Contents

1 Introduction

In this paper we research one method of optimization on a square in $\mathbb{R}^2$. The method was offered by Nesterov (see ex. 4 from [1]).

In the paper [2] there are some results for this method. Namely, there are some estimates for iterations number and accuracy for task on segment (see the next section or [2]). Also there are comparison of this method with method of ellipsoids. Discussed method showed better results on time.

In this paper we want to continue this research. In the next section there is description of method and pseudocode for it. In the section 3. there are results where this method works correctly. The section 4. includes two ways to estimate accuracy for a task on segment and it is one of two main results in this paper. Following important result is theorem 5.3 for iterations number that is asymptotically twice as good as the estimate in [2].

Section Tests include different tests that show some theoretical results in practice and include comparison of this method with gradient descent.

2 Description Of Method

Let's consider a following task:

$$\min_{(x,y)} \{f(x,y) | (x,y) \in Q\},$$

where f is a convex function, Q - is a square on the plane.

Let's consider a following method. One solves task of minimization for a function $g(x) = f\left(x, y_0 = \frac{a}{2}\right)$ on a segment $[0, a]$ with an accuracy δ on function. After that one calculates a sub-gradient in a received point and chooses the rectangle which the sub-gradient "does not look" in. Similar actions are repeated for a vertical segment. As a result we have the square decreased twice. Let's find a possible value of error δ_0 for task on segment and a sufficient iteration's number N to solve the initial task with accuracy ϵ on function.

Let's describe an algorithm formally. See pseudo-code 1.

Algorithm 1 Algorithm of the method

function METHOD(convex function f, square $Q = [a, b] \times [c, d]$)
  $x_0 := solve(g = f(\cdot, \frac{c+d}{2}), [a, b], \delta)$
  $g = subgradient(f, (x_0, \frac{c+d}{2}))$
  if g[1] > 0 **then**
    $Q := [a, b] \times [c, \frac{c+d}{2}]$
  else
    $Q := [a, b] \times [\frac{c+d}{2}, d]$
  end if
  $y_0 := solve(g = f(\frac{a+b}{2}, \cdot), [c, d], \delta)$
  $g := subgradient(f, (\frac{a+b}{2}), y_0)$
  if g[0] > 0 **then**
    $Q := [a, \frac{a+b}{2}] \times [c, d]$
  else
    $Q := [\frac{a+b}{2}, b] \times [c, d]$
  end if
  if StopRec() == False **then**
    Method(f, Q)
  end ifreturn $(\frac{a+b}{2}, \frac{c+d}{2})$
end function

3 Algorithm correctness

Let's $\mathbf{x}_0$ is solution of the task on segment, Q_1 is choosed rectangle, Q_2 is not choosed rectangle.

3.1 Zero Error

Lemma 3.1. *If the optimization task on segment is solved with zero error and the f is convex and differentiable at a point-solution, rectangle with solution of initial task was choosed correct.*

Proof. From sub-gradient definition, $\mathbf{x}^* \in \{x|(\mathbf{g}(\mathbf{x}_0), \mathbf{x}_0 - \mathbf{x}^*) \geq 0)\}$. Lemma's statement follows from it and a fact that the first (or the second for vertical segment) gradient's component in point $\mathbf{x}_0$ is zero. □

3.2 Nonzero Error

Theorem 3.1. *Let's the f has continuous derivative on the square. Then there is a neighbourhood of a solution of optimization task on segment such as a choice of rectangle will not change if one use any point from the neighbourhood.*

Proof. Let's consider a case when we work with horizontal segment. Case with vertical segment is considered analogously. Then we are interesting in $f'_y(x_0, y_0)$. If $\mathbf{x}_0$ is not solution of initial task, then $f'_y(x_0, y_0) \neq 0$.

From a continuity of the derivative:

$$\lim_{\delta \to 0} f'_y(x_0 + \delta, y_0) = f'_y(x_0, y_0)$$

Therefore,

$$\exists \delta_0 : \forall \mathbf{x}_\delta \in B_{\delta_0}(x_0) \times y_0 \Rightarrow \operatorname{sign}(f'_y(\mathbf{x}_\delta)) = \operatorname{sign}(f'_y(\mathbf{x}_0))$$

From it and lemma 4.1 theorem's statement follows.

□

3.3 Undifferentiable convex function

The method does not work for all convex functions even for zero error on segment.

Example 1. There is an example in [2].

4 Error's Value

From derivative continuously we have following obvious result:

Lemma 4.1. *If f has continuous derivative. If $|f'_y(x, y_0)| > 0$ for all x on horizontal segment, then the second gradient's component has same sign at all points of segment. If $|f'_x(x_0, y)| > 0$ for all y on vertical segment, then the first gradient's component has same sign at all points of segment.*

Example 2. All functions f of the following type meet conditions of written above lemma:

$$f(x, y) = \psi(x) + \phi(y),$$

where ψ, ϕ are convex and differentiable functions.

Example 3. Let's illustrate that we can not always take any point from segment. Let's consider following task:

$$\min\left\{(x-y)^2 + x^2 \middle| Q = [0,1]^2\right\}$$

On segment $[0,1] \times \left\{\frac{1}{2}\right\}$ this task has solution $f^* = f\left(\frac{1}{4}, \frac{1}{2}\right) = \frac{1}{8}$. Derivative on y at this point is $f'_y\left(\frac{1}{4}, \frac{1}{2}\right) = \frac{]}{2}$ but at the point $\left(1, \frac{1}{2}\right)$ is equal to -1. We can see that in this case rectangle will selected non-correctly.

Let's find possible value of error's value, i.e. let's find a number δ_0 such as if an error for a solution on a segment is less δ_0 then rectangle for the segment is defined correctly. Rectangles are defined correctly for a horizontal optimization task, if:

$$\forall \delta : |\delta| < \delta_0 \Rightarrow f'_y(\mathbf{x}_0) f'_y(x_0 + \delta, y_0) > 0 \tag{1}$$

Analogically, for a vertical segment:

$$\forall \delta : |\delta| < \delta_0 \Rightarrow f'_x(\mathbf{x}_0) f'_x(x_0, y_0 + \delta) > 0 \tag{2}$$

Theorem 4.1. *Let function f be convex and has L-Lipschitz continuous gradient and a point $\boldsymbol{x}_0$ is a solution of optimization's task on a current segment.*

The current segment is horizontal and $\exists M > 0 :\Rightarrow |f'_y(\boldsymbol{x}_0)| \geq M$ or the current segment is vertical and $\exists M > 0 :\Rightarrow |f'_x(\boldsymbol{x}_0)| \geq M$. Then rectangle is defined correctly if the possible value of error is less than $\frac{M}{L}$.

Proof. Condition (1) is met if there is a derivative $f'_y(x_0 + \delta, y_0)$ in a neighbourhood of $f'_y(\mathbf{x}_0)$ with radius $\left|f'_y(\mathbf{x}_0)\right|$:

$$\left|f'_y(\mathbf{x}_0) - f'_y(x_0 + \delta, y_0)\right| < \left|f'_y(\mathbf{x}_0)\right|$$

The L-Lipschitz continuity gives following inequality:

$$\left|f'_y(\mathbf{x}_0) - f'_y(x_0 + \delta, y_0)\right| \leq L|\delta|$$

Therefore the following possible value is sufficient to select rectangle correctly:

$$\delta_0 < \frac{M}{L} \leq \frac{\left|f'_y(\mathbf{x}_0)\right|}{L}$$

Statement for vertical segment is proved similarly. □

In written above theorem the estimate needs some lower bound for derivatibe in point-solution on segment and this task can be hard in practise. Using other interpretation of the condition (1) one can take more useful result.

Theorem 4.2. *Let function f be convex and has L-Lipschitz continuous gradient and a point $\boldsymbol{x}_0$ is a solution of optimization's task on a current segment.*

The current segment is horizontal and $M = |f'_y(\boldsymbol{x}_{current})|$ or the current segment is vertical and $M = |f'_y(\boldsymbol{x}_{current})|$. Then rectangle is defined correctly if a distance between $\boldsymbol{x}_{cur}$ and accurate solution on segment is less than $\frac{M}{L}$.

Proof. Condition (1) is met if there is a derivative $f'_y(x_0 + \delta, y_0)$ in a neighbourhood of $f'_y(\mathbf{x}_0)$ with radius $\left|f'_y(\mathbf{x}_{cur})\right|$:

$$\left|f'_y(\mathbf{x}_0) - f'_y(\mathbf{x}_{cur})\right| < \left|f'_y(\mathbf{x}_{cur})\right|$$

The L-Lipschitz continuity gives following inequality:

$$\left|f'_y(\mathbf{x}_0) - f'_y(\mathbf{x}_{cur})\right| \leq L|\delta|$$

Therefore the following possible value is sufficient to select rectangle correctly:

$$\delta_0 < \frac{M}{L} \leq \frac{\left|f'_y(\mathbf{x}_0)\right|}{L}$$

Statement for vertical segment is proved similarly. □

Theorems 4.1 and 4.2 give stop conditions in the case when gradient in point-solution on segment and close points is large. But what should we do if gradient in this point is small?

Theorem 4.3. *Let f be convex and has L-Lipschitz continuous gradient. Then for accuracy on function ϵ following condition in point $\boldsymbol{x}$ is sufficient:*

$$\|\nabla f(\boldsymbol{x})\| \leq \frac{\epsilon}{a\sqrt{2}},$$

where a is size of current square.

Proof. Let's consider following inequality for convex functions (see prove in [3]):

$$f(\mathbf{x}^*) - f(\mathbf{x}) \geq (\nabla f(\mathbf{x}), \mathbf{x}^* - \mathbf{x})$$

Using Cauchy–Bunyakovsky–Schwarz inequality one has following inequality

$$f(\mathbf{x}) - f(\mathbf{x}^*) \leq -(\nabla f(\mathbf{x}), \mathbf{x}^* - \mathbf{x}) \leq$$
$$\leq \|\nabla f(\mathbf{x})\| \|\mathbf{x}^* - \mathbf{x}\| \leq \|\nabla f(\mathbf{x})\| a\sqrt{2}$$

This inequality proves theorem's statement. □

Let's make a couple of remarks.

Firstly, we can replace the Lipschitz condition on the square by the Lipschitz condition on the segments in the theorems 4.1 and 4.2.

Thirdly, all theorems in this section use f'_y on horizontal segment and f'_x on vertical segment. As a result, if one checks condition of stop on each iteration method will use full gradient on each iteration. It can slow down method but this estimates can be better than in [2]. As a result, method can work faster.

5 Number of iterations

Following estimates are correct if each iterations was correct (a rectangle is selected correctly on each iterations).

Theorem 5.1. *If function f is convex and L_f-Lipschitz continuous, then for to solve initial task with accuracy ϵ on function one has to take a center of a current square as approximate solution and make following iteration's numbers:*

$$N = \left\lceil \log_2 \frac{L_f a}{\sqrt{2}\epsilon} \right\rceil \tag{3}$$

where a is a size of the initial square Q.

This estimate is a little improved estimate from [2] because of we use a center of a current square as approximate solution. The prove is similar to prove of not improved estimate.

There are functions which estimates from written above theorem are very accurate for.

Example 5. Let's consider following task with positive constant A:

$$\min\left\{A(x+y)|Q=[0,1]^2\right\}$$

If one take a center of a current solution as approximate solution one have value $\frac{A}{2^N}$ after N iterations. Therefore, for accuracy ϵ one has to $\lceil \log_2 \frac{A}{\epsilon} \rceil$. For this function $L_f = 2A$. Therefore, estimate (3) is accurate for such tasks with little error that not more one iteration.

But any convex function is locally Lipschitz continuous at all $x \in \operatorname{int} Q$. Therefore, we have following theorem.

Theorem 5.2. *If function f is convex and a solution $x^* \in \operatorname{int} Q$, then for to solve initial task with accuracy ϵ on function one has to take a center of a current square as approximate solution and make following iteration's numbers:*

$$N = \left\lceil \log_2 \max\left\{\frac{a}{\epsilon_0(\boldsymbol{x}^*)}, \frac{L_f a}{\sqrt{2}\epsilon}\right\}\right\rceil \tag{4}$$

where a is a size of the initial square Q, $\epsilon_0(x^*)$ is size of neighbourhood of x^* which f is L_f-Lipschitz continuous in, $\Delta f = f(x_0) - f(x^*)$, x_0 is a center of square Q.

We can improve written above estimates if to add new conditions:

Theorem 5.3. *Let function f be convex and has L-Lipschitz continuous gradient.*

If solution is a internal point, then for to solve initial task with accuracy ϵ on function one has to take a center of a current square as approximate solution and make following iteration's numbers:

$$N = \left\lceil \frac{1}{2}\log_2 \frac{La^2}{4\epsilon}\right\rceil \tag{5}$$

where a is a size of the initial square Q.

Proof. For all convex functions there is following inequality (one may find proof in [3]):

$$f(\mathbf{x}) - f(\mathbf{x}^*) - (f'(\mathbf{x}^*), \mathbf{x} - \mathbf{x}^*) \leq \frac{L}{2}\|\mathbf{x} - \mathbf{x}^*\|^2$$

If $\mathbf{x}^*$ is a solution and an internal point, then $f'(\mathbf{x}^*) = 0$:

$$f(\mathbf{x}) - f(\mathbf{x}^*) \leq \frac{L}{2}\|\mathbf{x} - \mathbf{x}^*\|^2$$

After N iterations we have the estimate:

$$f(\mathbf{x}) - f(\mathbf{x}^*) \leq \frac{L}{4}\left(\frac{a}{2^N}\right)^2$$

Using it we have estimate (5). □

6 Tests

In this section we show estimate on number iterations of practice and compare work time our new method with work time classic methods of optimization such as gradient descent and method ellipsoid[1]. All code was made in Anaconda 5.3.1 Python 3.6 (see cite [4])

6.1 About function *solve*

If you look at pseudocode 1 you can find function *solve*. This function solves task of minimization on segment. A cost of one iteration depends on a choice of its implementation. In our tests we will use golden search selection. It is the best method for one-dimension task in our case because on each iteration the GSS need to calcualted value of function in only one new point. It is important because we will use this method for dual tasks and calculating function in such tasks are hard enough.

6.2 Tests for iterations number

Estimate 3 works in all cases and better then estimate 5 when following condition is met:

$$\frac{2L_f^2}{L_g} \leq \epsilon,$$

где L_f, L_g is Lipschitz constants for function and for gradient. On the other hand, the estimate 5 works if point with zero gradient exists in the square.

[1]You can find all code in the repository [7]

Let's consider some quadratic form:

$$f(x, y) = (Ax + By)^2 + Cy^2 + Dx + Ey + F,\ C > 0.$$

One minimize this function on square that involve global solution. We took parameteres randomly. Then we have following values of the estimates 3 and 5:

Theoretical Iteration Number through function constant equals 40

Theoretical Iteration Number through gradient constant equals 20

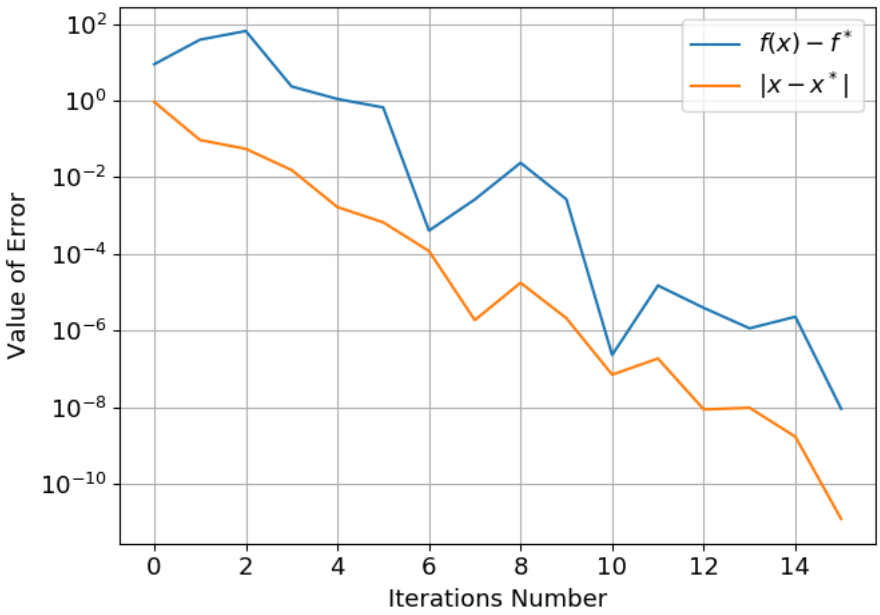

Figure 1: Test for iterations number when the estimate 5 is reached

We can see on 6.2 that the estimate 5 is reached in this case. But let's consider following task:

$$f(x, y) = (x + y)^2 + x^2, Q = [1, 2]^2,\ (x^*, y^*) = (1, 1)$$

$$\min_{(x,y)\in Q} f(x, y)$$

Theoretical Iteration Number through function constant 30
Theoretical Iteration Number through gradient constant 14

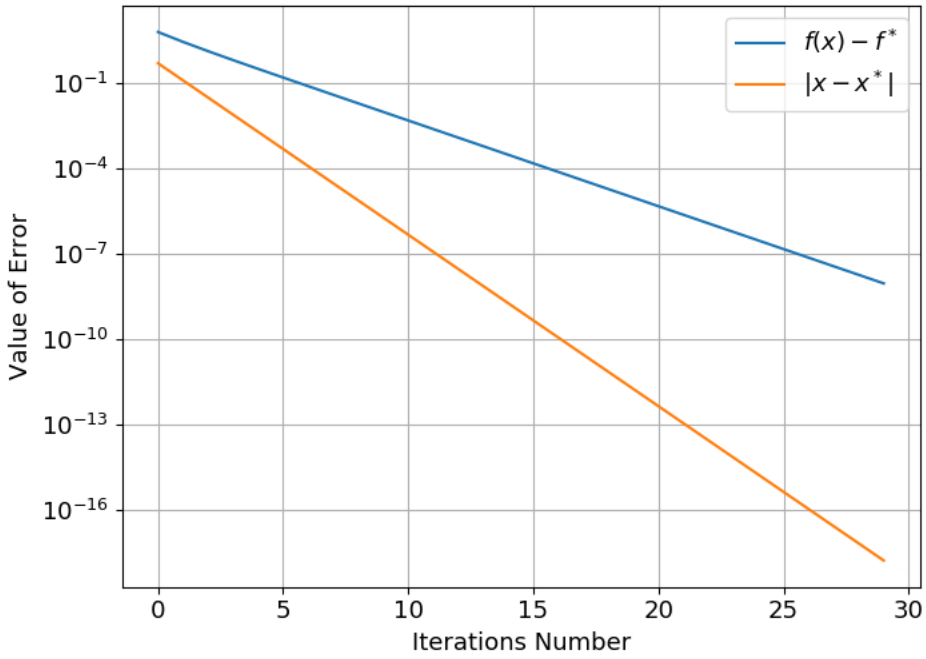

Figure 2: Test for iterations number when the estimate 5 is not reached

And we see on 6.2 that the estimate 5 is not reached in this case and in this case we should use estimate 3.

6.3 Comparison Of Stop Conditions

In current section we will compare different stop conditions for the solving task on separating segments. Fistly, let's define all stop conditions.

- Constant estimate (**Const**): $\delta \leq \frac{\epsilon}{2La\sqrt{5}\log_2 \frac{2Ma\sqrt{2}}{\epsilon}}$ from [2],
- Constant gradient estimate (**ConstGrad**): $\delta \leq \frac{M_{der}}{L}$ from 4.1,

- Estimate through current gradient (**CurGrad**): $\frac{|f'(\mathbf{x}_{cur})|}{L}$ from 4.2.

Obviously, that in practic we can not use the second strategy because to calculate this estimate for one segment is harder than to solve all task. This estimate is the improved the third estimate. And there is interesting question: how much better does the second strategy work than the third?

We will test on the simple quadratic functions. Parameters for this functions will be generated randomly. We will research dependence of work time on required accuracy ϵ. For each epsilon we will generate N tested functions and measure work time of each method. In our expiriements N is equal to 1000 times. Results of experimetns you can see on the picture 6.3.

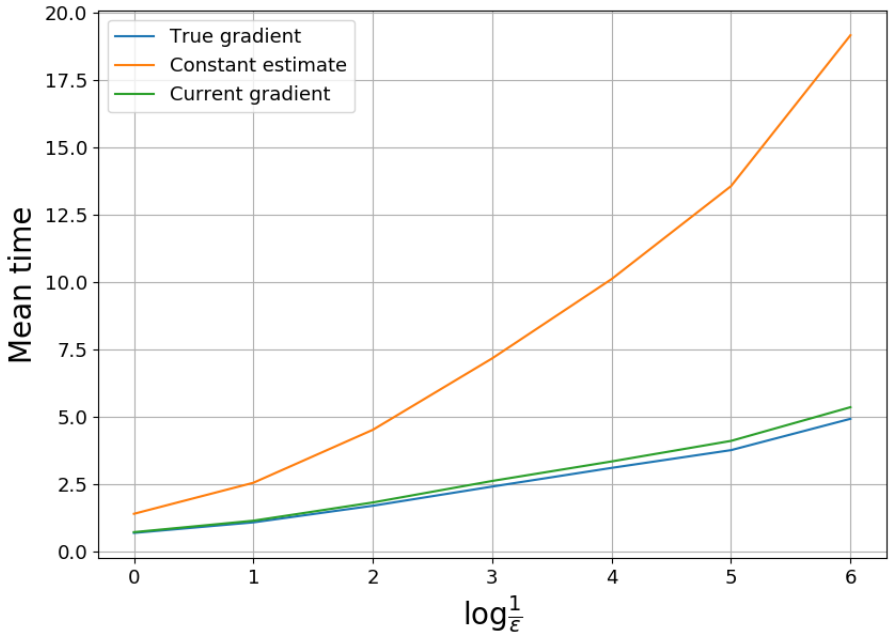

Figure 3: Comparison of different optimization method

6.4 Comparison With Other Methods

Let's compare out method with ellipsoid method and gradient descent with optimal constant step $\frac{1}{L_g}$ on some dual task.

We consider following prime task:

$$f(\mathbf{x}) = \ln\left(1 + \sum_{k=1}^{n} e^{a_k x_k}\right) + \|\mathbf{x}\|_2^2 \to \min_{\mathbf{x}} \tag{6}$$

$$\text{s.t. } g_1(\mathbf{x}) = \|\mathbf{x}\|_2 - R_1 \leq 0 \tag{7}$$

$$g_2(\mathbf{x}) = \|\mathbf{x}\|_2 - R_2 \leq 0 \tag{8}$$

We introduce the following notation:

$$\phi(\lambda_1, \lambda_2) = -\min_{\mathbf{x}} \left(f(\mathbf{x}) + \lambda_1 g_1(\mathbf{x}) + \lambda_2 g_2(\mathbf{x})\right) \tag{9}$$

In such notations the dual task for the task 6 looks like:

$$\phi(\lambda_1, \lambda_2) \to \min_{\lambda_1, \lambda_2} \tag{10}$$

$$\text{s.t } \lambda_1, \lambda_2 \geq 0 \tag{11}$$

According to [1] (see ex. 4.1), we can add following restraint for the dual variavles:

$$\|\lambda\|_1 \leq \frac{1}{\gamma}\left(f(\overline{\mathbf{x}}) - \min_{\mathbf{x}} f(\mathbf{x})\right),$$

$$\text{where } \overline{\mathbf{x}} : g_i(\overline{\mathbf{x}}) < 0, \gamma = \min_i g_i(\overline{\mathbf{x}})$$

Obviously, $\min_{\mathbf{x}} f(\mathbf{x}) \geq 0$. Therefore, we have following condition on the dual variables:

$$|\lambda_k| \leq \lambda_{\max} = \frac{f(\overline{\mathbf{x}})}{\gamma}, k = 1, 2$$

And we understand that there is the dual task's solution in a square $Q = [0, \lambda_{\max}]^2$. And we have following optimization task:

$$\phi(\lambda_1, \lambda_2) \to \min_{\lambda_1, \lambda_2} \tag{12}$$

$$\text{s.t.} \lambda \in Q \tag{13}$$

The gradient of function ϕ will be calculated according well-known Demyanov-Danskin-Rubinov Theorem, see [**?**].

Theorem 6.1. *Let* $\phi(\lambda) = \min\limits_{x \in X} F(x, \lambda)$ *for all* $\lambda \geq 0$*, where* F *is a smooth convex function with respect to* λ *and* $x(\lambda)$ *is the only maximimum point. Then*

$$\nabla \phi(\lambda) = F'_{\lambda}\left(x(\lambda), \lambda\right)$$

In the our case:

$$\phi'_{\lambda_k}(\lambda) = g_k\left(\mathbf{x}(\lambda)\right) \tag{14}$$

Additionally we need a Lipschitz constant for gradient. In the work [6] there is following theorem:

Theorem 6.2. *Let* $f(x)$ *be a* μ_f*-strongly convex function, the function* $g(x)$ *satisfies the Lipschitz condition with a constant* M_g*. Then the function* $\phi(\lambda)$ *defined in 9, where* $x(\lambda) = \arg\min\limits_{x}(f(x) + \lambda g(x))$*, has Lipschitz smooth gradient with constant* $L_\phi = \frac{M_g^2}{\mu_f}$

This theorem can be proved easy for the second dimension. In this case g is a vector-function. Obviously, in the our task function f is 2-strongly convex.

And now we can compare speed of different methods for the dual task. We will compare ellipsoids method, gradient descent and our method HalvingSquare with two stop-condition for the task on segment. Parameters $\mathbf{a}$ we will generate randomly.

We can see on 6.4 following results. Firstly, gradient descent is the slowest method in this test. Our method with estimate through current gradient is faster than method with constanst estimate. And our method approached more little value of dual task than ellipsoid method on suct time interval.

This experiment showed that our method is more effective on some dual task than some classic methods. Additionally, our method an estimate through current gradient is faster than with constant estimate.

7 Conclusion

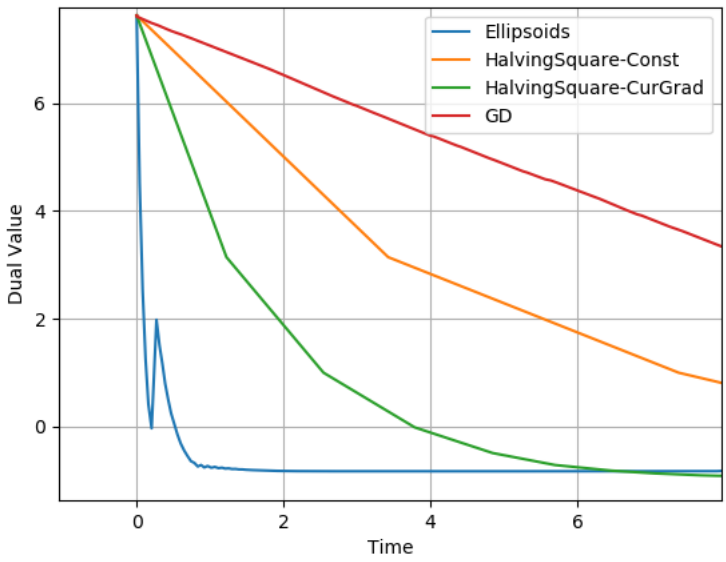

Figure 4: Comparison of different optimization method

References

[1] Gasnikov A. Universal gradient descent // MIPT — 2018, 240 p.

[2] Pasechnyk D.A., Stonyakin F.S. One method for minimization a convex Lipchitz continuous function of two variables on a fixed square // arXiv.org e-Print archive. 2018. – URL: https://arxiv.org/pdf/1812.10300.pdf

[3] Nesterov U.E. Methods of convex optimization // M.MCNMO — 2010, 262 p.

[4] Anaconda[site]. At available: https://www.anaconda.com

[5] Danskin, J.M.: The theory of Max-Min, with applications. J. SIAM Appl. Math.14(4) (1966)

[6] Fedor S. Stonyakin, Mohammad S. Alkousa, Alexander A. Titov,and Victoria V. Piskunova1 On Some Methods for Strongly ConvexOptimization Problems with OneFunctional Constraint // ...

[7] Repository with code: https://github.com/ASEDOS999/Optimization-Halving-The-Square