \ln\left(\Delta f / \varepsilon\right)\right)$ oracle calls and $O\left(n^2\right)$ iteration complexity. In [76, 15] a special version of cutting plane method was proposed. This method (Vayda's method) requires $N = \tilde{O}\left(n \ln\left(\Delta f / \varepsilon\right)\right)$ oracle calls and has iteration complexity $\tilde{\mathrm{O}}\left(n^{2.37}\right)$. In the work [44] there proposed a method with $N = \tilde{O}\left(n \ln\left(\Delta f / \varepsilon\right)\right)$ oracle calls and iteration complexity $\tilde{O}\left(n^2\right)$. Unfortunately, for the moment it's not obvious that this method is very practical one due to the large log-factors in $\tilde{O}()$.

Based on ellipsoid method in the late 70-th Leonid Khachyan showed [40] that LP is in P in byte complexity. Let us shortly explain the idea. The main question is whether $A\vec{x} \leq \vec{b}$ is solvable or not, where $n = \dim \vec{x}$, $m = \dim \vec{b}$ and all elements of A and $\vec{b}$ are integers. We would like also to find one of the exact solutions $\vec{x}_*$. This problem up to a logarithmic factor in complexity is equivalent to the problem to find the exact solution of LP problem $\langle \vec{c}, \vec{x} \rangle \to \min\limits_{A\vec{x} \leq \vec{b}}$ with integer A, $\vec{b}$ and $\vec{c}$. We consider only inequality constraints as it is known that to find the exact solution of $A\vec{x} = \vec{b}$ one can use polynomial Gauss elimination algorithm with $O\left(n^3\right)$ arithmetic operations (a.o.) complexity.

Let us introduce

$$\Lambda = \sum_{i,j=1,1}^{m,n} \log_2 \left|a_{ij}\right| + \sum_{i=1}^{m} \log_2 \left|\vec{b}_i\right| + \log_2(mn) + 1.$$

If $A\vec{x} \leq \vec{b}$ is compatible, then there exists such $\vec{x}_*$ that $\|\vec{x}_*\|_\infty \leq 2^\Lambda$, $A\vec{x}_* \leq \vec{b}$ otherwise

$$\min_{\vec{x}} \left\|(A\vec{x} - \vec{b})_+\right\|_\infty \geq 2^{-(\Lambda - 1)}.$$

[1] Here and below for all (large) n: $\tilde{O}(g(n)) \leq C \cdot (\ln n)^r g(n)$ with some constants $C > 0$ and $r \geq 0$. Typically, $r = 1$. If $r = 0$, then $\tilde{O}(\cdot) = O(\cdot)$.

Thus, the question of compatibility of $A\vec{x} \le \vec{b}$ is equivalent to the problem of finding minimum of the following non-smooth convex optimization problem

$$\left\|(A\vec{x}-\vec{b})_+\right\|_\infty \to \min_{\|\vec{x}_*\|_\infty \le 2^\Lambda}.$$

The approach of [40] is to apply ellipsoid method for this problem with $\varepsilon = 2^{-\Lambda}$. From the complexity of this method, it follows that in $O(n\Lambda)$-bit arithmetic with $\tilde{O}\left(mn+n^2\right)$ cost of PC memory one can find $\vec{x}_*$ (if it exists) in $\tilde{O}\left(n^3\left(n^2+m\right)\Lambda\right)$ a.o.

Note, that in the ideal arithmetic with real numbers it is still an open question [10] whether it is possible to find the exact solution of LP an problem (with the data given by real numbers) in polynomial time in the ideal arithmetic ($\pi \cdot e$ – costs $O(1)$).

Now let us consider **'large dimensional problems'**

$$N \le n = \dim \vec{x}.$$

Table 1 describes (for more details see [9, 15, 60]) optimal estimates for the number of oracle calls for convex optimization problem (13) in the case when $N \le n$. Now Q is not necessarily compact set.

Table 1: Optimal estimates for the number of oracle calls

$N \le n$	$\lvert f(\vec{y})-f(\vec{x})\rvert \le M\|\vec{y}-\vec{x}\|$	$\|\nabla f(\vec{y})-\nabla f(\vec{x})\|_* \le L\|\vec{y}-\vec{x}\|$
$f(\vec{x})$ convex	$O\left(\frac{M^2R^2}{\varepsilon^2}\right)$	$O\left(\sqrt{\frac{LR^2}{\varepsilon}}\right)$
$f(\vec{x})$ $\mu-$strongly convex in $\|\cdot\|$-norm	$\tilde{O}\left(\frac{M^2}{\mu\varepsilon}\right)$	$\tilde{O}\left(\sqrt{\frac{L}{\mu}}\left\lceil \ln\left(\frac{\mu R^2}{\varepsilon}\right)\right\rceil\right)$ $(\forall N)$

Here R is a "distance" (up to a $\ln n$-factor) between starting point and the nearest solution

$$R = \tilde{O}\left(\|\vec{x}^0 - \vec{x}_*\|\right).$$

Let's describe optimal method in the most simple case: $Q = \mathbb{R}^n$, $\|\cdot\| = \|\cdot\|_2$ [64, 54]. Define

$$B_2^n(\vec{x}_*, R) = \left\{\vec{x} \in \mathbb{R}^n : \ \|\vec{x}-\vec{x}_*\|_2 \le R\right\}.$$

The main iterative process is (for simplicity we'll denote arbitrary element of $\partial f(\vec{x})$ as $\nabla f(\vec{x})$)

$$\vec{x}^{k+1} = \vec{x}^k - h\nabla f\left(\vec{x}^k\right). \tag{14}$$

Assume that under $\vec{x} \in B_2^n\left(\vec{x}_*, \sqrt{2}R\right)$

$$\|\nabla f(\vec{x})\|_2 \le M, \tag{15}$$

where $R = \left\|\vec{x}^0 - \vec{x}_*\right\|_2$.

Hence, from (14), (15) we have

$$\left\|\vec{x}-\vec{x}^{k+1}\right\|_2^2 = \left\|\vec{x}-\vec{x}^k + h\nabla f\left(\vec{x}^k\right)\right\|_2^2 =$$

$$= \left\|\vec{x}-\vec{x}^k\right\|_2^2 + 2h\left\langle \nabla f\left(\vec{x}^k\right), \vec{x}-\vec{x}^k\right\rangle + h^2\left\|\nabla f\left(\vec{x}^k\right)\right\|_2^2 \leq$$

$$\leq \left\|\vec{x}-\vec{x}^k\right\|_2^2 + 2h\left\langle \nabla f\left(\vec{x}^k\right), \vec{x}-\vec{x}^k\right\rangle + h^2 M^2. \tag{16}$$

Here we choose $\vec{x}=\vec{x}_*$ (if $\vec{x}_*$ isn't unique, we choose the nearest $\vec{x}_*$ to $\vec{x}^0$)

$$f\left(\frac{1}{N}\sum_{k=0}^{N-1}\vec{x}^k\right) - f_* \leq \frac{1}{N}\sum_{k=0}^{N-1} f\left(\vec{x}^k\right) - f\left(\vec{x}_*\right) \leq \frac{1}{N}\sum_{k=0}^{N-1}\left\langle \nabla f\left(\vec{x}^k\right), \vec{x}^k - \vec{x}_*\right\rangle \leq$$

$$\leq \frac{1}{2hN}\sum_{k=0}^{N-1}\left\{\left\|\vec{x}_* - \vec{x}^k\right\|_2^2 - \left\|\vec{x}_* - \vec{x}^{k+1}\right\|_2^2\right\} + \frac{hM^2}{2} =$$

$$= \frac{1}{2hN}\left(\|\vec{x}_* - \vec{x}^0\|_2^2 - \|\vec{x}_* - \vec{x}^N\|_2^2\right) + \frac{hM^2}{2}.$$

If

$$h = \frac{R}{M\sqrt{N}}, \quad \bar{\vec{x}}^N = \frac{1}{N}\sum_{k=0}^{N-1}\vec{x}^k, \tag{17}$$

then

$$f\left(\bar{\vec{x}}^N\right) - f_* \leq \frac{MR}{\sqrt{N}}. \tag{18}$$

Note that the precise lower bound for fixed steps first-order methods for the class of convex optimization problems with (15) [25]

$$f\left(\vec{x}^N\right) - f_* \geq \frac{MR}{\sqrt{N+1}}.$$

Inequality (18) means that (see also Table 1)

$$N = \frac{M^2R^2}{\varepsilon^2}, \quad h = \frac{\varepsilon}{M^2}.$$

So, one can mentioned that if we will use in (14)

$$\vec{x}^{k+1} = \vec{x}^k - h_k\nabla f\left(\vec{x}^k\right), \quad h_k = \frac{\varepsilon}{\left\|\nabla f\left(\vec{x}^k\right)\right\|_2^2} \tag{19}$$

the result (18) holds with [54]

$$\bar{\vec{x}}^N = \frac{1}{\sum_{k=0}^{N-1} h_k}\sum_{k=0}^{N-1} h_k\vec{x}^k.$$

If we put in (19),

$$h_k = \frac{R}{\left\|\nabla f\left(\vec{x}^k\right)\right\|_2\sqrt{N}},$$

like in (17), the result similar to (18) also holds

$$\min_{k=0,\ldots,N-1} f\left(\vec{x}^k\right) - f_* \leq \frac{MR}{\sqrt{N}}$$

not only for the convex functions, but also for quasi-convex functions [13, 52]:

$$f(\alpha\vec{x}+(1-\alpha)\vec{y}) \leq \max\{f(\vec{x}),f(\vec{y})\} \text{ for all } \vec{x},\vec{y} \in Q, \alpha \in [0,1].$$

Note that

$$0 \leq \frac{1}{2hk}\left(\left\|\vec{x}_* - \vec{x}^0\right\|_2^2 - \left\|\vec{x}_* - \vec{x}^k\right\|_2^2\right) + \frac{hM^2}{2}.$$

Hence, for all $k = 0,...,N$,

$$\left\|\vec{x}_* - \vec{x}^k\right\|_2^2 \leq \left\|\vec{x}_* - \vec{x}^0\right\|_2^2 + h^2M^2k \leq 2\left\|\vec{x}_* - \vec{x}^0\right\|_2^2,$$

therefore

$$\left\|\vec{x}^k - \vec{x}_*\right\|_2 \leq \sqrt{2}\left\|\vec{x}^0 - \vec{x}_*\right\|_2, \quad k = 0,...,N. \tag{20}$$

Inequality (20) justifies that we need assumption (15) holds only with $\vec{x} \in B_2^n\left(\vec{x}_*, \sqrt{2}R\right)$.

For the general (constrained) case (13) we introduce a norm $\|\cdot\|$ and some prox-function $d(\vec{x}) \geq 0$, which is continuous and 1-strongly convex with respect to $\|\cdot\|$, i.e. $d(y) - d(x) - \langle d(x), y - x\rangle \geq \frac{1}{2}\|x-y\|^2$, for all $x, y \in Q$. We also introduce Bregman's divergence [9]

$$V[\vec{x}](\vec{y}) = d(\vec{y}) - d(\vec{x}) - \langle \nabla d(\vec{x}), \vec{y} - \vec{x}\rangle.$$

We set $R^2 = V[\vec{x}^0](\vec{x}_*)$, where $\vec{x}_*$ – is solution of (13) (if $\vec{x}_*$ isn't unique then we assume that $\vec{x}_*$ is minimized $V[\vec{x}^0](\vec{x}_*)$. The natural generalization of iteration process (14) is Mirror Descent algorithm [48, 9] which iterates as

$$\vec{x}^{k+1} = \mathrm{Mirr}_{\vec{x}^k}\left(h\nabla f\left(\vec{x}^k\right)\right), \quad \mathrm{Mirr}_{\vec{x}^k}(\mathrm{v}) = \arg\min_{\vec{x}\in Q}\left\{\left\langle \mathrm{v}, \vec{x} - \vec{x}^k\right\rangle + V[\vec{x}^k](\vec{x})\right\}.$$

For this iteration process instead of (16) we have

$$2V[\vec{x}^{k+1}](\vec{x}) \leq 2V[\vec{x}^k](\vec{x}) + 2h\left\langle \nabla f\left(\vec{x}^k\right), \vec{x} - \vec{x}^k\right\rangle + h^2M^2,$$

where $\|\nabla f(\vec{x})\|_* \leq M$ for all $x : V[\vec{x}](\vec{x}_*) \leq 2V[\vec{x}^0](\vec{x}_*) = 2R^2$, see also Section 4.

Analogues of formulas (17), (18), (20) are also valid

$$f\left(\bar{\vec{x}}^N\right) - f_* \leq \frac{\sqrt{2}MR}{\sqrt{N}},$$

where

$$\bar{\vec{x}}^N = \frac{1}{N}\sum_{k=0}^{N-1}\vec{x}^k, \quad h = \frac{\varepsilon}{M^2}$$

and

$$\left\|\vec{x}^k - \vec{x}_*\right\| \leq 2R, \quad k = 0,...,N.$$

In [9] authors discus how to choose $d(\vec{x})$ for different simple convex sets Q. One of these examples (unit simplex) will considered below. Note, that in all these examples one can guarantees that [9]:

$$R \leq C\sqrt{\ln n}\cdot\left\|\vec{x}_* - \vec{x}^0\right\|.$$

Note, that if $Q=\mathbb{R}^n$, $\|\cdot\|=\|\cdot\|_2$ then $d(\vec{x})=\frac{1}{2}\|\vec{x}\|_2^2$, $V[\vec{x}](\vec{y})=\frac{1}{2}\|\vec{y}-\vec{x}\|_2^2$,

$$\vec{x}^{k+1}=\text{Mirr}_{\vec{x}^k}\left(h\nabla f\left(\vec{x}^k\right)\right)=\arg\min_{\vec{x}\in\mathbb{R}^n}\left\{h\left\langle\nabla f\left(\vec{x}^k\right),\vec{x}-\vec{x}^k\right\rangle+\frac{1}{2}\|\vec{x}-\vec{x}^k\|_2^2\right\}=$$
$$=\vec{x}^k-h\nabla f\left(\vec{x}^k\right),$$

that corresponds to the standard gradient-type iteration process (14).

Example (unit simplex). *We have*

$$Q=S_n(1)=\left\{\vec{x}\in\mathrm{R}_+^n:\ \sum_{i=1}^n\vec{x}_i=1\right\},\quad\|\nabla f(\vec{x})\|_\infty\le M_\infty,\quad\vec{x}\in Q,$$

$$\|\cdot\|=\|\cdot\|_1,\quad d(\vec{x})=\ln n+\sum_{i=1}^n\vec{x}_i\ln\vec{x}_i,\quad h=M_\infty^{-1}\sqrt{2\ln n/N},\quad\vec{x}_i^0=1/n,\quad i=1,...,n.$$

For $k=0,...,N-1$, $i=1,...,n$

$$\vec{x}_i^{k+1}=\frac{\exp\left(-h\sum\limits_{r=1}^k\nabla_i f(\vec{x}^r)\right)}{\sum\limits_{l=1}^n\exp\left(-h\sum\limits_{r=1}^k\nabla_l f(\vec{x}^r)\right)}=\frac{\vec{x}_i^k\exp\left(-h\nabla_i f\left(\vec{x}^k\right)\right)}{\sum\limits_{l=1}^n\vec{x}_l^k\exp\left(-h\nabla_l f\left(\vec{x}^k\right)\right)}.$$

The main result here is

$$f\left(\bar{\vec{x}}^N\right)-f_*\le M_\infty\sqrt{\frac{2\ln n}{N}},\quad\bar{\vec{x}}^N=\frac{1}{N}\sum_{k=0}^{N-1}\vec{x}^k.$$

Note, that if we use $\|\cdot\|_2$-norm and $d(\vec{x})=\frac{1}{2}\left\|\vec{x}-\vec{x}^0\right\|_2^2$ here, we will have higher iteration complexity (2-norm projections on unit simplex) and

$$f\left(\bar{\vec{x}}^N\right)-f_*\le\frac{M_2}{\sqrt{N}},\quad\|\nabla f(\vec{x})\|_2\le M_2,\quad\vec{x}\in Q.$$

Since typically $M_2=\mathrm{O}\left(\sqrt{n}M_\infty\right)$, it is worth to use $\|\cdot\|_1$-norm.

Assume now that $f(x)$ in (13) is additionally μ-strongly convex in $\|\cdot\|_2$ norm:

$$f(\vec{y})\ge f(\vec{x})+\langle\nabla f(\vec{x}),y-x\rangle+\frac{\mu}{2}\|\vec{y}-\vec{x}\|_2^2\text{ for all }\vec{x},\vec{y}\in Q.$$

Let

$$\vec{x}^{k+1}\text{= Mirr}_{\vec{x}^k}\left(h_k\nabla f\left(\vec{x}^k\right)\right)=\arg\min_{\vec{x}\in Q}\left\{h_k\left\langle\nabla f\left(\vec{x}^k\right),\vec{x}-\vec{x}^k\right\rangle+\frac{1}{2}\left\|\vec{x}-\vec{x}^k\right\|_2^2\right\},$$

where

$$h_k=\frac{2}{\mu\cdot(k+1)},\quad d(\vec{x})=\frac{1}{2}\left\|\vec{x}-\vec{x}^0\right\|_2^2,\quad\|\nabla f(\vec{x})\|_2\le M,\quad\vec{x}\in Q.$$

Then [67]

$$f\left(\sum_{k=1}^N\frac{2k}{k(k+1)}\vec{x}^k\right)-f_*\le\frac{2M^2}{\mu\cdot(k+1)}.$$

Hence (see also Table 1),

$$N\simeq\frac{2M^2}{\mu\varepsilon}.$$

This bound is also un-improvable up to a constant factor [50, 60].

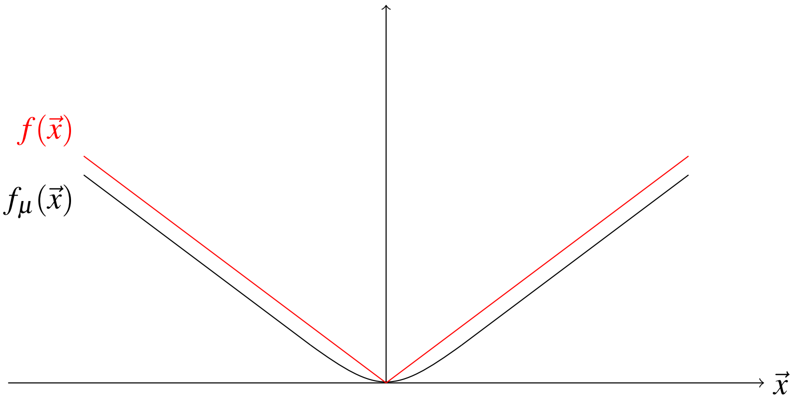

Figure 2: Function $f_\mu(\vec{x})$ is a smooth approximation to non-smooth function $f(\vec{x})$.

3 Looking into the Black-Box

In this section we consider how problem special structure can be used to solve non-smooth optimization problems with the convergence rate $O\left(\frac{1}{k}\right)$, which is faster than the lover bound $O\left(\frac{1}{\sqrt{k}}\right)$ for general class of non-smooth convex problems [50]. Nevertheless, there is no contradiction as additional structure is used and we are looking inside the black-box.

3.1 Nesterov's smoothing

In this subsection, following [53], we consider the problem

$$\min_{\vec{x}\in Q_1\subset E_1}\{f(\vec{x})=h(\vec{x})+\max_{\vec{u}\in Q_2\subset E_2}\{\langle A\vec{x},\vec{u}\rangle-\phi(\vec{u})\}\}, \tag{21}$$

where $A: E_1\to E_2^*$ is a linear operator, $\phi(\vec{u})$ is a continuous convex function on Q_2, Q_1, Q_2 are convex compacts, h is convex function with L_h-Lipschitz-continuous gradient.

Let us consider an example of $f(x)=\|A\vec{x}-\vec{b}\|_\infty$ with $A\in\mathbb{R}^{m\times n}$. Then,

$$f(x)=\max_{\vec{u}\in\mathbb{R}^m}\left\{\langle\vec{u},A\vec{x}-\vec{b}\rangle:\|\vec{u}\|_1\le 1\right\},$$

$h=0$, $E_2=\mathbb{R}^m$, $\phi(\vec{u})=\langle\vec{u},\vec{b}\rangle$ and Q_2 is the ball in 1-norm.

The main idea of Nesterov is to add regularization inside the definition of f in (21). More precisely, a prox-function $d_2(\vec{u})$ (see definition in Section 2) is introduced for the set Q_2 and a smoothed counterpart $f_\mu(\vec{x})$ for f is defined as

$$f_\mu(\vec{x})=h(\vec{x})+\max_{\vec{u}\in Q_2}\{\langle A\vec{x},\vec{u}\rangle-\phi(\vec{u})-\mu d_2(\vec{u})\}$$

and $\vec{u}_\mu(\vec{x})$ is the optimal solution of this maximization problem.

Theorem 2 ([53])**.** *The function $f_\mu(\vec{x})$ is well defined, convex and continuously differentiable at any $\vec{x}\in E_1$ with $\nabla f_\mu(\vec{x})=\nabla h(\vec{x})+A^*\vec{u}_\mu(\vec{x})$. Moreover, $\nabla f_\mu(\vec{x})$ is Lipschitz continuous with constant $L_\mu=L_h+\frac{\|A\|_{1,2}^2}{\mu}$.*

Here the adjoint operator A^* is defined by equality $\langle A\vec{x},\vec{u}\rangle = \langle A^*\vec{u},\vec{x}\rangle$, $\vec{x}\in E_1,\vec{u}\in E_2$ and the norm of the operator $\|A\|_{1,2}$ is defined by $\|A\|_{1,2} = \max_{\vec{x},\vec{u}}\{\langle A\vec{x},\vec{u}\rangle : \|\vec{x}\|_{E_1} = 1, \|\vec{u}\|_{E_2} = 1\}$.

Since Q_2 is bounded, $f_\mu(\vec{x})$ is a uniform approximation for the function f, namely, for all $\vec{x}\in Q_1$,

$$f_\mu(\vec{x}) \le f(\vec{x}) \le f_\mu(\vec{x}) + \mu D_2, \tag{22}$$

where $D_2 = \max\{d_2(\vec{u}) : \vec{u}\in Q_2\}$.

Then, the idea is to choose μ sufficiently small and apply accelerated gradient method to minimize $f_\mu(\vec{x})$ on Q_1. We use accelerated gradient method from [33, 32] which is different from the original method of [53].

Algorithm 1 Accelerated Gradient Method

```
Input: Objective $f(\vec{x})$, feasible set $Q$, Lipschitz constant $L$ of the $\nabla f(\vec{x})$, starting point $\vec{x}^0 \in Q$„ prox-setup: $d(\vec{x})$ – 1-strongly convex w.r.t. $\|\cdot\|_{E_1}$, $V[\vec{z}](\vec{x}) := d(\vec{x}) - d(\vec{z}) - \langle\nabla d(\vec{z}),\vec{x}-\vec{z}\rangle$.
Set $k = 0, C_0 = \alpha_0 = 0, \vec{y}^0 = \vec{z}^0 = \vec{x}^0$.
for $k = 0,1,\dots$ do
    Find $\alpha^{k+1}$ as the largest root of the equation
        $$C_{k+1} := C_k + \alpha_{k+1} = L\alpha_{k+1}^2. \tag{23}$$
    $$\vec{x}^{k+1} = \frac{\alpha_{k+1}\vec{z}^k + C_k\vec{y}^k}{C_{k+1}}. \tag{24}$$
    $$\vec{z}^{k+1} = \arg\min_{\vec{x}\in Q}\{V[\vec{z}^k](\vec{x}) + \alpha_{k+1}(f(\vec{x}^{k+1}) + \langle\nabla f(\vec{x}^{k+1}),\vec{x}-\vec{x}^{k+1}\rangle)\}. \tag{25}$$
    $$\vec{y}^{k+1} = \frac{\alpha_{k+1}\vec{z}^{k+1} + C_k\vec{y}^k}{C_{k+1}}. \tag{26}$$
    Set $k = k+1$.
end for
Output: The point $\vec{y}^{k+1}$.
```

Theorem 3 ([33, 32])**.** *Let the sequences $\{\vec{x}^k,\vec{y}^k,\vec{z}^k,\alpha_k,C_k\}$, $k\ge 0$ be generated by Algorithm 1. Then, for all $k\ge 0$, it holds that*

$$f(\vec{y}^k) - f^* \le \frac{4LV[\vec{z}_0](\vec{x}^\star)}{(k+1)^2}. \tag{27}$$

Following the same steps as in the proof of Theorem 3 in [53], we obtain

Theorem 4. *Let Algorithm 1 be applied to minimize $f_\mu(\vec{x})$ on Q_1 with $\mu = \frac{2\|A\|_{1,2}}{N+1}\sqrt{\frac{D_1}{D_2}}$, where $D_1 = \max\{d_1(\vec{x}) : \vec{x}\in Q_1\}$. Then, after N iterations, we have*

$$0 \le f(\vec{y}^N) - f_\star \le \frac{4\|A\|_{1,2}\sqrt{D_1D_2}}{N+1} + \frac{4L_hD_1}{(N+1)^2}. \tag{28}$$

Proof. Applying Theorem 3 to f_μ, and using (22), we obtain

$$\begin{aligned} 0 \le f(\vec{y}^N) - f_\star \le f_\mu(\vec{y}^N) + \mu D_2 - f_\mu(\vec{x}^\star_\mu) &\le \mu D_2 + \frac{4L_\mu D_1}{(N+1)^2} + \frac{4L_h D_1}{(N+1)^2} \\ &= \mu D_2 + \frac{4\|A\|^2_{1,2} D_1}{\mu(N+1)^2} + \frac{4L_h D_1}{(N+1)^2}. \end{aligned}$$

Substituting the value of μ from the theorem statement, we finish the proof. ■ □

A generalization of the smoothing technique for the case of non-compact sets Q_1, Q_2, which is especially interesting when dealing with problems dual to problems with linear constraints, can be found in [72]. Ubiquitous entropic regularization of optimal transport [20] can be seen as a particular case of the application of smoothing technique, especially in the context of Wasserstein barycenters [21, 74, 29].

3.2 Nemirovski's Mirror Prox

In his paper [47], Nemirovski considers problem (21) in the following form

$$\min_{\vec{x}\in Q_1 \subset E_1} \{f(\vec{x}) = h(\vec{x}) + \max_{\vec{u}\in Q_2 \subset E_2} \{\langle A\vec{x}, \vec{u}\rangle + \langle \vec{b}, \vec{u}\rangle\}\}, \tag{29}$$

pointing to the fact that this problem is as general as (21). Indeed, the change of variables $\vec{u} \leftarrow (\vec{u}, t)$ and the feasible set $Q_2 \leftarrow \{(\vec{u}, t) : \min_{\vec{u}' \in Q_2} \phi(\vec{u}') \le t \le \phi(\vec{u})\}$ allows to make ϕ linear. His idea is to consider problem (29) directly as a convex-concave saddle point problem and associated weak variational inequality (VI).

$$\text{Find } \ \vec{z}^\star = (\vec{x}^\star, \vec{u}^\star) \in Q_1 \times Q_2 \ \text{ s.t. } \ \langle \Phi(\vec{z}), \vec{z}^\star - \vec{z}\rangle \le 0 \ \forall \vec{z} \in Q_1 \times Q_2, \tag{30}$$

where the operator

$$\Phi(\vec{z}) = \begin{pmatrix} \nabla h(\vec{x}) + A^* \vec{u} \\ -A\vec{x} - \vec{b} \end{pmatrix} \tag{31}$$

is monotone, i.e. $\langle \Phi(\vec{z}_1) - \Phi(\vec{z}_2), \vec{z}_1 - \vec{z}_2\rangle \ge 0$, and Lipschitz-continuous, i.e. $\|\Phi(\vec{z}_1) - \Phi(\vec{z}_2)\|_* \le L\|\vec{z}_1 - \vec{z}_2\|$. With the appropriate choice of norm on $E_1 \times E_2$ and prox-function for $Q_1 \times Q_2$, see Section 5 in [47], the Lipschitz constant for Φ can be estimated as $L = 2\|A\|_{1,2}\sqrt{D_1 D_2} + L_h D_1$.

Theorem 5 ([47])**.** *Assume that $\Phi(\vec{z})$ is monotone and L-Lipschitz-continuous. Then, for any $k \ge 1$ and any $\vec{u} \in Q$,*

$$\max_{\vec{z}\in Q} \langle \Phi(\vec{z}), \widehat{\vec{w}}^k - \vec{z}\rangle \le \frac{L}{k} \max_{\vec{z}\in Q} V[\vec{z}^0](\vec{z}). \tag{34}$$

Moreover, if the VI is associated with a convex-concave saddle point problem, i.e.

- $E = E_1 \times E_2$,
- $Q = Q_1 \times Q_2$ *with convex compact sets* $Q_1 \subset E_1$, $Q_2 \subset E_2$
- $\Phi(\vec{z}) = \Phi(\vec{x}, \vec{u}) = \begin{pmatrix} \nabla_{\vec{x}} f(\vec{x}, \vec{u}) \\ -\nabla_{\vec{u}} f(\vec{x}, \vec{u}) \end{pmatrix}$ *for a continuously differentiable function $f(\vec{x}, \vec{u})$ which is convex in $\vec{x} \in Q_1$ and concave in $\vec{u} \in Q_2$,*

Algorithm 2 Mirror Prox

```
Input: General VI on a set $Q \subset E$ with operator $\Phi(\vec{z})$, Lipschitz constant $L$ of $\Phi(\vec{z})$, prox-setup: $d(\vec{z})$, $V[\vec{z}](\vec{w})$.
1: Set $k = 0$, $\vec{z}^0 = \arg\min_{\vec{z}\in Q} d(\vec{z})$.
2: for $k = 0, 1, \dots$ do
3:     Calculate
         $\vec{w}^k = \arg\min_{\vec{z}\in Q} \left\{ \langle \Phi(\vec{z}^k), \vec{z} \rangle + LV[\vec{z}^k](\vec{z}) \right\}.$    (32)
4:     Calculate
         $\vec{z}^{k+1} = \arg\min_{\vec{z}\in Q} \left\{ \langle \Phi(\vec{w}^k), \vec{z} \rangle + LV[\vec{z}^k](\vec{z}) \right\}.$    (33)
5:     Set $k = k + 1$.
6: end for
Output: $\widehat{\vec{w}}^k = \frac{1}{k}\sum_{i=0}^{k-1} \vec{w}^i$.
```

then

$$[\max_{\vec{u}\in Q_2} f(\widehat{\vec{x}}^k, \vec{u}) - \min_{\vec{x}\in Q_1}\max_{\vec{u}\in Q_2} f(\vec{x},\vec{u})] + [\min_{\vec{x}\in Q_1}\max_{\vec{u}\in Q_2} f(\vec{x},\vec{u}) - \min_{\vec{x}\in Q_1} f(\vec{x}, \widehat{\vec{u}}^k)] \le \frac{L}{k}\max_{\vec{z}\in Q} V[\vec{z}^0](\vec{z}). \quad (35)$$

Choosing appropriately the norm in the space $E_1 \times E_2$ and applying Mirror Prox algorithm to solve problem (29) as a saddle point problem, we obtain that the saddle point error in the l.h.s. of (35) decays as $\frac{2\|A\|_{1,2}\sqrt{D_1D_2}+L_hD_1}{k}$. This is slightly worse than the rate in (27) since the accelerated gradient method allows the faster decay for the smooth part $h(\vec{x})$. An accelerated Mirror Prox method with the same rate as in (27) can be found in [18].

4 Non-Smooth Optimization in Large Dimensions

The optimization of non-smooth functionals with constraints attracts widespread interest in large-scale optimization and its applications [8, 61]. Subgradient methods for non-smooth optimization have a long history starting with the method for deterministic unconstrained problems and Euclidean setting in [70] and the generalization for constrained problems in [63], 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 [48, 50] and was later analyzed in [6]. An extension for constrained problems was proposed in [50], see also recent version in [5]. To prove faster convergence rate of Mirror Descent for strongly convex objective in an unconstrained case, the restart technique [49, 50, 51] was used in [37]. 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 step-sizes, which do not require this information, are considered in [48] for problems without inequality constraints, and in [5] for constrained problems.

Formally speaking, we consider the following convex constrained minimization problem

$$\min\{f(\vec{x}): \quad \vec{x} \in X \subset E, \quad g(\vec{x}) \le 0\}, \quad (36)$$

where X is a convex closed subset of a finite-dimensional real vector space E, $f : X \to \mathbb{R}$, $g : E \to \mathbb{R}$ are convex functions.

We assume g to be a non-smooth Lipschitz-continuous function and the problem (3) to be regular. The last means that there exists a point $\bar{\vec{x}}$ in relative interior of the set X, such that $g(\bar{\vec{x}}) < 0$.

Note that, despite problem (36) contains only one inequality constraint, considered algorithms allow to solve more general problems with a number of constraints given as $\{g_i(\vec{x}) \leq 0, i = 1,...,m\}$. The reason is that these constraints can be aggregated and represented as an equivalent constraint given by $\{g(\vec{x}) \leq 0\}$, where $g(\vec{x}) = \max_{i=1,...,m} g_i(\vec{x})$.

We consider two adaptive Mirror Descent methods [4] for the problem (36). Both considered methods have complexity $O\left(\frac{1}{\varepsilon^2}\right)$ and optimal.

We consider algorithms, which are based on Mirror Descent method. Thus, we start with the description of proximal setup and basic properties of Mirror Descent step. Let E be a finite-dimensional real vector space and E^* be its dual. We denote the value of a linear function $g \in E^*$ at $\vec{x} \in E$ by $\langle \vec{g}, \vec{x} \rangle$. Let $\|\cdot\|_E$ be some norm on E, $\|\cdot\|_{E,*}$ be its dual, defined by $\|\vec{g}\|_{E,*} = \max\limits_{\vec{x}} \big\{ \langle \vec{g}, \vec{x} \rangle, \|\vec{x}\|_E \leq 1 \big\}$. We use $\nabla f(\vec{x})$ to denote any subgradient of a function f at a point $\vec{x} \in \mathrm{dom} f$.

Given a vector $\vec{x} \in X^0$, and a vector $\vec{p} \in E^*$, the Mirror Descent step is defined as

$$\vec{x}^+ = \mathrm{Mirr}[\vec{x}](\vec{p}) := \arg\min_{\vec{z} \in X} \big\{ \langle \vec{p}, \vec{z} \rangle + V[\vec{x}](\vec{z}) \big\} = \arg\min_{\vec{z} \in X} \big\{ \langle \vec{p}, \vec{z} \rangle + d(\vec{z}) - \langle \nabla d(\vec{x}), \vec{z} \rangle \big\}. \tag{37}$$

We make the simplicity assumption, which means that $\mathrm{Mirr}[\vec{x}](\vec{p})$ is easily computable.

The following lemma [9] describes the main property of the Mirror Descent step.

Lemma 1. *Let f be some convex function over a set X, $h > 0$ be a step-size, $\vec{x} \in X^0$. Let the point $\vec{x}^+$ be defined by $\vec{x}^+ = \mathrm{Mirr}[\vec{x}](h(\nabla f(\vec{x})))$. Then, for any $\vec{z} \in X$,*

$$\begin{aligned} h\big(f(\vec{x}) - f(\vec{z})\big) &\leq h\langle \nabla f(\vec{x}), \vec{x} - \vec{z} \rangle \\ &\leq \frac{h^2}{2} \|\nabla f(\vec{x})\|^2 + V[\vec{x}](\vec{z}) - V[\vec{x}^+](\vec{z}). \end{aligned} \tag{38}$$

The following analog of Lemma 1 for δ-subgradient $\nabla_\delta f$ holds.

Lemma 2. *Let f be some convex function over a set X, $h > 0$ be a step-size, $\vec{x} \in X^0$. Let the point $\vec{x}^+$ be defined by $\vec{x}^+ = \mathrm{Mirr}[\vec{x}](h \cdot (\nabla_\delta f(\vec{x})))$. Then, for any $\vec{z} \in X$,*

$$\begin{aligned} h \cdot \big(f(\vec{x}) - f(\vec{z})\big) &\leq h \cdot \langle \nabla f(\vec{x}), \vec{x} - \vec{z} \rangle + h \cdot \delta \\ &\leq \frac{h^2}{2} \|\nabla_\delta f(\vec{x})\| + h \cdot \delta + V[\vec{x}](\vec{z}) - V[\vec{x}^+](\vec{z}). \end{aligned} \tag{39}$$

We consider problem (36) in two different settings, namely, non-smooth Lipschitz-continuous objective function f and general objective function f, which is not necessarily Lipschitz-continuous, e.g. a quadratic function. In both cases, we assume that g is non-smooth and is Lipschitz-continuous

$$|g(\vec{x}) - g(\vec{y})| \leq M_g \|\vec{x} - \vec{y}\|_E, \quad \vec{x}, \vec{y} \in X. \tag{40}$$

Let $\vec{x}_*$ be a solution to (36). We say that a point $\tilde{\vec{x}} \in X$ is an *ε-solution* to (36) if

$$f(\tilde{\vec{x}}) - f(\vec{x}_*) \leq \varepsilon, \quad g(\tilde{\vec{x}}) \leq \varepsilon. \tag{41}$$

All considered in this section methods (Algorithms 3 and 4) are applicable in the case of using δ-subgradient instead of usual subgradient. For this case we can get an ε-solution $\tilde{\vec{x}} \in X$:

$$f(\tilde{\vec{x}}) - f(\vec{x}_*) \leq \varepsilon + O(\delta), \quad g(\tilde{\vec{x}}) \leq \varepsilon + O(\delta). \tag{42}$$

The methods we describe are based on the of Polyak's switching subgradient method [63] for constrained convex problems, also analyzed in [55], and Mirror Descent method originated in [50]; see also [48].

4.1 Convex Non-Smooth Objective Function

In this subsection, we assume that f is a non-smooth Lipschitz-continuous function

$$|f(\vec{x}) - f(\vec{y})| \leq M_f \|\vec{x} - \vec{y}\|_E, \quad \vec{x}, \vec{y} \in X. \tag{43}$$

Let $\vec{x}_*$ be a solution to (36) and assume that we know a constant $\Theta_0 > 0$ such that

$$d(\vec{x}_*) \leq \Theta_0^2. \tag{44}$$

For example, if X is a compact set, one can choose $\Theta_0^2 = \max_{\vec{x} \in X} d(\vec{x})$.

Algorithm 3 Adaptive Mirror Descent (Non-Smooth Objective)

```
Input: accuracy $\varepsilon > 0$; $\Theta_0$ s.t. $d(\vec{x}_*) \leq \Theta_0^2$.
$\vec{x}^0 = \arg\min_{\vec{x} \in X} d(\vec{x})$.
Initialize the set $I$ as empty set.
Set $k = 0$.
repeat
    if $g(\vec{x}^k) \leq \varepsilon$ then
        $M_k = \|\nabla f(\vec{x}^k)\|_{E,*}$,
        $h_k = \frac{\varepsilon}{M_k^2}$
        $\vec{x}^{k+1} = \mathrm{Mirr}[\vec{x}^k](h_k \nabla f(\vec{x}^k))$ ("productive step")
        Add $k$ to $I$.
    else
        $M_k = \|\nabla g(\vec{x}^k)\|_{E,*}$
        $h_k = \frac{\varepsilon}{M_k^2}$
        $\vec{x}^{k+1} = \mathrm{Mirr}[\vec{x}^k](h_k \nabla g(\vec{x}^k))$ ("non-productive step")
    end if
    Set $k = k + 1$.
until $\sum_{j=0}^{k-1} \frac{1}{M_j^2} \geq \frac{2\Theta_0^2}{\varepsilon^2}$
Output: $\bar{\vec{x}}^k := \frac{\sum_{i \in I} h_i \vec{x}^i}{\sum_{i \in I} h_i}$
```

Theorem 6. *Assume that inequalities* (40) *and* (43) *hold and a known constant* $\Theta_0 > 0$ *is such that* $d(\vec{x}_*) \leq \Theta_0^2$. *Then, Algorithm 3 stops after not more than*

$$k = \left\lceil \frac{2 \max\{M_f^2, M_g^2\} \Theta_0^2}{\varepsilon^2} \right\rceil \tag{45}$$

iterations and $\bar{\vec{x}}^k$ *is an* ε*-solution to* (36) *in the sense of* (41).

Let us now show that Algorithm 3 allows to reconstruct an approximate solution to the problem, which is dual to (36). We consider a special type of problem (36) with g given by

$$g(\vec{x}) = \max_{i\in\{1,...,m\}} \left\{g_i(\vec{x})\right\}. \tag{46}$$

Then, the dual problem to (36) is

$$\varphi(\lambda) = \min_{\vec{x}\in X}\left\{f(\vec{x}) + \sum_{i=1}^{m}\lambda_i g_i(\vec{x})\right\} \to \max_{\lambda_i\geq 0, i=1,...,m} \varphi(\lambda), \tag{47}$$

where $\lambda_i \geq 0, i = 1,...,m$ are Lagrange multipliers.

We slightly modify the assumption (44) and assume that the set X is bounded and that we know a constant $\Theta_0 > 0$ such that

$$\max_{\vec{x}\in X} d(\vec{x}) \leq \Theta_0^2.$$

As before, denote $[k] = \{j \in \{0,...,k-1\}\}$, $J = [k]\setminus I$. Let $j \in J$. Then a subgradient of $g(\vec{x})$ is used to make the j-th step of Algorithm 3. To find this subgradient, it is natural to find an active constraint $i \in 1,...,m$ such that $g(\vec{x}^j) = g_i(\vec{x}^j)$ and use $\nabla g(\vec{x}^j) = \nabla g_i(\vec{x}^j)$ to make a step. Denote $i(j) \in 1,...,m$ the number of active constraint, whose subgradient is used to make a non-productive step at iteration $j \in J$. In other words, $g(\vec{x}^j) = g_{i(j)}(\vec{x}^j)$ and $\nabla g(\vec{x}^j) = \nabla g_{i(j)}(\vec{x}^j)$. We define an approximate dual solution on a step $k \geq 0$ as

$$\bar{\lambda}_i^k = \frac{1}{\sum\limits_{j\in I} h_j} \sum_{j\in J, i(j)=i} h_j, \quad i \in \{1,...,m\}. \tag{48}$$

and modify Algorithm 3 to return a pair $(\bar{\vec{x}}^k, \bar{\lambda}^k)$.

Theorem 7. *Assume that the set X is bounded, the inequalities* (40) *and* (43) *hold and a known constant $\Theta_0 > 0$ is such that $d(\vec{x}_*) \leq \Theta_0^2$. Then, modified Algorithm 3 stops after not more than*

$$k = \left\lceil \frac{2\max\{M_f^2, M_g^2\}\Theta_0^2}{\varepsilon^2} \right\rceil$$

iterations and the pair $(\bar{\vec{x}}^k, \bar{\lambda}^k)$ returned by this algorithm satisfies

$$f(\bar{\vec{x}}^k) - \varphi(\bar{\lambda}^k) \leq \varepsilon, \quad g(\bar{\vec{x}}^k) \leq \varepsilon. \tag{49}$$

Now we consider an interesting example of huge-scale problem [57, 61] with a sparse structure. We would like to illustrate two important ideas. Firstly, consideration of the dual problem can simplify the solution, if it is possible to reconstruct the solution of the primal problem by solving the dual problem. Secondly, for a special sparse non-smooth piece-wise linear functions we suggest a very efficient implementation of one subgradient iteration [57]. In such cases simple subgradient methods (for example, Algorithm 3) can be useful due to the relatively inexpensive cost of iterations.

Recall (see e.g. [61]) that Truss Topology Design problem consists in finding the best mechanical structure resisting to an external force with an upper bound for the total weight of construction. Its mathematical formulation looks as follows:

$$\min_{w\in R_+^m}\{\langle\overline{\vec{f}}, \vec{z}\rangle : A(\vec{w})\vec{z} = \overline{\vec{f}},\ \langle\vec{e}, \vec{w}\rangle = T\}, \tag{50}$$

where $\overrightarrow{f}$ is a vector of external forces, $\vec{z} \in R^{2n}$ is a vector of virtual displacements of n nodes in R^2, $\vec{w}$ is a vector of m bars, and T is the total weight of construction. The compliance matrix $A(\vec{w})$ has the following form:

$$A(\vec{w}) = \sum_{i=1}^{m} \vec{w}_i \vec{a}_i \vec{a}_i^T,$$

where $\vec{a}_i \in R^{2n}$ are the vectors describing the interactions of two nodes connected by an arc. These vectors are very sparse: for 2D-model they have at most 4 nonzero elements.

Let us rewrite the problem (50) as a Linear Programming problem.

$$\begin{aligned}
&\min_{\vec{z},\vec{w}}\{\langle \overrightarrow{f}, \vec{z}\rangle : A(\vec{w})\vec{z} = \overrightarrow{f}, \vec{w} \geq 0, \langle \vec{e}, \vec{w}\rangle = T\} = \\
&= \min_{\vec{w}}\{\langle \overrightarrow{f}, A^{-1}(\vec{w})\overrightarrow{f}\rangle : \vec{w} \in \triangle(T) = \{\vec{w} \geq 0, \langle \vec{e}, \vec{w}\rangle = T\}\} = \\
&= \min_{\vec{w}\in\triangle(T)} \max_{\vec{z}}\{2\langle \overrightarrow{f}, \vec{z}\rangle - \langle A(\vec{w})\vec{z}, \vec{z}\rangle\} \geq \max_{\vec{z}} \min_{w\in\triangle(T)}\{2\langle \overrightarrow{f}, \vec{z}\rangle - \langle A(w)\vec{z}, \vec{z}\rangle\} = \\
&= \max_{\vec{z}}\{2\langle \overrightarrow{f}, \vec{z}\rangle - T \max_{1\leq i\leq m} \langle \vec{a}_i, \vec{z}\rangle^2\} = \max_{\lambda,\vec{y}}\{2\lambda\langle \overrightarrow{f}, \vec{y}\rangle - \lambda^2 T \max_{1\leq i\leq m}\langle \vec{a}_i, \vec{y}\rangle^2\} = \\
&= \max_{\vec{y}} \frac{\langle \overrightarrow{f}, \vec{y}\rangle^2}{T \max\limits_{1\leq i\leq m}\langle \vec{a}_i, \vec{y}\rangle^2} = \frac{1}{T}\left(\max_{\vec{y}}\{\langle \overrightarrow{f}, \vec{y}\rangle : \max_{1\leq i\leq m} |\langle \vec{a}_i, \vec{y}\rangle| \leq 1\}\right)^2.
\end{aligned} \tag{51}$$

Note that for the inequality in the third line we do not need any assumption.

Denote by $\vec{y}^*$ the optimal solution of the optimization problem in the brackets. Then there exist multipliers $\vec{x}^* \in R^m_+$ such that

$$\overrightarrow{f} = \sum_{i\in J_+} \vec{a}_i \vec{x}_i^* - \sum_{i\in J_-} \vec{a}_i \vec{x}_i^*, \qquad \vec{x}_i^* = 0, i \notin J_+ \bigcap J_-, \tag{52}$$

where $J_+ = \{i : \langle \vec{a}_i, \vec{y}^*\rangle = 1\}$, and $J_- = \{i : \langle \vec{a}_i, \vec{y}^*\rangle = -1\}$. Multiplying the first equation in (52) by $\vec{y}^*$, we get

$$\langle \overrightarrow{f}, \vec{y}^*\rangle = \langle \vec{e}, \vec{x}^*\rangle. \tag{53}$$

Note that the first equation in (52) can be written as

$$\overrightarrow{f} = A(\vec{x}^*)\vec{y}^*. \tag{54}$$

Let us reconstruct now the solution of the primal problem. Denote

$$w^* = \frac{T}{\langle \vec{e}, \vec{x}^*\rangle} \cdot \vec{x}^*, \qquad \vec{z}^* = \frac{\langle \vec{e}, \vec{x}^*\rangle}{T} \cdot \vec{y}^*. \tag{55}$$

Then, in view of (54) we have $\overrightarrow{f} = A(\vec{w}^*)\vec{z}^*$, and $\vec{w}^* \in \triangle(T)$. Thus, the pair (55) is feasible for the primal problem. On the other hand,

$$\langle \overrightarrow{f}, \vec{z}^*\rangle = \langle \overrightarrow{f}, \frac{\langle \vec{e}, \vec{x}^*\rangle}{T} \cdot \vec{y}^*\rangle = \frac{1}{T} \cdot \langle \vec{e}, \vec{x}^*\rangle \cdot \langle \overrightarrow{f}, \vec{y}^*\rangle = \frac{1}{T} \cdot \langle \overrightarrow{f}, \vec{y}^*\rangle^2.$$

Thus, the duality gap in the chain (51) is zero, and the pair $(\vec{w}^*, \vec{z}^*)$, defined by (55) is the optimal solution of the primal problem.

The above discussion allows us to concentrate on the following (dual) Linear Programming problem:

$$\max_{\vec{y}}\{\langle\bar{\vec{f}},\vec{y}\rangle : \max_{1\leq i\leq m}\langle\pm\vec{a}_i,\vec{y}\rangle\leq 1\}, \tag{56}$$

which we can solve by the primal-dual Algorithm 3.

Assume that we have *local* truss: each node is connected only with few neighbors. It allows to apply the property of *sparsity* for vectors $\vec{a}_i$ $(1\leq i\leq m)$. In this case the computational cost of each iteration grows as $O(\log_2 m)$ [57, 61].

In [57] a special class of huge-scale problems with sparse subgradient was considered. According to [57] for smooth functions this is a very rare feature. For example, for quadratic function $f(\vec{y})=\frac{1}{2}\langle A\vec{y},\vec{y}\rangle$ the gradient $\nabla f(\vec{y})=A\vec{y}$ usually is dense even for a sparse matrix A.

However, the subgradient of non-smooth function $f(\vec{y})=\max_{1\leq i\leq m}\langle\vec{a}_i,\vec{y}\rangle$ (see (56) above) are sparse provided that all vectors $\vec{a}_i$ share this property. This fact is based on the following observation. For the function $f(\vec{y})=\max_{1\leq i\leq m}\langle\vec{a}_i,\vec{y}\rangle$ with sparse matrix $A=(\vec{a}_1,\vec{a}_2,...,\vec{a}_m)$ the vector $\nabla f(\vec{y})=\vec{a}_{i(\vec{y})}$ is a subgradient at point $\vec{y}$. Then the standard subgradient step

$$\vec{y}_+=\vec{y}-h\cdot\nabla f(\vec{y})$$

changes only a few entries of vector $\vec{y}$ and the vector $\vec{z}_+=A^T\vec{y}_+$ differs from $\vec{z}=A^T\vec{y}$ also in a few positions only. Thus, the function value $f(\vec{y}_+)$ can be easily updated provided that we have an efficient procedure for recomputing the maximum of m values.

Note the objective functional in (56) is linear and the costs of iteration of Algorithm 3 and considered in [57] switching simple subgradient scheme is comparable. At the same time, the step productivity condition is simpler for Algorithm 3 as considered in [57] switching subgradient scheme. Therefore main observations for [57] are correct for Algorithm 3.

4.2 General Convex and Quasi-Convex Objective Functions

In this subsection, we assume that the objective function f in (36) might not satisfy (43) and, hence, its subgradient could be unbounded. One of the examples is a quadratic function. We also assume that inequality (44) holds.

We further consider ideas in [55, 59] and adapt them for problem (36), in a way that our algorithm allows to use non-Euclidean proximal setup, as does Mirror Descent, and does not require to know the constant M_g. Following [55], given a function f for each subgradient $\nabla f(\vec{x})$ at a point $\vec{y}\in X$, we define

$$v_f[\vec{y}](\vec{x})=\begin{cases}\left\langle\dfrac{\nabla f(\vec{x})}{\|\nabla f(\vec{x})\|_{E,*}},\vec{x}-\vec{y}\right\rangle, & \nabla f(\vec{x})\neq 0\\ 0 & \nabla f(\vec{x})=0\end{cases},\quad \vec{x}\in X. \tag{57}$$

The following result gives complexity estimate for Algorithm 4 in terms of $v_f[\vec{x}_*](\vec{x})$. Below we use this theorem to establish complexity result for smooth objective f.

Theorem 8. *Assume that inequality* (40) *holds and a known constant* $\Theta_0>0$ *is such that* $d(\vec{x}_*)\leq\Theta_0^2$. *Then, Algorithm 4 stops after not more than*

$$k=\left\lceil\frac{2\max\{1,M_g^2\}\Theta_0^2}{\varepsilon^2}\right\rceil \tag{58}$$

iterations and it holds that $\min_{i\in I}v_f[\vec{x}_*](\vec{x}^i)\leq\varepsilon$ *and* $g(\bar{\vec{x}}^k)\leq\varepsilon$.

Algorithm 4 Adaptive Mirror Descent (General Convex Objective)

```
Input: accuracy $\varepsilon > 0$; $\Theta_0$ s.t. $d(\vec{x}_*) \leq \Theta_0^2$.
$\vec{x}^0 = \arg\min_{\vec{x}\in X} d(\vec{x})$.
Initialize the set $I$ as empty set.
Set $k = 0$.
repeat
    if $g(\vec{x}^k) \leq \varepsilon$ then
        $h_k = \frac{\varepsilon}{\|\nabla f(\vec{x}^k)\|_{E,*}}$
        $\vec{x}^{k+1} = \mathrm{Mirr}[\vec{x}^k](h_k \nabla f(\vec{x}^k))$ ("productive step")
        Add $k$ to $I$.
    else
        $h_k = \frac{\varepsilon}{\|\nabla g(\vec{x}^k)\|_{E,*}^2}$
        $\vec{x}^{k+1} = \mathrm{Mirr}[\vec{x}^k](h_k \nabla g(\vec{x}^k))$ ("non-productive step")
    end if
    Set $k = k+1$.
until $|I| + \sum_{j\in J} \frac{1}{\|\nabla g(\vec{x}^j)\|_{E,*}^2} \geq \frac{2\Theta_0^2}{\varepsilon^2}$
Output: $\bar{\vec{x}}^k := \arg\min_{\vec{x}^j, j\in I} f(\vec{x}^j)$
```

To obtain the complexity of our algorithm in terms of the values of the objective function f, we define non-decreasing function

$$\omega(\tau) = \begin{cases} \max\limits_{\vec{x}\in X}\{f(\vec{x}) - f(\vec{x}_*) : \|\vec{x} - \vec{x}_*\|_E \leq \tau\} & \tau \geq 0, \\ 0 & \tau < 0. \end{cases} \tag{59}$$

and use the following lemma from [55].

Lemma 3. *Assume that f is a convex function. Then, for any $\vec{x} \in X$,*

$$f(\vec{x}) - f(\vec{x}_*) \leqslant \omega(v_f[\vec{x}_*](\vec{x})). \tag{60}$$

Corollary 9. *Assume that the objective function f in* (36) *is given as $f(\vec{x}) = \max_{i\in\{1,\ldots,m\}} f_i(\vec{x})$, where $f_i(\vec{x})$, $i = 1,\ldots,m$ are differentiable with Lipschitz-continuous gradient*

$$\|\nabla f_i(\vec{x}) - \nabla f_i(\vec{y})\|_{E,*} \leq L_i \|\vec{x} - \vec{y}\|_E \quad \forall \vec{x},\vec{y} \in X, \quad i \in \{1,\ldots,m\}. \tag{61}$$

Then $\bar{\vec{x}}^k$ is $\widetilde{\varepsilon}$-solution to (36) *in the sense of* (41)*, where*

$$\widetilde{\varepsilon} = \max\{\varepsilon, \varepsilon \max_{i=1,\ldots,m} \|\nabla f_i(\vec{x}_*)\|_{E,*} + \varepsilon^2 \max_{i=1,\ldots,m} L_i/2\}.$$

Remark 1. According to [52, 60] main lemma 3 holds for quasi-convex objective functions [13] too:

$$f(\alpha\vec{x} + (1-\alpha)\vec{y}) \leq \max\{f(\vec{x}), f(\vec{y})\} \text{ for all } \vec{x}, \vec{y}, \alpha \in [0,1].$$

This means that results of this subsection are valid for quasi-convex objectives.

Remark 2. In view of the Lipschitzness and, generally speaking, non-smoothness of functional limitations, the obtained estimate for the number of iterations means that the proposed method is optimal from the point of view of oracle evaluations: $O\left(\frac{1}{\varepsilon^2}\right)$ iterations are sufficient for achieving the required accuracy ε of solving the problem for the class of target functionals considered in this section of the article. Note also that the considered algorithm 3 applies to the considered classes of problems with constraints with convex objective functionals of different smoothness levels. However, the nonfulfillment, generally speaking, of the Lipschitz condition for the objective functional f does not allow one to substantiate the optimality of the algorithms 3 in the general situation (for example, with a Lipschitz-continuous gradient). More precisely, situations are possible when the productive steps of the norm (sub)gradients of the objective functional $\|\nabla f(\vec{x}^k)\|_*$ are large enough and this will interfere with the speedy achievement of the stopping criterion of the 3.

5 Universal Methods

In this section we consider problem

$$\min_{\vec{x}\in Q\subseteq E} f(\vec{x}), \tag{62}$$

where Q is a convex set and f is a convex function with Hölder-continuous subgradient

$$\|\nabla f(\vec{x}_1)-\nabla f(\vec{x}_2)\|_* \leq L_\nu \|\vec{x}_1-\vec{x}_2\|^\nu \tag{63}$$

with $\nu\in[0,1]$. The case $\nu=0$ corresponds to non-smooth optimization and the case $\nu=1$ corresponds to smooth optimization. The goal of this section is to present the Universal Accelerated Gradient method first proposed by Nesterov [58]. This method is a black-box method which does not require the knowledge of constants ν, L_ν and works in accordance with the lower complexity bound $O\left(\left(\frac{L_\nu R^{1+\nu}}{\varepsilon}\right)^{\frac{2}{1+3\nu}}\right)$ obtained in [50].

The main idea is based on the observation that a non-smooth convex function can be upper bounded by a quadratic objective function slightly shifted above. More precisely, for any $\vec{x},\vec{y}\in Q$,

$$\begin{aligned} f(\vec{y}) &\leq f(\vec{x})+\langle\nabla f(\vec{x}),\vec{y}-\vec{x}\rangle+\frac{L_\nu}{1+\nu}\|\vec{y}-\vec{x}\|^{1+\nu} \\ &\leq f(\vec{x})+\langle\nabla f(\vec{x}),\vec{y}-\vec{x}\rangle+\frac{L(\delta)}{2}\|\vec{y}-\vec{x}\|^2+\delta, \end{aligned} \tag{64}$$

where

$$L(\delta)=\left(\frac{1-\nu}{1+\nu}\frac{1}{\delta}\right)^{\frac{1-\nu}{1+\nu}} L_\nu^{\frac{2}{1+\nu}}.$$

The next idea is to apply an accelerated gradient method with backtracking procedure to adapt for the unknown $L(\delta)$ with appropriately chosen δ. The method we present is based on accelerated gradient method from [33, 32] and, thus is different from the original method of [58].

Inequality (64) guarantees that the backtracking procedure in the inner cycle is finite.

Theorem 10 ([58])**.** *Let f satisfy* (63)*. Then,*

$$f(\vec{y}^{k+1})-f_\star \leq \left(\frac{2^{2+4\nu}L_\nu^2}{\varepsilon^{1-\nu}k^{1+3\nu}}\right)^{\frac{1}{1+\nu}} V[\vec{x}^0](\vec{x}^\star)+\frac{\varepsilon}{2}. \tag{70}$$

Moreover, the number of oracle calls is bounded by

$$4(k+1)+2\log_2\left((2V[\vec{x}^0](\vec{x}^\star))^{\frac{1-\nu}{1+3\nu}}\left(\frac{1}{\varepsilon}\right)^{\frac{3(1-\nu)}{1+3\nu}} L_\nu^{\frac{4}{1+3\nu}}\right).$$

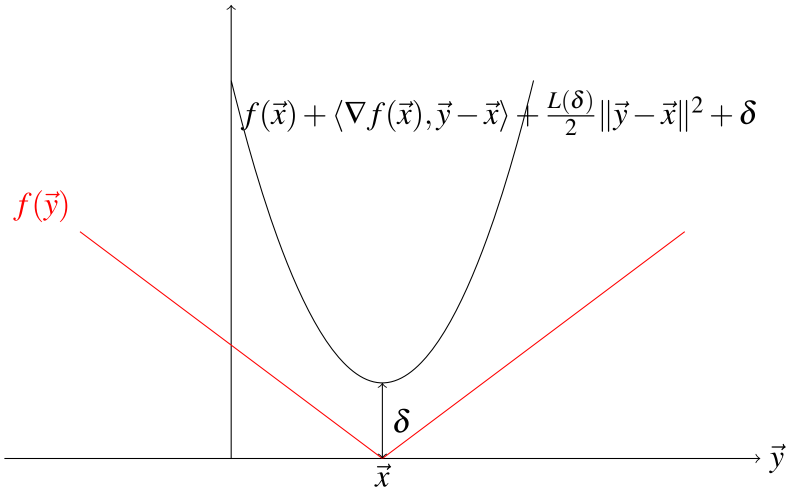
$f(\vec{x})+\langle\nabla f(\vec{x}),\vec{y}-\vec{x}\rangle+\frac{L(\delta)}{2}\|\vec{y}-\vec{x}\|^2+\delta$
$f(\vec{y})$
δ
$\vec{x}$
$\vec{y}$

Figure 3: Quadratic majorant of a non-smooth function $f(\vec{x})$.

Algorithm 5 Universal Accelerated Gradient Method

```
Input: Accuracy $\varepsilon$, starting point $\vec{x}^0 \in Q$, initial guess $L_0 > 0$, prox-setup: $d(\vec{x})$ – $1$-strongly convex w.r.t. $\|\cdot\|_E$,
    $V[\vec{z}](\vec{x}) := d(\vec{x}) - d(\vec{z}) - \langle \nabla d(\vec{z}), \vec{x} - \vec{z}\rangle$.
Set $k = 0$, $C_0 = \alpha_0 = 0$, $\vec{y}^0 = \vec{z}^0 = \vec{x}^0$.
for $k = 0, 1, \dots$ do
    Set $M_k = L_k/2$.
    repeat
        Set $M_k = 2M_k$, find $\alpha_{k+1}$ as the largest root of the equation
            $C_{k+1} := C_k + \alpha_{k+1} = M_k \alpha_{k+1}^2.$    (65)
        $\vec{x}^{k+1} = \frac{\alpha_{k+1}\vec{z}^k + C_k\vec{y}^k}{C_{k+1}}.$    (66)
        $\vec{z}^{k+1} = \arg\min_{\vec{x}\in Q}\{V[\vec{z}^k](\vec{x}) + \alpha_{k+1}(f(\vec{x}^{k+1}) + \langle \nabla f(\vec{x}^{k+1}), \vec{x} - \vec{x}^{k+1}\rangle)\}.$    (67)
        $\vec{y}^{k+1} = \frac{\alpha_{k+1}\vec{z}^{k+1} + C_k\vec{y}^k}{C_{k+1}}.$    (68)
    until
        $f(\vec{y}^{k+1}) \le f(\vec{x}^{k+1}) + \langle \nabla f(\vec{x}^{k+1}), \vec{y}^{k+1} - \vec{x}^{k+1}\rangle + \frac{M_k}{2}\|\vec{y}^{k+1} - \vec{x}^{k+1}\|^2 + \frac{\alpha_{k+1}\varepsilon}{2C_{k+1}}.$    (69)
    Set $L_{k+1} = M_k/2$, $k = k+1$.
end for
Output: The point $\vec{y}^{k+1}$.
```

Translating this rate of convergence to the language of complexity, we obtain that to obtain a solution with an accuracy ε the number of iterations is no more than

$$O\left(\inf_{\nu\in[0,1]}\left(\frac{L_\nu}{\varepsilon}\right)^{\frac{2}{1+3\nu}}\left(V[\vec{x}^0](\vec{x}^\star)\right)^{\frac{1+\nu}{1+3\nu}}\right),$$

i.e. is optimal.

In his paper, Nesterov considers a more general composite optimization problem

$$\min_{\vec{x}\in Q\subseteq E} f(\vec{x})+h(\vec{x}), \tag{71}$$

where h is a simple convex function, and obtains the same complexity guarantees. Universal methods were extended for the case of strongly convex problems by a restart technique in [66], for non-convex optimization in [35] and for the case of non-convex optimization with inexact oracle in [28]. As we can see from (64), universal accelerated gradient method is connected to smooth problems with inexact oracle. The study of accelerated gradient methods with inexact oracle was first proposed in [22] and was very well developed in [24, 30, 11, 28] including stochastic optimization problems and strongly convex problems. A universal method with inexact oracle can be found in [31]. Experiments show [58] that universal method accelerates to $O\left(\frac{1}{k}\right)$ rate for non-smooth problems with a special ßmoothing friendly"(see Section 3) structure. This is especially interesting for traffic flow modeling problems, which possess such structure [3].

Now we consider universal analog of A.S. Nemirovski's proximal mirror method for variational inequalities with a Holder-continuous operator. More precisely, we consider universal extension of Algorithm 2 which allows to solve smooth and non-smooth variational inequalities without the prior knowledge of the smoothness. Main idea of the this method is the adaptive choice of constants and level of smoothness in minimized prox-mappings at each iteration. These constants are related to the Hölder constant of the operator and this method allows to find a suitable constant at each iteration.

Algorithm 6 Universal Mirror Prox

```
Input: General VI on a set $Q \subset E$ with operator $\Phi(\vec{z})$, accuracy $\varepsilon > 0$, initial guess $M_{-1} > 0$, prox-setup: $d(\vec{z})$, $V[\vec{z}](\vec{w})$.
Set $k = 0$, $\vec{z}^0 = \arg\min_{\vec{z}\in Q} d(\vec{z})$.
for $k = 0, 1, \ldots$ do
    Set $i_k = 0$
    repeat
        Set $M_k = 2^{i_k - 1} M_{k-1}$.
        Calculate
            $\vec{w}^k = \arg\min_{\vec{z}\in Q} \left\{ \langle \Phi(\vec{z}^k), \vec{z} \rangle + M_k V[\vec{z}^k](\vec{z}) \right\}$.   (72)
        Calculate
            $\vec{z}^{k+1} = \arg\min_{\vec{z}\in Q} \left\{ \langle \Phi(\vec{w}^k), \vec{z} \rangle + M_k V[\vec{z}^k](\vec{z}) \right\}$.   (73)
        $i_k = i_k + 1$.
    until
        $\langle \Phi(\vec{w}^k) - \Phi(\vec{z}^k), \vec{w}^k - \vec{z}^{k+1} \rangle \leq \frac{M_k}{2} \left( \|\vec{w}^k - \vec{z}^k\|^2 + \|\vec{w}^k - \vec{z}^{k+1}\|^2 \right) + \frac{\varepsilon}{2}$.   (74)
    Set $k = k + 1$.
end for
Output: $\widehat{\vec{w}}^k = \frac{1}{k}\sum_{i=0}^{k-1} \vec{w}^i$.
```

Theorem 11 ([34]). *For any $k \geq 1$ and any $\vec{z} \in Q$,*

$$\frac{1}{\sum_{i=0}^{k-1} M_i^{-1}} \sum_{i=0}^{k-1} M_i^{-1} \langle \Phi(\vec{w}^i), \vec{w}^i - \vec{z} \rangle \leq \frac{1}{\sum_{i=0}^{k-1} M_i^{-1}} (V[\vec{z}^0](\vec{z}) - V[\vec{z}^k](\vec{z})) + \frac{\varepsilon}{2}. \tag{75}$$

Note that if $\max_{\vec{z}\in Q} V[\vec{z}^0](\vec{z}) \leq D$, we can construct the following adaptive stopping criterion for our algorithm

$$\frac{D}{\sum_{i=0}^{k-1} M_i^{-1}} \leq \frac{\varepsilon}{2}.$$

Next, we consider the case of Hölder-continuous operator Φ and show that Algorithm 6 is universal. Assume for some $\nu \in [0,1]$ and $L_\nu \geq 0$

$$\|\Phi(\vec{x}) - \Phi(\vec{y})\|_* \leq L_\nu \|\vec{x} - \vec{y}\|^\nu, \quad \vec{x}, \vec{y} \in Q.$$

holds. The following inequality is a generalization of (64) for VI. For any $\vec{x}, \vec{y}, \vec{z} \in Q$ and $\delta > 0$,

$$\langle \Phi(\vec{y}) - \Phi(\vec{x}), \vec{y} - \vec{z} \rangle \leq \|\Phi(\vec{y}) - \Phi(\vec{x})\|_* \|\vec{y} - \vec{z}\| \leq L_\nu \|\vec{x} - \vec{y}\|^\nu \|\vec{y} - \vec{z}\| \leq$$

$$\leq \frac{1}{2} \left(\frac{1}{\delta} \right)^{\frac{1-\nu}{1+\nu}} L_\nu^{\frac{2}{1+\nu}} \left(\|\vec{x} - \vec{y}\|^2 + \|\vec{y} - \vec{z}\|^2 \right) + \frac{\delta}{2},$$

where

$$L(\delta) = \left(\frac{1}{\delta} \right)^{\frac{1-\nu}{1+\nu}} L_\nu^{\frac{2}{1+\nu}}. \tag{76}$$

So, we have

$$\langle \Phi(\vec{y}) - \Phi(\vec{x}), \vec{y} - \vec{z} \rangle \leq \frac{L(\delta)}{2} \left(\|\vec{y} - \vec{x}\|^2 + \|\vec{y} - \vec{z}\|^2 \right) + \delta. \tag{77}$$

Let us consider estimates of the necessary number of iterations are obtained to achieve a given quality of the variational inequality solution.

Corollary 12 (Universal Method for VI)**.** *Assume that the operator Φ is Hölder continuous with constant L_ν for some $\nu \in [0,1]$ and $M_{-1} \leq \left(\frac{2}{\varepsilon}\right)^{\frac{1-\nu}{1+\nu}} L_\nu^{\frac{2}{1+\nu}}$. Also assume that the set Q is bounded. Then, for all $k \geq 0$, we have*

$$\max_{\vec{z} \in Q} \langle \Phi(\vec{z}), \widehat{\vec{w}}_k - \vec{z} \rangle \leq \frac{(2L_\nu)^{\frac{2}{1+\nu}}}{k\varepsilon^{\frac{1-\nu}{1+\nu}}} \max_{\vec{z} \in Q} V[\vec{z}^0](\vec{z}) + \frac{\varepsilon}{2} \tag{78}$$

As it follows from (77), if $M_k \geq L(\frac{\varepsilon}{2})$, (74) holds. Thus, for all $i = 0, ..., k-1$, we have $M_i \leq 2 \cdot L(\frac{\varepsilon}{2})$ and

$$\frac{1}{\sum_{i=0}^{k-1} M_i^{-1}} \leq \frac{2L(\frac{\varepsilon}{2})}{k} \leq \frac{(2L_\nu)^{\frac{2}{1+\nu}}}{k\varepsilon^{\frac{1-\nu}{1+\nu}}},$$

(78) holds. Here $L(\cdot)$ is defined in (76). □

Let us add some remarks.

Remark 3. Since the algorithm does not use the values of parameters ν and L_ν, we obtain the following iteration complexity bound

$$2 \inf_{\nu \in [0,1]} \left(\frac{2L_\nu}{\varepsilon} \right)^{\frac{2}{1+\nu}} \cdot \max_{\vec{z} \in Q} V[\vec{z}_0](\vec{z})$$

to achieve

$$\max_{\vec{z} \in Q} \langle \Phi(\vec{z}), \widehat{\vec{w}}_k - \vec{z} \rangle \leq \varepsilon.$$

Using the same reasoning as in [58], we estimate the number of oracle calls for Algorithm 6. The number of oracle calls on each iteration k is equal to $2i_k$. At the same time, $M_k = 2^{i_k - 2} M_{k-1}$ and, hence, $i_k = 2 + \log_2 \frac{M_k}{M_{k-1}}$. Thus, the total number of oracle calls is

$$\sum_{j=0}^{k-1} i_j = 4k + 2 \sum_{i=0}^{k-1} \log_2 \frac{M_j}{M_{j-1}} < 4k + 2\log_2 \left(2L\left(\frac{\varepsilon}{2}\right) \right) - 2\log_2(M_{-1}), \tag{79}$$

where we used that $M_k \leq 2L(\frac{\varepsilon}{2})$.

Thus, the number of oracle calls of the Algorithm 6 does not exceed:

$$4 \inf_{\nu \in [0,1]} \left(\frac{2 \cdot L_\nu}{\varepsilon} \right)^{\frac{2}{1+\nu}} \cdot \max_{u \in Q} V[z_0](u) + 2 \inf_{\nu \in [0,1]} \log_2 2 \left(\left(\frac{2}{\varepsilon} \right)^{\frac{1-\nu}{1+\nu}} L_\nu^{\frac{2}{1+\nu}} \right) - 2\log_2(M_{-1}).$$

Remark 4. We can apply this method to convex-concave saddle problems of the form

$$f(\vec{x}, \vec{y}) \to \min_{\vec{x} \in Q_1} \max_{\vec{y} \in Q_2}, \tag{80}$$

where $Q_{1,2}$ are convex compacts in $\mathbb{R}^n$, f is convex in $\vec{x}$ and concave in $\vec{y}$, there is $\nu \in [0,1]$ and constants $L_{11,\nu}, L_{12,\nu}, L_{21,\nu}, L_{22,\nu} < +\infty$:

$$\|\nabla_{\vec{x}} f(\vec{x}+\Delta\vec{x}, \vec{y}+\Delta\vec{y}) - \nabla_{\vec{x}} f(\vec{x},\vec{y})\|_{1,*} \leq L_{11,\nu}\|\Delta\vec{x}\|_1^\nu + L_{12,\nu}\|\Delta\vec{y}\|_2^\nu,$$

$$\|\nabla_{\vec{y}} f(\vec{x}+\Delta\vec{x}, \vec{y}+\Delta\vec{y}) - \nabla_{\vec{y}} f(\vec{x},\vec{y})\|_{2,*} \leq L_{21,\nu}\|\Delta\vec{x}\|_1^\nu + L_{22,\nu}\|\Delta\vec{y}\|_2^\nu$$

for all $\vec{x}, \vec{x}+\Delta\vec{x} \in Q_1, \vec{y}, \vec{y}+\Delta\vec{y} \in Q_2$.

It is possible to achieve an acceptable approximation $(\widehat{\vec{x}}, \widehat{\vec{y}}) \in Q_1 \times Q_2$:

$$\max_{\vec{y}\in Q_2} f(\widehat{\vec{x}}, \vec{y}) - \min_{\vec{x}\in Q_1} f(\vec{x}, \widehat{\vec{y}}) \leq \varepsilon \tag{81}$$

for the saddle point $(\vec{x}_*, \vec{y}_*) \in Q_1 \times Q_2$ of the (80) problem in no more than

$$O\left(\left(\frac{1}{\varepsilon}\right)^{\frac{2}{1+\nu}}\right)$$

iterations, which indicates the optimality of the proposed method, at least for $\nu = 0$ and $\nu = 1$. However, in practice experiments show that (81) can be achieved much faster due to the adaptability of the method.

6 Concluding remarks

Modern numerical methods for non-smooth convex optimization problems are typically based on the structure of the problem. We start with one of the most powerful example of such type. For geometric median search problem there exists efficient method that significantly outperform described above lower complexity bounds [19]. In Machine Learning we typically meet the problems with hidden affine structure and small effective dimension (SVM) that allow us to use different smoothing techniques [1]. Description of one of these techniques (Nesterov's smoothing technique) one can find in this survey. The other popular technique is based on averaging of the function around the small ball with the center at the point in consideration [27]. A huge amount of data since applications lead to composite optimization problems with non smooth composite (LASSO). For this class of problems accelerated (fast) gradient methods are typically applied [7], [56], [41]. This approach (composite optimization) have been recently expanded for more general class of problems [73]. In different Image Processing applications one can find a lot of non-smooth problems formulations with saddle-point structure. That is the goal function has Legendre representation. In this case one can apply special versions of accelerated (primal-dual) methods [16], [17], [43]. Universal Mirror Prox method described above demonstrates the alternative approach which can be applied in rather general context. Unfortunately, the most of these tricks have proven to be beyond the scope of this survey. But we include in the survey the description of the Universal Accelerated Gradient Descent algorithm [73] which in the general case can also be applied to a wide variety of problems.

Another important direction in Non-smooth Convex Optimization is huge-scale optimization for sparse problems [57]. The basic idea that reduce huge dimension to non-smoothness is as follows:

$$\langle \vec{a}_k, \vec{x}\rangle - b_k \leq 0, \quad k = 1, \ldots m, \quad m \gg 1$$

is equivalent to the single non-smooth constraint:

$$\max_{k=1,\ldots m} \{\langle \vec{a}_k, \vec{x}\rangle - b_k\} \leq 0.$$

We demonstrated this idea above on Truss Topology Design example.

One should note that we concentrate in this survey only on deterministic convex optimization problems, but the most beautiful things in non smooth optimization is that stochasticity [50], [26], [38], [39] and online context [36] in general doesn't change (up to a logarithmic factor in the strongly convex case) anything in complexity estimates. As an example, of stochastic (randomized) approach one can mentioned the work [2] where one can find reformulation of Google problem as non smooth convex optimization problem. Special randomized Mirror Descent algorithm allows to solve this problem almost independently on the number of vertexes.

Finally, let's note that in the decentralized distributed non smooth (stochastic) convex optimization for the last few years there appear optimal methods [42], [75], [14].

References

[1] Z. Allen-Zhu and E. Hazan. Optimal black-box reductions between optimization objectives. In *Advances in Neural Information Processing Systems*, pages 1614–1622, 2016.

[2] A. Anikin, A. Gasnikov, A. Gornov, D. Kamzolov, Y. Maximov, and Y. Nesterov. Efficient numerical methods to solve sparse linear equations with application to pagerank. *arXiv preprint arXiv:1508.07607*, 2015.

[3] D. Baimurzina, A. Gasnikov, E. Gasnikova, P. Dvurechensky, E. Ershov, M. Kubentaeva, and A. Lagunovskaya. Universal similar triangulars method for searching equilibriums in traffic flow distribution models. *arXiv:1701.02473*, 2017.

[4] A. Bayandina, P. Dvurechensky, A. Gasnikov, F. Stonyakin, and A. Titov. Mirror descent and convex optimization problems with non-smooth inequality constraints. In P. Giselsson and A. Rantzer, editors, *Large-Scale and Distributed Optimization*, chapter 8, pages 181–215. Springer International Publishing, 2018. arXiv:1710.06612.

[5] A. Beck, A. Ben-Tal, N. Guttmann-Beck, and L. Tetruashvili. The comirror algorithm for solving nonsmooth constrained convex problems. *Operations Research Letters*, 38(6):493 – 498, 2010.

[6] A. Beck and M. Teboulle. Mirror descent and nonlinear projected subgradient methods for convex optimization. *Oper. Res. Lett.*, 31(3):167–175, May 2003.

[7] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009.

[8] A. Ben-Tal and A. Nemirovski. Robust truss topology design via semidefnite programming. *SIAM J. Optim.*, 7(4):991 – 1016, 1997.

[9] A. Ben-Tal and A. Nemirovski. *Lectures on Modern Convex Optimization (Lecture Notes)*. Personal web-page of A. Nemirovski, 2015.

[10] L. Blum, F. Cucker, M. Shub, and S. Smale. *Complexity and real computation*. Springer Science & Business Media, 2012.

[11] L. Bogolubsky, P. Dvurechensky, A. Gasnikov, G. Gusev, Y. Nesterov, A. M. Raigorodskii, A. Tikhonov, and M. Zhukovskii. Learning supervised pagerank with gradient-based and gradient-free optimization methods. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 4914–4922. Curran Associates, Inc., 2016. arXiv:1603.00717.

[12] R. Brent. *Algorithms for Minimization Without Derivatives*. Dover Books on Mathematics. Dover Publications, 1973.

[13] B.T.Polyak. Minimization of nonsmooth functionals. *USSR Computational Mathematics and Mathematical Physics*, 9(3):14–29, 1969.

[14] K. S. F. B. S. Bubeck, Y. T. Lee, and L. Massoulie. Optimal algorithms for non-smooth distributed optimization in networks.

[15] S. Bubeck. Convex optimization: algorithms and complexity. *Foundations and Trends in Machine Learning*, 8(3–4):231–357, 2015.

[16] A. Chambolle and T. Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of Mathematical Imaging and Vision*, 40(1):120–145, 2011.

[17] Y. Chen, G. Lan, and Y. Ouyang. Optimal primal-dual methods for a class of saddle point problems. *SIAM Journal on Optimization*, 24(4):1779–1814, 2014.

[18] Y. Chen, G. Lan, and Y. Ouyang. Accelerated schemes for a class of variational inequalities. *Mathematical Programming*, 165(1):113–149, 2017.

[19] M. B. Cohen, Y. T. Lee, G. Miller, J. Pachocki, and A. Sidford. Geometric median in nearly linear time. In *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*, pages 9–21. ACM, 2016.

[20] M. Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 2292–2300. Curran Associates, Inc., 2013.

[21] M. Cuturi and A. Doucet. Fast computation of wasserstein barycenters. In E. P. Xing and T. Jebara, editors, *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 685–693, Bejing, China, 22–24 Jun 2014. PMLR.

[22] A. d'Aspremont. Smooth optimization with approximate gradient. *SIAM J. on Optimization*, 19(3):1171–1183, Oct. 2008.

[23] A. Demyanov, V. Demyanov, and V. Malozemov. Minmaxmin problems revisited. *Optimization Methods and Software*, 17(5):783–804, 2002.

[24] O. Devolder, F. Glineur, and Y. Nesterov. First-order methods of smooth convex optimization with inexact oracle. *Mathematical Programming*, 146(1):37–75, 2014.

[25] Y. Drori and M. Teboulle. An optimal variants of kelley's cutting-plane method. *Mathematical Programming*, 160(1–2):321–351, 2016.

[26] J. Duchi. Introductory lectures on stochastic optimization. *Park City Mathematics Institute, Graduate Summer School Lectures*, 2016.

[27] J. C. Duchi, P. L. Bartlett, and M. J. Wainwright. Randomized smoothing for stochastic optimization. *SIAM Journal on Optimization*, 22(2):674–701, 2012.

[28] P. Dvurechensky. Gradient method with inexact oracle for composite non-convex optimization. *arXiv:1703.09180*, 2017.

[29] P. Dvurechensky, D. Dvinskikh, A. Gasnikov, C. A. Uribe, and A. Nedić. Decentralize and randomize: Faster algorithm for Wasserstein barycenters. In *Proceedings of the 32th Conference on Neural Information Processing Systems*, NIPS'18, 2018. (Accepted), arXiv:1802.04367.

[30] P. Dvurechensky and A. Gasnikov. Stochastic intermediate gradient method for convex problems with stochastic inexact oracle. *Journal of Optimization Theory and Applications*, 171(1):121–145, 2016.

[31] P. Dvurechensky, A. Gasnikov, and D. Kamzolov. Universal intermediate gradient method for convex problems with inexact oracle. *arXiv:1712.06036*, 2017.

[32] P. Dvurechensky, A. Gasnikov, and A. Kroshnin. Computational optimal transport: Complexity by accelerated gradient descent is better than by Sinkhorn's algorithm. In J. Dy and A. Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 1367–1376, 2018. arXiv:1802.04367.

[33] P. Dvurechensky, A. Gasnikov, S. Omelchenko, and A. Tiurin. Adaptive similar triangles method: a stable alternative to sinkhorn's algorithm for regularized optimal transport. *arXiv:1706.07622*, 2017.

[34] P. Dvurechensky, A. Gasnikov, F. Stonyakin, and A. Titov. Generalized Mirror Prox: Solving variational inequalities with monotone operator, inexact oracle, and unknown Hölder parameters. *arXiv:1806.05140*, 2018.

[35] S. Ghadimi, G. Lan, and H. Zhang. Generalized uniformly optimal methods for nonlinear programming. *arXiv:1508.07384*, 2015.

[36] E. Hazan et al. Introduction to online convex optimization. *Foundations and Trends® in Optimization*, 2(3-4):157–325, 2016.

[37] A. Juditsky and A. Nemirovski. First order methods for non-smooth convex large-scale optimization, i: General purpose methods. In S. W. Suvrit Sra, Sebastian Nowozin, editor, *Optimization for Machine Learning*, pages 121–184. Cambridge, MA: MIT Press, 2012.

[38] A. Juditsky, A. Nemirovski, et al. First order methods for nonsmooth convex large-scale optimization, i: general purpose methods. *Optimization for Machine Learning*, pages 121–148, 2011.

[39] A. Juditsky, A. Nemirovski, et al. First order methods for nonsmooth convex large-scale optimization, ii: utilizing problems structure. *Optimization for Machine Learning*, pages 149–183, 2011.

[40] L. G. Khachiyan. A polynomial algorithm in linear programming. In *Doklady Academii Nauk SSSR*, volume 244, pages 1093–1096, 1979.

[41] G. Lan. Gradient sliding for composite optimization. *Mathematical Programming*, 159(1):201–235, Sep 2016.

[42] G. Lan, S. Lee, and Y. Zhou. Communication-efficient algorithms for decentralized and stochastic optimization. *arXiv preprint arXiv:1701.03961*, 2017.

[43] G. Lan and Y. Ouyang. Accelerated gradient sliding for structured convex optimization. *arXiv preprint arXiv:1609.04905*, 2016.

[44] Y. T. Lee, A. Sidford, and S. C.-w. Wong. A faster cutting plane method and its implications for combinatorial and convex optimization. In *Foundations of Computer Science (FOCS), 2015 IEEE 56th Annual Symposium on*, pages 1049–1065. IEEE, 2015.

[45] A. Y. Levin. On an algorithm for the minimization of convex functions. *Soviet Math. Doklady*, 1965.

[46] A. Nedić and A. Ozdaglar. Approximate primal solutions and rate analysis for dual subgradient methods. *SIAM Journal on Optimization*, 19(4):1757–1780, 2009.

[47] A. Nemirovski. Prox-method with rate of convergence $o(1/t)$ for variational inequalities with Lipschitz continuous monotone operators and smooth convex-concave saddle point problems. *SIAM Journal on Optimization*, 15(1):229–251, 2004.

[48] A. Nemirovskii. Efficient methods for large-scale convex optimization problems. *Ekonomika i Matematicheskie Metody*, 15, 1979. In Russian.

[49] A. Nemirovskii and Y. Nesterov. Optimal methods of smooth convex minimization. *USSR Computational Mathematics and Mathematical Physics*, 25(2):21 – 30, 1985.

[50] A. Nemirovsky and D. Yudin. *Problem Complexity and Method Efficiency in Optimization*. J. Wiley & Sons, New York, 1983.

[51] Y. Nesterov. A method of solving a convex programming problem with convergence rate $o(1/k^2)$. *Soviet Mathematics Doklady*, 27(2):372–376, 1983.

[52] Y. Nesterov. *Effective methods in nonlinear programming*. Moscow, 1989.

[53] Y. Nesterov. Smooth minimization of non-smooth functions. *Mathematical Programming*, 103(1):127–152, 2005.

[54] Y. Nesterov. Primal-dual subgradient methods for convex problems. *Mathematical Programming*, 120(1):221–259, Aug 2009. First appeared in 2005 as CORE discussion paper 2005/67.

[55] Y. Nesterov. *Introduction to Convex Optimization*. Moscow, MCCME, 2010.

[56] Y. Nesterov. Gradient methods for minimizing composite functions. *Mathematical Programming*, 140(1):125–161, 2013. First appeared in 2007 as CORE discussion paper 2007/76.

[57] Y. Nesterov. Subgradient methods for huge-scale optimization problems. *Mathematical Programming*, 146(1):275–297, Aug 2014. First appeared in 2012.

[58] Y. Nesterov. Universal gradient methods for convex optimization problems. *Mathematical Programming*, 152(1):381–404, 2015.

[59] Y. Nesterov. Subgradient methods for convex functions with nonstandard growth properties, 2016. http://www.mathnet.ru:8080/PresentFiles/16179/growthbm_nesterov.pdf.

[60] Y. Nesterov. *Lectures on Convex Optimization*. Springer International Publishing, 2018.

[61] Y. Nesterov and S. Shpirko. Primal-dual subgradient method for huge-scale linear conic problems. *SIAM Journal on Optimization*, 24(3):1444–1457, 2014.

[62] D. Newman. Location of the maximum on unimodal surfaces. *Journal of the Association for Computing Machinery*, 12:395–398, 1965.

[63] B. Polyak. A general method of solving extremum problems. *Soviet Mathematics Doklady*, 8(3):593–597, 1967.

[64] B. Polyak. *Introduction to Optimization*. New York, Optimization Software, 1987.

[65] R. Rockafellar. *Convex Analysis*. Priceton University, Princeton, 1970.

[66] V. Roulet and A. d'Aspremont. Sharpness, restart and acceleration. *arXiv:1702.03828*, 2017.

[67] M. S. S. Lacost-Julien and F. Bach. A simpler approach to obtaining $o(1/t)$ convergence rate for the projected stochastic subgradient method. arxiv preprint arxiv:1212.2002. 2012.

[68] N. Shor. *Minimization of Nondifferentiable Functions*. Naukova Dumka, 1979.

[69] N. Shor. *Minimization Methods for Non-Differentiable Functions*. Springer-Verlag Berlin Heidelberg, 1985.

[70] N. Z. Shor. Generalized gradient descent with application to block programming. *Kibernetika*, 3(3):53–55, 1967.

[71] N. Z. Shor, K. C. Kiwiel, and A. Ruszczynski. *Minimization Methods for Non-Differentiable Functions*, volume 3 of *Springer Series in Computational Mathematics*. Springer Berlin Heidelberg, 2012.

[72] Q. Tran-Dinh, O. Fercoq, and V. Cevher. A smooth primal-dual optimization framework for nonsmooth composite convex minimization. *SIAM Journal on Optimization*, 28(1):96–134, 2018. arXiv:1507.06243.

[73] A. Tyurin and A. Gasnikov. Fast gradient descent method for convex optimization problems with an oracle that generates a model of a function in a requested point. *arXiv preprint arXiv:1711.02747*, 2017.

[74] C. A. Uribe, D. Dvinskikh, P. Dvurechensky, A. Gasnikov, and A. Nedić. Distributed computation of Wasserstein barycenters over networks. In *2018 IEEE 57th Annual Conference on Decision and Control (CDC)*, 2018. Accepted, arXiv:1803.02933.

[75] C. A. Uribe, S. Lee, A. Gasnikov, and A. Nedić. Optimal algorithms for distributed optimization. *arXiv preprint arXiv:1712.00232*, 2017.

[76] P. M. Vaidya. Speeding-up linear programming using fast matrix multiplication. *In Foundations of Computer Science, 1989, 30th Annual Symposium on*, pages 332–337, 1989.