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 of gradient descent and our new method - halving square[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 gradient descent with step $h_k = \frac{h}{\sqrt{k}} = \frac{a}{4\sqrt{k}}$, where a is size of current segment.

6.2 Tests for iterations number

Let's make tests for the estimate (3).

Functions $-\sin\frac{\pi x}{a}$ and $-\sin\frac{\pi x}{b}$ are convex on square $Q = [0,1]^2$ when $a, b \geq 1$. Therefore, a function $-A\sin\frac{\pi x}{a} - B\sin\frac{\pi y}{b}$ is convex for all $A, B \geq 0$ as cone combination of convex function.

Functions x^n are convex and monotonously non-decreasing on $[0,1]$ for all $n \in \mathbb{N}$ that's why functions $\left(-A\sin\frac{\pi x}{a} - B\sin\frac{\pi y}{b} + A + B + D\right)^n$ are convex for all $D \geq 0$.

Therefore, following function is convex:

$$f(x,y) = -A_1\sin\frac{\pi x}{a_1} - B_1\sin\frac{\pi x}{b_1} + \sum_{n=2}^{N}\left(-A_n\sin\frac{\pi x}{a_n} - B_n\sin\frac{\pi y}{b_n} + A_n + B_n + D_n\right)^n,$$

[1] You can find all code in the repository [5]

where $A_i, B_i.D_i \geq 0$ and $a_i, b_i \geq 1$ for all $i = \overline{1,n}$.

The function f is differentiable infinite times and we can use it to test the method.

Let's take $a_1 = \cdots = a_n = a$ and $b_1 = \cdots = b_n = b$:

$$f(x,y) = -A_1 \sin\frac{\pi x}{a} - B_1 \sin\frac{\pi x}{b} + \sum_{n=2}^{N} \left(-A_n \sin\frac{\pi x}{a} - B_n \sin\frac{\pi y}{b} + A_n + B_n + D_n\right)^n,$$

where $A_i, B_i.D_i \geq 0$ for all $i = \overline{1,n}$ and $a, b \geq 1$.

We have following estimates for derivatives on horizontal and vertical segments:

$$|f'_x|\Big|_{x=x_0} \geq \left(A_1 + \sum_{n=2}^{N} nA_n D_n^{n-1}\right) \frac{\pi}{a} \left|\cos\frac{\pi x_0}{a}\right|$$

$$|f'_y|\Big|_{y=y_0} \geq \left(B_1 + \sum_{n=2}^{N} nB_n D_n^{n-1}\right) \frac{\pi}{b} \left|\cos\frac{\pi y_0}{b}\right|$$

Therefore this functions met conditions of lemma 4.1 and we can take gradient at any point of segment. One can see examples of functions f on fig. 1

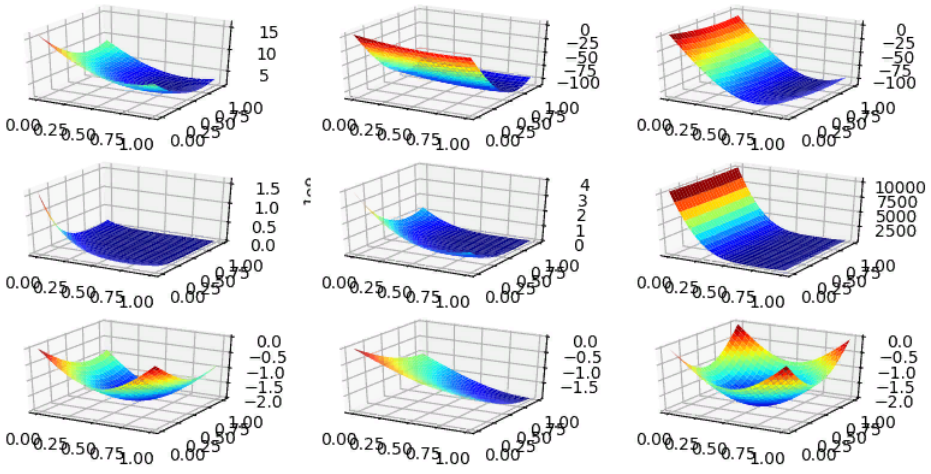

Figure 1: Examples of tests functions

We will use a and b from $[1, 2]$ and $N = 2$. Let's solve task of minimization function f with different parameters on square $[0, 1]^2$ through new method and compares number of iteration with estimate (3).

Result of the test you can see on fig. 2.

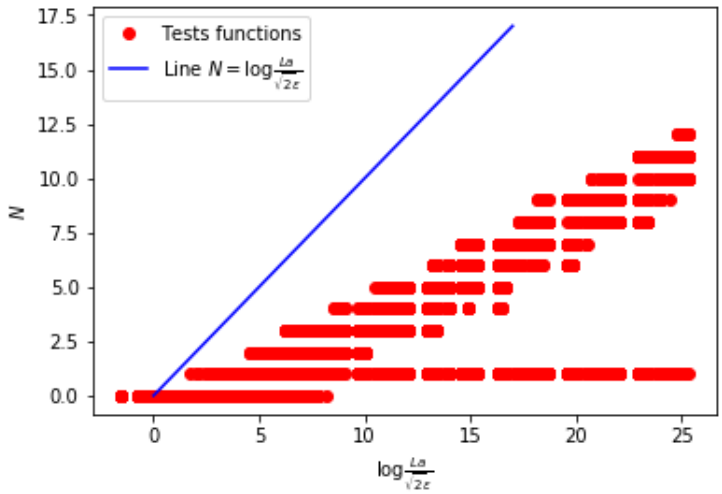

Figure 2: Tests' results for iterations number

On graphic N is number of done iterations for accuracy ϵ, $\log_2 \frac{La}{\sqrt{2}\epsilon}$ is the parameter of tests tasks. Line $N = \log_2 \frac{La}{\sqrt{2}\epsilon}$ is our estimates (3).

We can see that there are tests points under this line. It confirms our estimates (3).

6.3 Comparison

Let's compare experimentally work's time of gradient descent and method of halving square. We will compare it on optimization task for functions from previous subsection and random quadric function with positive semidifenite matrix:

$$f(x, y) = -A_1 \sin \frac{\pi x}{a} - B_1 \sin \frac{\pi x}{b} + \left(-A_2 \sin \frac{\pi x}{a} - B_2 \sin \frac{\pi y}{b} + A_2 + B_2 + D_2\right)^2, \tag{6}$$

where $A_i, B_i.D_i \geq 0$ for all $i = \overline{1, 2}$ and $a, b \in [1, 2]$.

$$f(x, y) = (Ax + By)^2 + Cx^2 + Dx + Ey + F, C \geq 0 \tag{7}$$

Halving square will stop when current value will be no further than ϵ from optimal value. Gradient descent will stop when distance between current and previous value will be not more than ϵ.

We will use gradient descent with a step $h_k = \frac{h}{\sqrt{k}} = \frac{a}{4\sqrt{k}}$, where a is size of the square. Also we add limit $N_{max} = 1000$ on the iterations number for method of halving square, solving on segment and gradient descent.

When we compare two method we distinguish the following situations (similar name of situations on graphics):

- Type 1 (T1) - tasks which gradient descent completed but halving square was not complete,
- Type 2 (T2) - gradient descent is faster than halving square,
- Type 3 (T3) - work's times are approximately equal or time is too short to measure,
- Type 4 (T4) - gradient descent is slower than halving square,
- Type 5 (T5) - gradient descent was not complete but halving square completed,
- Type 6 (T6) - both methods were not completed successfully

6.3.1 Sinuses

As already mentioned in previous subsection, functions (6) meet conditions of lemma 4.1, therefore, we can take any point from segment on each iteration in method of halving square. Using all written above we have following results (see fig. 3).

We can see that new method is not less efficient on the most tasks and more efficient on the task's half than gradient descent.

6.3.2 Quadric functions

Quadric functions do not always meet conditions of lemma 4.1, therefore, we can not think that error for task on segment can be very big. We can try to estimate this error but let's use for this task gradient descent with $|x - x_{prev}| < \delta$, where δ is much less than size of segment. In our experiments we will use $\delta = \frac{a}{400}$. Using it we have following results (see fig. 4).

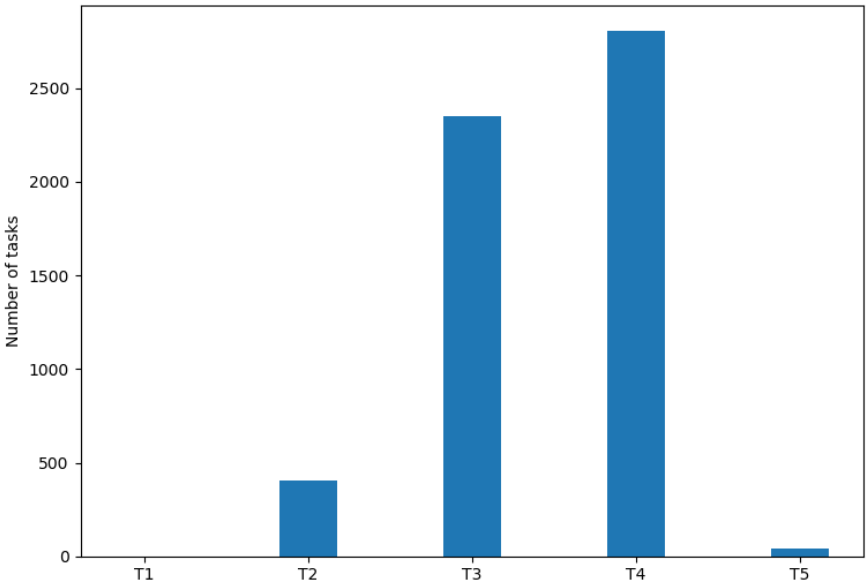

Figure 3: Tests' results for comparison on sinuses

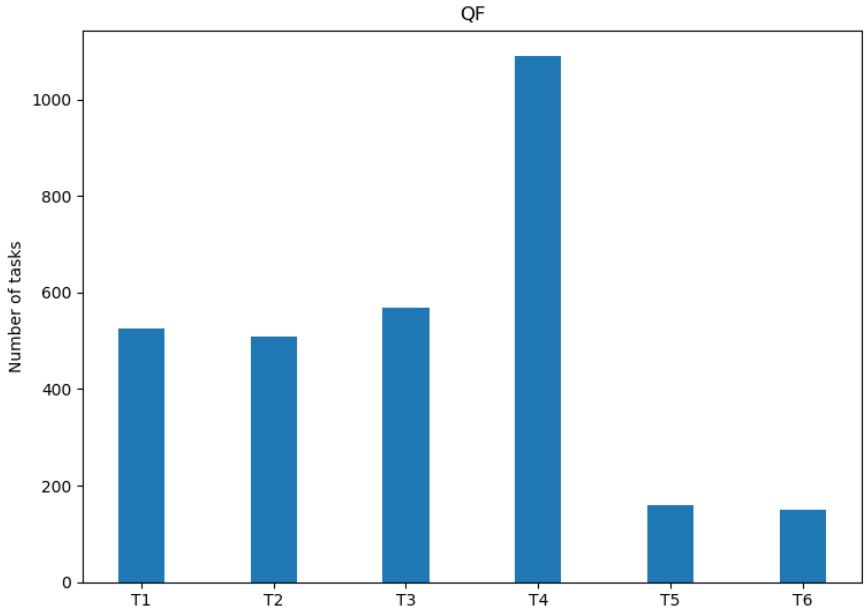

Figure 4: Tests' results for comparison on quadric function

We can see that halving square was not completed successfully on big parts of task - about $\frac{1}{6}$ from all tasks. We can decrease δ in its modification, but work's time increases in this case.

Also gradient descent was better than halving square in more than half tests. Increasing δ lead to decreasing work's time but also it can lead to that method will not completed successfully more often.

Therefore, halving square with proposed function *solve* and proposed stupid strategy for δ is not efficient.

6.4 Comparison Of Stop Conditions

6.5 Example: Least Square

7 Conclusion

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