A universal modification of the linear coupling method

Sergey Guminov[a*], Alexander Gasnikov[a,b], Anton Anikin[c], Alexander Gornov[c]

[a] *Moscow Institute of Physics and Technology, Moscow, Russia*
[b] *Institute for Information Transmission Problems RAS, Moscow, Russia*
[c]*Matrosov Institute for System Dynamics and Control Theory of Siberian Branch of Russian Academy of Sciences, Irkutsk*

In the late sixties, N. Shor and B. Polyak independently proposed optimal first-order methods for solving non-smooth convex optimization problems. In 1982 A. Nemirovski proposed optimal first-order methods for solving smooth convex optimization problems, which utilized auxiliary line search. In 1985 A. Nemirovski and Yu. Nesterov proposed a parametric family of optimal first-order methods for solving convex optimization problems with intermediate smoothness. In 2013 Yu. Nesterov proposed a universal gradient method which combined all good properties of the previous methods, except the possibility of using auxiliary line search. One can typically observe that in practice auxiliary line search improves performance for many tasks. In this paper, we propose the apparently first such method of non-smooth convex optimization allowing the use of the line search procedure. Moreover, it is based on the universal gradient method, which does not require any a priori information about the actual degree of smoothness of the problem. Numerical experiments demonstrate that the proposed method is, in some cases, considerably faster than Nesterov's universal gradient method.

Keywords: Convex optimization; First-order methods; Non-smooth optimization; Line search

AMS Subject Classification: 90C25, 68Q25

1. Introduction

Traditionally, convex optimization problems have been divided into two main classes: the class of smooth problems and the class of non-smooth problems [12]. However, introducing an intermediate class of problems with convex differentiable objectives with Hölder continuous gradient allows us to view the classes of smooth and non-smooth convex optimization problems as two extreme cases of this intermediate class.

The first optimal methods for this class were introduced in [8]. However, both these procedures and some others presented later had a serious drawback: they required too much information about the objective (for example, the degree of the objective function's smoothness or the distance from the initial point to the solution) to be used efficiently.

In [11] the Universal Fast Gradient Method is presented. It is optimal for the class of problems with convex differentiable objectives with Hölder continuous gradient, has a low iteration cost, and does not involve any parameters dependent on the objective.

Some minimization methods allow for the use of an exact line search procedure. A classic example of such a method is the steepest descent method, which is a version of

---

the gradient descent method in which on each iteration instead of performing a step of fixed length in the direction of the negative gradient the objective function is minimized along said direction. Although this does not improve the worst-case convergence rate, such line search procedures often perform very well in practice. The aim of this work was to construct a universal method which allowed for the use of an exact line search procedure. By combining the core idea of Nesterov's Universal Fast Gradient Method with the framework described by Allen-Zhu et al. in [1], such a method was devised. As far as it is known to the authors of this paper, our work contains the first example of such a method, although a method utilising exact line search for solving minimization problems with convex Lipschitz continuous objectives was recently constructed by Drori et al [4]. Their work also contains an example of a universal method which uses an exact three dimension subspace minimization on each iteration. Our numerical experiments indicate that the exact line search step does indeed demonstrate great performance on some non-smooth problems. Note that in the well-known Shor's type methods with variable metric for non-smooth convex optimization problems line search is performed not in the direction of the negative gradient. These methods also require quadratic memory [12].

The paper is organized as follows. Firstly, we define the intermediate class of problems which we refer to above, set the problem and give other definition used later in this paper. Secondly, we define Nesterov's Universal Fast Gradient Method, which we will be using as a benchmark in our numerical experiments. In **Section 2** we present our Universal Linear Coupling Method, prove its convergence and equip it with a stopping criterion. **Section 3** contains notes on how to implement the line search procedure and how its accuracy affects the method's convergence. Finally, **Section 4** is dedicated to the results of our numerical experiments.

1.1 *Preliminaries*

One of the conditions often used in convergence analysis of numerical optimization methods is L-smoothness.

DEFINITION 1   A function $f : \mathbb{R}^n \to \mathbb{R}^m$ is called Lipschitz continuous with constant L if

$$\|f(x) - f(y)\| \leqslant L\|x - y\| \quad \forall x, y \in \mathbb{R}^n.$$

DEFINITION 2   A differentiable function $f : \mathbb{R}^n \to \mathbb{R}^m$ is called L-smooth if its gradient is Lipschitz continuous with constant L:

$$\|\nabla f(x) - \nabla f(y)\| \leqslant L\|x - y\| \quad \forall x, y \in \mathbb{R}^n.$$

We will be using the following natural generalisation of Lipschitz continuity.

DEFINITION 3   A function $f : \mathbb{R}^n \to \mathbb{R}^m$ satisfies the Hölder condition (or is Hölder continuous) if there exist constants $\nu \in [0, 1]$ and $M_\nu \geqslant 0$, such that

$$\|f(x) - f(y)\| \leqslant M_\nu \|x - y\|^\nu \quad \forall\ x, y \in \mathbb{R}^n.$$

The constant ν in this definition is called the exponent of the Hölder condition. Hölder continuity coincides with Lipschitz continuity if $\nu = 1$. On the other hand, Hölder con-

tinuity with $\nu = 0$ is just boundedness. If a function is differentiable and its gradient is Hölder continuous, then exponent ν is a measure of the function's smoothness.

Throughout this paper we will be working with the problem

$$f(x) \to \min_{x \in \mathbb{R}^n},$$

where $f(x)$ is a convex differentiable function and its gradient satisfies the Hölder condition for some $\nu \in [0, 1]$ with some constant M_ν. We denote some solution to this problem as x^*.

Let us define Bregman divergence $V_x(y)$ as follows:

$$V_x(y) = \omega(y) - \langle \nabla \omega(x), y - x \rangle - \omega(x),$$

where $\omega(x)$ is a 1-strongly convex function. ω is also called a distance generating function. By definition,

$$V_x(y) \geqslant \frac{1}{2} \|y - x\|^2.$$

1.2 *Universal Method*

In [3] it is shown that the notion of inexact oracle allows one to apply some methods of smooth convex optimization to non-smooth problems. The following lemma plays a key role in this:

LEMMA 1.1 *Let function f be differentiable and have Hölder continuous gradient. Then for any $\delta > 0$ we have*

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

where

$$M = M\left(\delta, \nu, M_\nu\right) = \left[\frac{1 - \nu}{1 + \nu} \frac{M_\nu}{\delta} \right]^{\frac{1 - \nu}{1 + \nu}} M_\nu.$$

The exact values $(f(x), \nabla f(x))$ of a differentiable function f with Hölder continuous gradient allow us to obtain an upper bound similar to the one obtained by using inexact information for a differentiable and L-smooth function. This allows one to apply methods reliant on the usage of an inexact oracle for L-smooth objectives to optimize objectives with Hölder continuos gradient.

However, knowledge of the parameters ν and M_ν from the definition of Hölder continuity is still required to apply such an approach. In [11] a line search procedure was used to estimate the needed parameters similarly to how the constant of L-smoothness is estimated in adaptive methods. For a general norm on $\mathbb{R}^n$ and a corresponding Bregman divergence $V_x(y)$ the Universal Fast Gradient Method may be written as follows.

Algorithm 1: UFGM(f, L_0, x_0, ε, T)

Input : f a differentiable convex function with Hölder continuous gradient; initial value of the "inexact" Lipschitz continuity constant L_0; initial point x_0; accuracy ε; number of iterations T.

$y_0 \leftarrow x_0$, $z_0 \leftarrow x_0$, $\alpha_0 \leftarrow 0$, $\psi_0(x) \leftarrow V_{x_0}(x)$

for $k = 0$ *to* $T - 1$ **do**

    $L_{k+1} \leftarrow \frac{L_k}{2}$

    while *True* **do**

        $v_k = \underset{x \in \mathbb{R}^n}{\operatorname{argmin}} \psi_k(x)$

        $\alpha_{k+1} \leftarrow \frac{1}{2L_{k+1}} + \sqrt{\frac{1}{4L_{k+1}^2} + \alpha_k^2 \frac{L_k}{L_{k+1}}}$

        $\tau_k \leftarrow \frac{1}{\alpha_{k+1} L_{k+1}}$

        $x_{k+1} \leftarrow \tau_k v_k + (1 - \tau_k) y_k$

        $z_{k+1} \leftarrow \underset{z \in \mathbb{R}^n}{\operatorname{argmin}} \, \alpha_{k+1} \langle \nabla f(x_{k+1}), z - v_k \rangle + V_{v_k}(z)$

        $y_{k+1} \leftarrow \tau_k z_{k+1} + (1 - \tau_k) y_k$

        if $f(y_{k+1}) \leqslant f(x_{k+1}) + \langle \nabla f(x_{k+1}), y_{k+1} - x_{k+1} \rangle + \frac{L_{k+1}}{2} \|y_{k+1} - x_{k+1}\|^2 + \frac{\tau_k \varepsilon}{2}$

        then

          | **break**

        end

        else

          | $L_{k+1} \leftarrow 2L_{k+1}$

        end

    end

    $\psi_{k+1}(x) \leftarrow \psi_k(x) + \alpha_{k+1} \left[f(x_{k+1}) + \langle \nabla f(x_{k+1}), x - x_{k+1} \rangle \right]$

end

return y_T

The above method does not require a priori knowledge of the smoothness parameter ν or the corresponding M_ν. The following theorem gives the convergence rate of the above algorithm:

THEOREM 1.2 *Let f be a differentiable convex function with Hölder continuous gradient with some exponent ν and $M_\nu < \infty$. Let $L_0 \leqslant M(\varepsilon, \nu, M_\nu)$. Then*

$$f(y_k) - f(x^*) \leqslant \left[\frac{2^{2+4\nu} M_\nu^2}{\varepsilon^{1-\nu} k^{1+3\nu}} \right]^{\frac{1}{1+\nu}} + \frac{\varepsilon}{2}.$$

What follows is that one may obtain an ε-accurate solution in

$$k \leqslant \inf_{\nu \in [0,1]} \left[\left(\frac{2^{\frac{3+5\nu}{2}} M_\nu}{\varepsilon} \right)^{\frac{2}{1+3\nu}} \left(\frac{1}{2} \|x_0 - x^*\|^2 \right)^{\frac{1+\nu}{1+3\nu}} \right]$$

iterations. If the problem admits multiple solutions, then x^* may be considered to be the solution minimizing $\frac{1}{2}\|x_0 - x^*\|^2$. As shown in [9], this is optimal up to a multiplicative constant independent of the accuracy, the initial point, and the objective function.

2. Universal Linear Coupling Method

We are now ready to present our universal method based on the linear coupling method proposed by Allen-Zhu et al. [1] The Linear Coupling framework is chosen as a basis for our method because it allows for the usage of an exact line search step, which is our goal. The original linear coupling method utilizes gradient and mirror descent steps to guarantee optimal convergence rate for convex objectives. However, it is clear from the convergence analysis of said method that the gradient step is only used to obtain a lower bound on the decrease of the objective during this step. This means that any procedure capable of guaranteeing at least such a decrease may be utilized instead. Since in the unconstrained Euclidean setting the gradient step is always performed in the direction of the negative of the gradient, one may use the steepest descent method instead. This idea combined with the idea of Nesterov's universal method allows us to modify the Linear Coupling method in the following way:

Algorithm 2: $\text{ULCM}(f, L_0, x_0, \varepsilon, T)$

Input : f a differentiable convex function with Hölder continuous gradient; initial value of the "inexact" Lipschitz continuity constant L_0; initial point x_0; accuracy ε; number of iterations T.

$y_0 \leftarrow x_0$, $z_0 \leftarrow x_0$, $\alpha_0 \leftarrow 0$

for $k = 0$ *to* $T - 1$ **do**

    $L_{k+1} \leftarrow \frac{L_k}{2}$

    while *True* **do**

        $\alpha_{k+1} \leftarrow \frac{1}{2L_{k+1}} + \sqrt{\frac{1}{4L_{k+1}^2} + \alpha_k^2 \frac{L_k}{L_{k+1}}}$

        $\tau_k \leftarrow \frac{1}{\alpha_{k+1} L_{k+1}}$

        $x_{k+1} \leftarrow \tau_k z_k + (1 - \tau_k) y_k$

        $h_{k+1} \leftarrow \underset{h \geqslant 0}{\text{argmin}}\, f(x_{k+1} - h\nabla f(x_{k+1}))$

        $y_{k+1} \leftarrow x_{k+1} - h_{k+1}\nabla f(x_{k+1})$

        $z_{k+1} \leftarrow z_k - \alpha_{k+1}\nabla f(x_{k+1})$

        if $\langle \alpha_{k+1}\nabla f(x_{k+1}), z_k - z_{k+1} \rangle - \frac{1}{2}\|z_k - z_{k+1}\|^2 \leqslant$ $\alpha_{k+1}^2 L_{k+1}(f(x_{k+1}) - f(y_{k+1}) + \frac{\tau_k \varepsilon}{2})$ **then**

          | **break**

        end

        else

          | $L_{k+1} \leftarrow 2L_{k+1}$

        end

    end

end

return y_T

As far as it is known to the authors of this paper, this is the first universal method of non-smooth optimization utilizing steepest descent steps.

From this point onwards L_k will always denote the value obtained at the end of a full iteration of the "for" loop.

We shall now show that the above algorithm is well-defined. To be more precise, we shall prove that the if-condition inside the while loop is satisfied after a finite number of iterations for any k.

LEMMA 2.1 $f(x)$ is a convex differentiable function and its gradient satisfies the Hölder condition for some $\nu \in [0,1]$ with some constant M_ν. Then for all steps k of above algorithm

$$\alpha_{k+1}\langle \nabla f(x_{k+1}), z_k - z_{k+1} \rangle - \frac{1}{2}\|z_k - z_{k+1}\|^2 \leqslant \alpha_{k+1}^2 L_{k+1}\left(f(x_{k+1}) - f(y_{k+1}) + \frac{\tau_k \varepsilon}{2}\right),$$

for all L_{k+1} satisfying

$$L_{k+1} \geqslant M(\tau_k \varepsilon, \nu, M_\nu) = \left[\frac{1-\nu}{1+\nu}\frac{M_\nu}{\tau_k \varepsilon}\right]^{\frac{1-\nu}{1+\nu}} M_\nu.$$

Proof.

$$\alpha_{k+1}\langle \nabla f(x_{k+1}), z_k - z_{k+1} \rangle - \frac{1}{2}\|z_k - z_{k+1}\|^2$$
$$\leqslant \frac{\alpha_{k+1}^2}{2}\|\nabla f(x_{k+1})\|^2 \leqslant M\alpha_{k+1}^2\left(f(x_{k+1}) - f(y_{k+1}) + \frac{\tau_k \varepsilon}{2})\right)$$

Here the first inequality follows from the fact that $\|\alpha_{k+1}\nabla f(x_{k+1}) - (z_k - z_{k+1})\|^2 \geqslant 0$. To get the last inequality we will use LEMMA 1.1 with $\delta = \tau_k \varepsilon$ and $x = x_{k+1}$, $y = x_{k+1} - \beta \nabla f(x_{k+1})$:

$$f(y) \leqslant f(x_{k+1}) + \langle \nabla f(x_{k+1}), -\beta \nabla f(x_{k+1}) \rangle + \frac{\beta^2 M}{2}\|\nabla f(x_{k+1})\|^2 + \frac{\tau_k \varepsilon}{2}$$
$$= f(x_{k+1}) - \beta\|\nabla f(x_{k+1})\|^2 + \frac{\beta^2 M}{2}\|\nabla f(x_{k+1})\|^2 + \frac{\tau_k \varepsilon}{2}.$$

Minimising the right-hand side over $\beta \in \mathbb{R}$, we get $\beta = \frac{1}{M}$. This results in the following guarantee:

$$f(y) - f(x_{k+1}) \leqslant -\frac{\|\nabla f(x_{k+1})\|^2}{2M} + \frac{\tau_k \varepsilon}{2}.$$

In our algorithm

$$y_{k+1} = x_{k+1} - h_{k+1}\nabla f(x_{k+1}),$$
$$h_{k+1} = \operatorname*{argmin}_{h \geqslant 0} f(x_{k+1} - h\nabla f(x_{k+1})),$$

6

so

$$f(y_{k+1}) - f(x_{k+1}) \leqslant f(y) - f(x_{k+1}) \leqslant -\frac{\|\nabla f(x_{k+1})\|^2}{2M} + \frac{\tau_k \varepsilon}{2}.$$

∎

2.1 Comparison with the UFGM method

Note that in the case of Euclidean norm and $V_x(y) = \frac{1}{2}\|x - y\|^2$, in the UFGM algorithm the mirror descent step

$$z_{k+1} \leftarrow \operatorname*{argmin}_{z \in \mathbb{R}^n} \alpha_{k+1}\langle \nabla f(x_{k+1}), z - v_k \rangle + V_{v_k}(z)$$

may be rewritten as

$$z_{k+1} \leftarrow v_k - \alpha_{k+1} \nabla f(x_{k+1}).$$

Moreover, in the case of the Euclidean norm the sequence $\{v_k\}$ turns out to be identical to the sequence $\{z_k\}$. Now by direct substitution of z_{k+1} and by using $(1 - \tau_k)y_k = x_{k+1} - \tau_k v_k$ we get that

$$y_{k+1} = \tau_k(z_k - \alpha_{k+1} \nabla f(x_{k+1})) + (1 - \tau_k)y_k = x_{k+1} - \frac{1}{L_{k+1}} \nabla f(x_{k+1}).$$

This means that the two methods are not just very similar, but are practically identical. The only difference between them is the usage of exact line search instead of a fixed-length gradient descent step.

2.2 Convergence Analysis

To ascertain the convergence of the above algorithm we will require the following lemmas:

LEMMA 2.2 *For any $u \in \mathbb{R}^n$*

$$\alpha_{k+1}\langle \nabla f(x_{k+1}), z_k - u \rangle \leqslant \alpha_{k+1}^2 L_{k+1}\left(f(x_{k+1}) - f(y_{k+1}) + \frac{\tau_k \varepsilon}{2}\right) + \frac{1}{2}\|z_k - u\|^2 - \frac{1}{2}\|z_{k+1} - u\|^2.$$

Proof.

$$
\begin{aligned}
\alpha_{k+1}\langle \nabla f(x_{k+1}), z_k - u \rangle &= \alpha_{k+1}\langle \nabla f(x_{k+1}), z_k - z_{k+1} \rangle + \alpha_{k+1}\langle \nabla f(x_{k+1}), z_{k+1} - u \rangle \\
&\overset{①}{=} \alpha_{k+1}\langle \nabla f(x_{k+1}), z_k - z_{k+1} \rangle + \langle z_k - z_{k+1}, z_{k+1} - u \rangle \\
&\overset{②}{=} \alpha_{k+1}\langle \nabla f(x_{k+1}), z_k - z_{k+1} \rangle + \frac{1}{2}\|z_k - u\|^2 - \frac{1}{2}\|z_{k+1} - u\|^2 - \frac{1}{2}\|z_k - z_{k+1}\|^2 \\
&\overset{③}{\leqslant} \alpha_{k+1}^2 L_{k+1}\left(f(x_{k+1}) - f(y_{k+1}) + \frac{\tau_k \varepsilon}{2}\right) + \frac{1}{2}\|z_k - u\|^2 - \frac{1}{2}\|z_{k+1} - u\|^2.
\end{aligned}
$$

7

Here, ① is due to

$$z_{k+1} = \operatorname*{argmin}_{z \in \mathbb{R}^n} \langle \alpha_{k+1} \nabla f(x_{k+1}), z \rangle + \frac{1}{2} \|z_k - z\|^2,$$

which implies

$$\nabla \left(\frac{1}{2} \|z_k - z\|^2 + \langle \alpha_{k+1} \nabla f(x_{k+1}), z \rangle \right) \bigg|_{z = z_{k+1}} = 0.$$

② follows from the triangle equality of Bregman divergence

$$\langle -\nabla V_x(y), y - u \rangle = V_x(u) - V_y(u) - V_x(y),$$

which takes the following form when $V_x(y) = \frac{1}{2}\|x - y\|^2$:

$$\langle x - y, y - u \rangle = \frac{1}{2}\|x - u\|^2 - \frac{1}{2}\|y - u\|^2 - \frac{1}{2}\|x - y\|^2$$

Finally, ③ is due to our choice of L_{k+1}. ∎

LEMMA 2.3   *For any* $u \in \mathbb{R}^n$

$$\alpha_{k+1}^2 L_{k+1} f(y_{k+1}) - \left(\alpha_{k+1}^2 L_{k+1} - \alpha_{k+1} \right) f(y_k) +$$
$$\left(\frac{1}{2}\|z_{k+1} - u\|^2 - \frac{1}{2}\|z_k - u\|^2 \right) - \frac{\alpha_{k+1}\varepsilon}{2} \leqslant \alpha_{k+1} f(u).$$

Proof. We deduce the following sequence of relations:

$$\alpha_{k+1}(f(x_{k+1}) - f(u)) \leqslant \alpha_{k+1}\langle \nabla f(x_{k+1}), x_{k+1} - u \rangle$$
$$= \alpha_{k+1}\langle \nabla f(x_{k+1}), x_{k+1} - z_k \rangle + \alpha_{k+1}\langle \nabla f(x_{k+1}), z_k - u \rangle$$
$$\stackrel{①}{=} \frac{(1 - \tau_k)\alpha_{k+1}}{\tau_k}\langle \nabla f(x_{k+1}), y_k - x_{k+1} \rangle + \alpha_{k+1}\langle \nabla f(x_{k+1}), z_k - u \rangle$$
$$\stackrel{②}{\leqslant} \frac{(1 - \tau_k)\alpha_{k+1}}{\tau_k}(f(y_k) - f(x_{k+1})) + \alpha_{k+1}^2 L_{k+1}\left(f(x_{k+1}) - f(y_{k+1}) + \frac{\tau_k\varepsilon}{2} \right)$$
$$+ \frac{1}{2}\|z_k - u\|^2 - \frac{1}{2}\|z_{k+1} - u\|^2 \stackrel{③}{=} (\alpha_{k+1}^2 L_{k+1} - \alpha_{k+1})f(y_k) - \alpha_{k+1}^2 L_{k+1} f(y_{k+1})$$
$$+ \alpha_{k+1} f(x_{k+1}) + \left(\frac{1}{2}\|z_k - u\|^2 - \frac{1}{2}\|z_{k+1} - u\|^2 \right) + \frac{\alpha_{k+1}\varepsilon}{2}.$$

Here, ① uses the fact that our choice of x_{k+1} satisfies $\tau_k(x_{k+1} - z_k) = (1 - \tau_k)(y_k - x_{k+1})$. ② is by convexity of $f(\cdot)$ and LEMMA 2.2, while ③ uses the choice of $\tau_k = \frac{1}{\alpha_{k+1}L_{k+1}}$. ∎

We are now ready to begin our proof of the method's convergence.

THEOREM 2.4 *Let $f(x)$ be a convex, differentiable function such that its gradient satisfies the Hölder condition for some $\nu \in [0,1]$ with some finite M_ν. Let L_0 also satisfy*

$$L_0 \leqslant \inf_{\nu \in [0,1]} 4 \left[\frac{1-\nu}{1+\nu} \frac{M_\nu}{\varepsilon} \right]^{\frac{1-\nu}{1+\nu}} M_\nu.$$

Then $ULCM(f, L_0, x_0, \varepsilon, T)$ outputs y_T such that $f(y_T) - f(x^*) \leqslant \varepsilon$ in the number of iterations

$$T \leqslant \inf_{\nu \in [0,1]} \left[\frac{1-\nu}{1+\nu} \right]^{\frac{1-\nu}{1+3\nu}} \left[\frac{2^{\frac{3+5\nu}{2}} M_\nu}{\varepsilon} \right]^{\frac{2}{1+3\nu}} \Theta^{\frac{1+\nu}{1+3\nu}},$$

where Θ is any upper bound on $\frac{1}{2}\|x_0 - x^*\|^2$.

Proof. Note that our choice of α_{k+1} satisfies

$$\alpha_{k+1}^2 L_{k+1} - \alpha_{k+1} = \alpha_k^2 L_k, \tag{1}$$

which allows us to telescope LEMMA 2.3. Summing up LEMMA 2.3 for $k = 0, 1, \ldots, T-1$ and $u = x^*$, we obtain

$$\alpha_T^2 L_T f(y_T) + \left(\frac{1}{2}\|z_T - x^*\|^2 - \frac{1}{2}\|z_0 - x^*\|^2 \right) \leqslant \sum_{k=1}^{T} \alpha_k f(x^*) + \sum_{k=1}^{T} \frac{\alpha_k \varepsilon}{2}.$$

By using (1) we get that $\sum_{k=1}^{T} \alpha_k = \alpha_T^2 L_T$. We also notice that $\frac{1}{2}\|z_t - x^*\|^2 \geqslant 0$ and $\frac{1}{2}\|z_0 - x^*\|^2 \leqslant \Theta$. Therefore,

$$f(y_T) - f(x^*) \leqslant \frac{\Theta}{\alpha_T^2 L_T} + \frac{\varepsilon}{2}.$$

Note that our process of calculating L_k guarantees that if the step $L_{k+1} \leftarrow 2L_{k+1}$ of the algorithm was executed at least once for some k, then for that k

$$L_{k+1} \leqslant 2 \left[\frac{1-\nu}{1+\nu} \frac{M_\nu}{\varepsilon \tau_k} \right]^{\frac{1-\nu}{1+\nu}} M_\nu. \tag{2}$$

Assume that $L_n \leqslant 2 \left[\frac{1-\nu}{1+\nu} \frac{M_\nu}{\varepsilon \tau_{n-1}} \right]^{\frac{1-\nu}{1+\nu}} M_\nu$ and $L_{n+1} = \frac{L_n}{2}$ for some $n \geqslant 1$. Then

$$\frac{1}{\tau_n} = \alpha_{n+1} L_{n+1} = \frac{1}{2} + \sqrt{\frac{1}{4} + \alpha_n^2 L_n L_{n+1}} \geqslant \frac{1}{2} + \sqrt{\frac{1}{4} + \frac{\alpha_n^2 L_n^2}{2}} \geqslant \frac{1}{\sqrt{2}\tau_{n-1}}.$$

9

$$L_{n+1} = \frac{L_n}{2} \leqslant \left[\frac{1-\nu}{1+\nu} \frac{\sqrt{2}M_\nu}{\varepsilon\tau_n} \right]^{\frac{1-\nu}{1+\nu}} M_\nu \leqslant 2 \left[\frac{1-\nu}{1+\nu} \frac{M_\nu}{\varepsilon\tau_n} \right]^{\frac{1-\nu}{1+\nu}} M_\nu.$$

This shows that even if we don't execute the step $L_{k+1} \leftarrow 2L_{k+1}$, (2) remains true as long as it held true on the previous iteration. All of the above proves that the assumption about L_0 in the statement of the theorem implies that (2) is true for all $k = 0, \ldots T - 1$.

Denote $A_k = \alpha_k^2 L_k$. We may now proceed to attain a lower bound on A_T.

$$\frac{\alpha_k^2}{A_k} = \frac{1}{L_k} \geqslant \frac{1}{2M_\nu} \left[\frac{1+\nu}{1-\nu} \frac{\varepsilon}{M_\nu} \right]^{\frac{1-\nu}{1+\nu}} \left[\frac{\alpha_k}{A_k} \right]^{\frac{1-\nu}{1+\nu}}$$

$$\alpha_k \geqslant \frac{1}{2^{\frac{1+\nu}{1+3\nu}} M_\nu^{\frac{2}{1+3\nu}}} \left[\frac{1+\nu}{1-\nu}\varepsilon \right]^{\frac{1-\nu}{1+3\nu}} A_k^{\frac{2\nu}{1+3\nu}}.$$

Denote $\gamma = \frac{1+\nu}{1+3\nu} \geqslant \frac{1}{2}$. Since $A_{k+1} = A_k + \alpha_{k+1}$,

$$A_{k+1}^\gamma - A_{k+1}^\gamma \geqslant \frac{A_{k+1} - A_k}{A_{k+1}^{1-\gamma} + A_k^{1-\gamma}} \geqslant \frac{\alpha_{k+1}}{2A_{k+1}^{1-\gamma}} \geqslant \frac{1}{2^{\frac{2+4\nu}{1+3\nu}} M_\nu^{\frac{2}{1+3\nu}}} \left[\frac{1+\nu}{1-\nu}\varepsilon \right]^{\frac{1-\nu}{1+3\nu}}. \qquad (3)$$

Now we telescope (3) for $k = 0, \ldots, T - 1$ and get

$$A_T \geqslant \left[\frac{1+\nu}{1-\nu} \right]^{\frac{1-\nu}{1+\nu}} \frac{T^{\frac{1+3\nu}{1+\nu}} \varepsilon^{\frac{1-\nu}{1+\nu}}}{2^{\frac{2+4\nu}{1+\nu}} M_\nu^{\frac{2}{1+\nu}}}.$$

This allows us to estimate the number of iterations necessary to achieve error no more than ε. However, beforehand we shall note that this estimate heavily depends on ν. By allowing M_ν to be infinite, we make the gradient of any differentiable function satisfy the Hölder condition for all $\nu \in [0, 1]$. This in turn allows to easily select the most appropriate estimate:

$$T \leqslant \inf_{\nu \in [0,1]} \left[\frac{1-\nu}{1+\nu} \right]^{\frac{1-\nu}{1+3\nu}} \left[\frac{2^{\frac{3+5\nu}{2}} M_\nu}{\varepsilon} \right]^{\frac{2}{1+3\nu}} \Theta^{\frac{1+\nu}{1+3\nu}}.$$

Note that since the solution x^* was arbitrary, x^* may now be considered to be the solution which minimizes $\frac{1}{2}\|x_0 - x^*\|^2$.

■

2.3 Stopping criterion

In [2] it is shown that the original version of the Linear Coupling Method may be equipped with a stopping criterion. By using similar techniques, we are now going to show that our universal modification of said method may also be equipped with a calculable stopping criterion.

By ignoring the first inequality in the proof of LEMMA 2.3, we get that for all $u \in \mathbb{R}^n$ (remember that $A_k = \alpha_k^2 L_k$)

$$A_{k+1} f(y_{k+1}) - A_k f(y_k) + \frac{1}{2} \|z_{k+1} - u\|^2 - \frac{1}{2} \|z_k - u\|^2 - \frac{\alpha_{k+1} \varepsilon}{2}$$
$$\leqslant \alpha_{k+1} \left(f(x_{k+1}) + \langle \nabla f(x_{k+1}), u - x_{k+1} \rangle \right).$$

Summing up for $k = 0, \ldots, m - 1$, we obtain

$$f(y_m) \leqslant \frac{\varepsilon}{2} + \frac{1}{A_m} \min_{u \in \mathbb{R}^n} \left\{ \frac{1}{2} \|z_0 - u\|^2 + \sum_{i=1}^m \alpha_i \left(f(x_i) + \langle \nabla f(x_i), u - x_i \rangle \right) \right\}.$$

Denote

$$l_m(u) = \sum_{i=1}^m \left[\alpha_i \left(f(x_i) + \langle \nabla f(x_i), u - x_i \rangle \right) \right]$$

and

$$\hat{f}_m = \min_{u: \; \frac{1}{2} \|z_0 - u\|^2 \leqslant \Theta} \frac{1}{A_m} l_m(u).$$

Then by using strong duality one may see that

$$\hat{f}_m = \min_{u \in \mathbb{R}^n} \max_{\lambda \geqslant 0} \left\{ \frac{1}{A_m} l_m(u) + \lambda \left(\frac{1}{2} \|z_0 - u\|^2 - \Theta \right) \right\}$$
$$= \max_{\lambda \geqslant 0} \min_{u \in \mathbb{R}^n} \left\{ \frac{1}{A_m} l_m(u) + \lambda \left(\frac{1}{2} \|z_0 - u\|^2 - \Theta \right) \right\}.$$

By setting $\lambda = \frac{1}{A_m}$, we get that

$$\hat{f}_m \geqslant \frac{1}{A_m} \min_{u \in \mathbb{R}^n} \left\{ \frac{1}{2} \|z_0 - u\|^2 + \sum_{i=1}^m \alpha_i \left(f(x_i) + \langle \nabla f(x_i), u - x_i \rangle \right) \right\} - \frac{\Theta}{A_m}.$$

Then $f(y_m) - \hat{f}_m \leqslant \frac{\varepsilon}{2} + \frac{\Theta}{A_m}$. This means that our method is primal-dual. By the convexity of f we also get that $f(x^*) \geqslant \hat{f}_m$, so $f(y_m) - f(x^*) \leqslant f(y_m) - \hat{f}_m \leqslant \varepsilon$ may be used as an implementable stopping criterion. Of course, an estimate of Θ is required to compute $\hat{f}_m$. Overestimating Θ may lead to performing an excessive amount of iterations, while underestimating it invalidates the criterion completely. However, the

stopping criterion requires an estimate of only one unknown parameter, which is also not used in the algorithm's definition. On the other hand, three unknown parameters (ν, M_ν, Θ) need to be estimated to calculate the upper bound on the number of iterations required to get an ε-accurate solution

3. Line search

During all of the previous analysis we assumed that $\forall x \in \mathbb{R}^n\ f(x),\ \nabla f(x)$, the steepest descent step, and the mirror descent step may be calculated exactly. However, in relation to the steepest descent step this assumption is not critical for the method's convergence.

For any convex function of one real argument defined on a segment of the form $[a, b]$ of length $l = b - a$ a point y such that

$$\left\| y - \operatorname*{argmin}_{x \in [a,b]} f(x) \right\| \leqslant \varepsilon$$

may be found in $O(\log \frac{l}{\varepsilon})$ function value calculations by using the bisection method. However, to perform an exact line search in our algorithm one needs to localize the solution first. To do that we propose the following simple procedure:

Algorithm 3: Localize(f,l_0)

Input : $f(x)$ – convex function defined on $[0, +\infty)$; initial segment length l_0.
Output: l such that $\operatorname*{argmin}_{x \in [0,+\infty)} f(x) \in [0, l]$

$l \leftarrow l_0$
while $f(2l) \leqslant f(l)$ **do**
| $l \leftarrow 2l$
end
return l

Let us estimate the accuracy with which the steepest descent must be performed to guarantee our method's convergence. Let's say we want to get a solution with accuracy of $\varepsilon + \delta$, where δ is the term resulting from the inaccuracy of the steepest descent step. To do that we need to slightly modify our algorithm:

Algorithm 4: δ-ULCM$(f,\ L_0,\ x_0,\ \varepsilon,\ \delta,\ T)$

Input : f a differentiable convex function with Hölder continuous gradient; initial value of the "inexact" Lipschitz continuity constant L_0; initial point x_0; accuracy ε; line search accuracy δ; number of iterations T.

$y_0 \leftarrow x_0,\ z_0 \leftarrow x_0,\ \alpha_0 \leftarrow 0$

for $k = 0 \to T - 1$ **do**

    $L_{k+1} \leftarrow \frac{L_k}{2}$

    while *True* **do**

        $\alpha_{k+1} \leftarrow \frac{1}{2L_{k+1}} + \sqrt{\frac{1}{4L_{k+1}^2} + \alpha_k^2 \frac{L_k}{L_{k+1}}}$

        $\tau_k \leftarrow \frac{1}{\alpha_{k+1}L_{k+1}}$

        $x_{k+1} \leftarrow \tau_k z_k + (1 - \tau_k)y_k$

        Choose y_{k+1} such that $f(y_{k+1}) \leqslant \underset{h \geqslant 0}{\operatorname{argmin}}\, f(x_{k+1} - h\nabla f(x_{k+1})) + \frac{\tau_k \delta}{2}$

        $z_{k+1} \leftarrow \underset{z \in \mathbb{R}^n}{\operatorname{argmin}}\, \langle \alpha_{k+1}\nabla f(x_{k+1}), z - z_k \rangle + \frac{1}{2}\|z_k - z\|^2$

        if $\langle \alpha_{k+1}\nabla f(x_{k+1}), z_k - z_{k+1} \rangle - \frac{1}{2}\|z_k - z_{k+1}\|^2 \leqslant$

        $\alpha_{k+1}^2 L_{k+1}(f(x_{k+1}) - f(y_{k+1}) + \frac{\tau_k \varepsilon}{2})$ **then**

            | **break**

        end

        else

            | $L_{k+1} \leftarrow 2L_{k+1}$

        end

    end

end

return y_T

THEOREM 3.1 *Let $f(x)$ be a convex, differentiable function such that its gradient satisfies the Hölder condition for some $\nu \in [0,1]$ with some finite M_ν. Let L_0 also satisfy*

$$L_0 \leqslant \inf_{\nu \in [0,1]} 4\left[\frac{1 - \nu}{1 + \nu}\frac{M_\nu}{\varepsilon}\right]^{\frac{1-\nu}{1+\nu}} M_\nu.$$

Then δ-ULCM($f,\ L_0,\ x_0,\ \varepsilon,\ \delta,\ T$) outputs y_T such that $f(y_T) - f(x^*) \leqslant \varepsilon + \delta$ in the number of iterations

$$T \leqslant \inf_{\nu \in [0,1]} \left[\frac{1-\nu}{1+\nu}\right]^{\frac{1-\nu}{1+3\nu}} \left[\frac{2^{\frac{3+5\nu}{2}}M_\nu}{\varepsilon}\right]^{\frac{2}{1+3\nu}} \Theta^{\frac{1+\nu}{1+3\nu}},$$

where Θ is any upper bound on $\frac{1}{2}\|x_0 - x^*\|^2$.

This immediately follows from the proof of THEOREM 2.4. To see that, note that if for some L_{k+1} and the exact solution of the line search problem $\hat{y}_{k+1}$

$$\langle \alpha_{k+1}\nabla f(x_{k+1}), z_k - z_{k+1} \rangle - V_{z_k}(z_{k+1}) \leqslant \alpha_{k+1}^2 L_{k+1}\left(f(x_{k+1}) - f(\hat{y}_{k+1}) + \frac{\tau_k \varepsilon}{2}\right)$$

13

holds true, then by definition of y_{k+1} we have

$$\langle \alpha_{k+1} \nabla f(x_{k+1}), z_k - z_{k+1} \rangle - V_{z_k}(z_{k+1}) \leqslant \alpha_{k+1}^2 L_{k+1} \left(f(x_{k+1}) - f(y_{k+1}) + \frac{\tau_k(\varepsilon + \delta)}{2} \right).$$

This leads to an analogue of LEMMA 2.1. Then by proceeding with the proof the same way it was done in THEOREM 2.4, one gets the desired result.

3.1 *Simplified function evaluation during line search*

As noted in [7], for objectives of particular form the steepest descent step may be performed significantly faster.

Consider a function of the form

$$f(x) = \phi(\mathbf{A}x) + \psi(x),$$

where $x \in \mathbb{R}^n$, $\mathbf{A} \in \mathbb{R}^{n \times n}$.

If n is sufficiently large, the computation of $\mathbf{A}x$ may be the most time-consuming operation during computation of $f(x)$. However, if we are performing the steepest descent step, we can be sure that x is of the form $x_k + \alpha \nabla f(x_k)$. Then

$$\mathbf{A}x = \mathbf{A}x_k + \alpha \mathbf{A} \nabla f(x_k) = v_0 + \alpha v_1.$$

This shows that one may calculate the two points v_0 and v_1 just once at the beginning of a steepest descent step.

If $\psi(y)$ and $\phi(y)$ with y known may be calculated in $\mathcal{O}(n)$ arithmetic operations, then this representation of $\mathbf{A}x$ allows us to evaluate $f(x)$ in $\mathcal{O}(n)$ arithmetic operations after performing matrix multiplication, which requires $\mathcal{O}(n^2)$ arithmetic operations, only twice. This may significantly decrease the cost of one steepest descent step.

4. Numerical experiments

The proposed methods were implemented in C++ and tested using the modern versions of GCC, clang and ICC (Intel C Compiler) on both GNU/Linux, Mac OS X and Microsoft Windows operating systems. The source code is available at http://github.com/htower/ulcm.

For the presented computational experiments we have also implemented a variant of the conjugate gradients method proposed by Y. Nesterov in [10], which we denote as NCG. The method has high numerical stability and a number of interesting properties. In particular, it lacks a restart procedure. This results in an increased iteration complexity relatively to "classic" conjugate gradient methods, which may be attributed to the necessity of solving two line search problems at each iteration. Details are presented in Algorithm 5 and Figure 1.

The behaviour of the proposed methods was investigated by a series of numerical experiments on different smooth and non-smooth optimization problems. For all experiments we set the starting point x_0 to $10 \cdot e$, where $e = (1, ..., 1)$, and the precision value $\varepsilon = 10^{-4}$.

Algorithm 5: NCG(f, x_0, δ, T)

Input : f a differentiable convex function with Hölder continuous gradient; initial
point x_0; line search accuracy δ; number of iterations T.

$y_{-2} \leftarrow x_0$, $y_{-1} \leftarrow x_0$, $y_0 \leftarrow x_0$
for $k = 0$ *to* $T - 1$ **do**
$\quad \alpha_k \leftarrow \underset{\alpha \in \mathbb{R}}{\operatorname{argmin}} f(x_k + \alpha(y_{k-2} - x_k))$
$\quad y_k = x_k + \alpha_k(y_{k-2} - x_k)$
$\quad \beta_k \leftarrow \underset{\beta \geq 0}{\operatorname{argmin}} f(y_k - \beta \nabla f(y_k))$
$\quad x_{k+1} = y_k - \beta_k \nabla f(y_k)$
end
return x_T

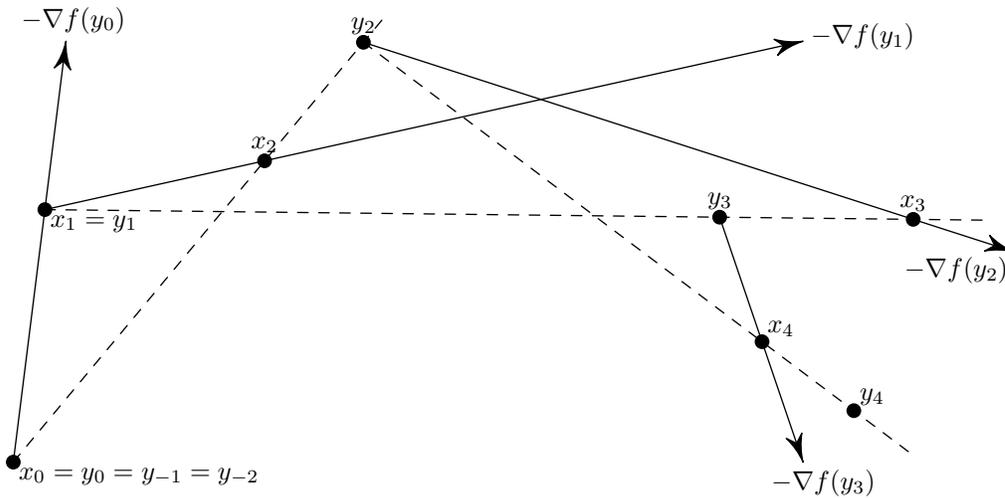Figure 1. Illustration of the NCG method.

n, problem size	$f(x_0)$	UFGM		ULCM		NCG	
		iterations	t, sec.	iterations	t, sec.	iterations	t, sec.
10^3	$5 \cdot 10^7$	743	0.035	722	0.035	121	0.004
10^4	$5 \cdot 10^9$	3230	1.429	3459	3.233	385	0.079
10^5	$5 \cdot 10^{11}$	15231	141.2	18053	372.6	1217	2.796
10^6	$5 \cdot 10^{13}$	73185	6857	84117	22373	3850	98.40

Table 1. Method's complexity for the smooth problems.

The methods were interrupted as soon as the objective function's value became lower
than $f(x^*) + 5\varepsilon = f(x^*) + 5 \times 10^{-4}$. The dimensionality of the problem was up to 10^6 .
Firstly, we considered the following smooth (quadratic) problem:

$$f(x) = \sum_{i=1}^{n} ix_i^2. \tag{4}$$

This function is L-smooth, but the parameter L depends on the number of dimensions n
linearly. This minimization problem can be solved analytically, the optimal value $f(x^*)$
is equal to 0. The results of our experiments are presented in Table 1 and Figure 2.
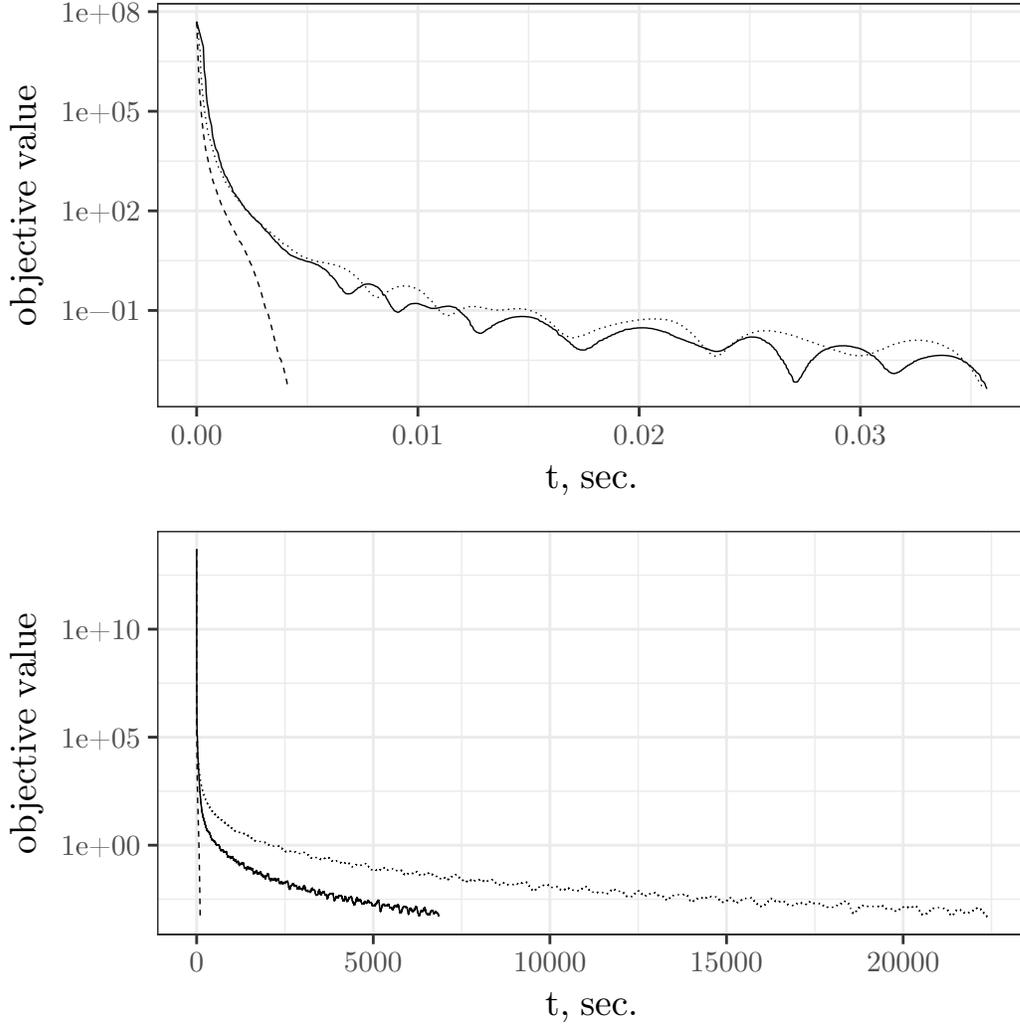
Figure 2. Methods convergence for the smooth problems with $n = 10^3$ (top) and $n = 10^6$ (bottom). The solid line stands for the UFGM method, the dotted line stands for the ULCM method, the dashed line stands for the NCG method.

Next, we consider the following non-smooth problem:

$$f(x) = \max_{i=1,\ldots,n} x_i + \frac{\mu}{2}\|x\|_2^2. \tag{5}$$

In our experiments $\mu = 0.1$. Though this function is differentiable almost everywhere. Though it does not have globally Hölder continuous gradients, the gradient satisfies the Hölder continuity condition on any bounded set.

This minimization problem can be solved analytically, the optimal value $f(x^*)$ is equal to $-\frac{1}{2\mu n} = -\frac{5}{n}$.

The gradient (subgradient, in case f is not differentiable at x) can be evaluated as

$$\nabla f(x) = \mu x + z(x), \quad z(x) = (0,\ldots 0, 1, 0, \ldots 0),$$

where 1 is located at position $k = \operatorname*{argmin}_{i=1,\ldots,n} x_i$.

16

n, problem size	$f(x_0)$	UFGM		ULCM	
		iterations	t, sec.	iterations	t, sec.
10^3	$1 \cdot 10^4$	535795	17.48	1376	0.175
10^4	$1 \cdot 10^5$	706870	233.8	6930	6.059
10^5	$1 \cdot 10^6$	1751285	4713	6950	34.18
10^6	$1 \cdot 10^7$	4341186	165435	6977	575.1

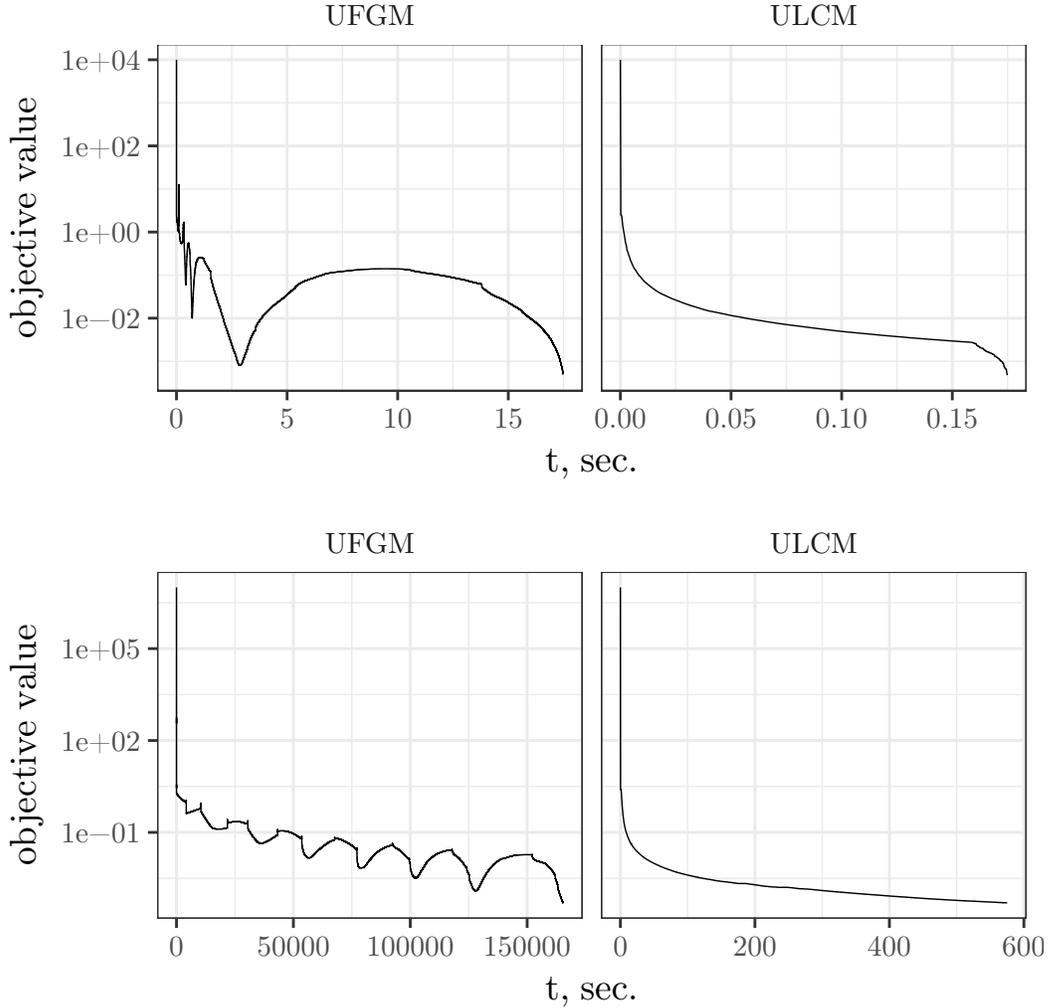Table 2. Method's complexity for the non-smooth problems.

Figure 3. Methods convergence for the non-smooth problems with $n = 10^3$ (top) and $n = 10^6$ (bottom).

The results are shown in Table 2 and Figure 3.

Note that in our particular case, since the ULCM and UFGM methods become identical if the steepest descent of the ULCM methods is replaced with a gradient descent step with step length $\frac{1}{L_{k+1}}$, all the differences in actual performance may be attributed to the line search procedure.

The results of our experiments may be summarized as follows:

(1) For the smooth problems (4) the NCG method showed best performance. Its conver-

gence rate significantly exceeds the convergence rates of UFGM and ULCM methods by up to two orders of magnitude. Although the ULCM method took less iterations to converge, it was slower (about 3 times) in terms of running time.

(2) For the non-smooth problems (5) the situation is opposite. In that case the ULCM method significantly outperformed UFGM, both in terms of required iterations and elapsed time. In the case of 10^6 arguments our method converged about 300 times faster.

Conclusions

In this paper we propose the first primal-dual method of non-smooth convex optimization with auxiliary line search. Practical experiments show that this method significantly outperforms Nesterov's Universal Fast Gradient Method [11]. Moreover, we prove that the presented method is also optimal for all the problems with intermediate level of smoothness. The advantage of such an approach is that one can generalize it to stochastic programming using mini-batches [6] and to gradient-free methods [5].

Acknowledgements

Funding

References

[1] Z. Allen-Zhu and L. Orecchia, *Linear coupling: An ultimate unification of gradient and mirror descent*, arXiv preprint arXiv:1407.1537 (2014).

[2] A. Anikin, A. Gasnikov, P. Dvurechensky, A. Tyurin, and A. Chernov, *Dual approaches to the minimization of strongly convex functionals with a simple structure under affine constraints*, Computational Mathematics and Mathematical Physics 57 (2017), pp. 1262–1276.

[3] O. Devolder, F. Glineur, and Y. Nesterov, *First-order methods of smooth convex optimization with inexact oracle*, Mathematical Programming 146 (2014), pp. 37–75, Available at https://doi.org/10.1007/s10107-013-0677-5.

[4] Y. Drori and A.B. Taylor, *Efficient first-order methods for convex minimization: a constructive approach*, arXiv preprint arXiv:1803.05676 (2018).

[5] P. Dvurechensky, A. Gasnikov, and A. Tiurin, *Randomized similar triangles method: A unifying framework for accelerated randomized optimization methods (coordinate descent, directional search, derivative-free method)*, arXiv preprint arXiv:1707.08486 (2017).

[6] A. Gasnikov and Y. Nesterov, *Universal fast gradient method for stochastic composit optimization problems*, arXiv preprint arXiv:1604.05275 (2016).

[7] G. Narkiss and M. Zibulevsky, *Sequential subspace optimization method for large-scale unconstrained problems*, Technion-IIT, Department of Electrical Engineering, 2005.

[8] A.S. Nemirovski and Y.E. Nesterov, *Optimal methods of smooth convex minimization*, Zhurnal Vychislitel'noi Matematiki i Matematicheskoi Fiziki 25 (1985), pp. 356–369.

[9] A. Nemirovskii and D.B. Yudin, *Problem complexity and method efficiency in optimization* (1983).

[10] Y. Nesterov, *Effective methods in nonlinear programming*, Radio and communication, 1989 (in Russian).

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

[12] B.T. Polyak, *Introduction to optimization. 1987*, Optimization Software, Inc, New York .