FEDERAL STATE AUTONOMOUS EDUCATIONAL INSTITUTION
OF HIGHER EDUCATION
ITMO UNIVERSITY

Report
on the practical task No. 2

Algorithms for unconstrained nonlinear optimization. Direct methods

Performed by
Dmitry Grigorev, Maximilian Golovach
J4133c

Accepted by
Dr Petr Chunaev

St. Petersburg
2022

1. Goal of the work

The goal of this work is to consider direct methods of unconstrained nonlinear optimization problem. The methods are compared in terms of precision, number of required iterations and number of function calculations.

2. Formulation of the problem

The task consists of two subtasks according to dimensionality of optimization problem. The first subtask is devoted to one-dimensional optimization problem. Here we consider next three functions on their domains:

1. $f(x) = x^3,\ x \in [0, 1]$,
2. $f(x) = |x - 0.2|,\ x \in [0, 1]$,
3. $f(x) = x \sin\left(\frac{1}{x}\right),\ x \in [0.01, 1]$.

The problem here is to optimize these functions and find point of minimum x^* with precision $\varepsilon = 10^{-3}$ using the following three direct methods:

- exhaustive search,
- dichotomy method,
- golden section method.

The second subtask is associated with approximation of data in terms of least squares. Let $\alpha,\ \beta \in (0, 1)$ are two arbitrary values, $x_k = \frac{k}{100},\ k = 0, \dots, 100$ and y_k are defined by i.i.d. $\delta_k \sim N(0, 1)$ according to the following rule:

$$y_k = \alpha x_k + \beta + \delta_k.$$

Given such two parametrized families of functions as:

1. $F(x, a, b) = ax + b$,
2. $F(x, a, b) = \frac{a}{1+bx}$,

we need to find the optimal function for given data $(x_k, y_k)_{k=0}^{100}$ according to the least squares using

- exhaustive search,

- Gauss method (coordinate descent),
- Nelder-Mead method.

All functions which will be obtained have to be visualized with the given data and the line which generates these data. Furthermore, we have to compare algorithms used in each subtasks in terms of precision, number of iterations and number of function evaluations.

3. Brief theoretical part

3.1. Exhaustive search

Let $f : [a, b] \to \mathbb{R}$ is a scalar function of one argument and $\varepsilon > 0$ is precision. The exhaustive search algorithm is organised as follows:

1. Take an integer n such that $\frac{b-a}{n} \leq \varepsilon$,
2. For each $k = 0, \ldots, n$ the value $f(x_k)$ is calculated where $x_k = a + k\frac{b-a}{n}$,
3. Then find $\widehat{x}$ among the set of x_k for which function f is minimal.

Directly from the algorithm construction the inequality $|\widehat{x} - x^*| < \varepsilon$ holds. Here x^* is the minimum point of f.

This approach can be extended to functions of several arguments likewise.

It is obvious that the time complexity of this algorithms is $O(\frac{1}{\varepsilon})$ in the case of one dimension and $O(\frac{1}{\varepsilon^2})$ in the case of two dimensions if the function evaluation time is supposed to be constant. This algorithm is very slow and is also called 'brute-force search' because it goes through all possible solutions and checks if every solution satisfies needed criteria.

3.2. Dichotomy method

Let $f : [a, b] \to \mathbb{R}$ is a convex function and we also have precision $\varepsilon > 0$. Take $\delta : \ 0 < \delta < \varepsilon$. The dichotomy method is constructed in the following way:

1. Put $a_0 = a, \ b_0 = b$;
2. Calculate $x_1 = \frac{a_0+b_0-\delta}{2}, \ x_2 = \frac{a_0+b_0+\delta}{2}$ and values $f(x_1)$ and $f(x_2)$;
3. Compare the function values:
   if $f(x_1) \leq f(x_2)$, then put $a_1 = a_0, \ b_1 = x_2$,
   otherwise, put $a_1 = x_1, \ b_1 = b_0$;

4. Repeat the algorithm with a_k and b_k whilst the condition $|a_k - b_k| < \varepsilon$ is met.

The time complexity of this algorithm is $O(\log(\frac{1}{\varepsilon}))$ if one suppose that the function evaluation time is constant.

3.3. Golden section method

The idea behind the method of golden section is to reduce the number of evaluations of optimized function: if the function evaluation is slow, it dramatically affects on the whole performance of optimizing algorithm. For example, in the method of dichotomy at each iteration we calculate the function value twice. As for the golden section method, it is an optimized dichotomy search algorithm that has a specific choice of parameter $\delta = \frac{3-\sqrt{5}}{2}$ which reduces function calculations quantity. Here is the description of the algorithm:

1. Put $a_0 = a, \ b_0 = b$;
2. Calculate $x_1 = a_0 + \delta(b_0 - a_0), \ x_2 = b_0 - \delta(b_0 - a_0)$ and values $f(x_1)$ and $f(x_2)$;
3. Compare the function values:
   if $f(x_1) \leq f(x_2)$, then put $a_1 = a_0, \ b_1 = x_2$ and $x_2 = x_1$. Then calculate x_1 with respect to the formula in the row 2. and $f(x_2)$,
   otherwise, put $a_1 = x_1, \ b_1 = b_0$ and $x_1 = x_2$. Then calculate x_2 with respect to the formula in the row 2. and $f(x_2)$;
4. Repeat the algorithm with a_k and b_k whilst the condition $|a_k - b_k| < \varepsilon$ is met.

This method's time complexity is also $O(\log(\frac{1}{\varepsilon}))$ (see [1]).

3.4. Gauss method

The Gauss method is a representative of the class of algorithms called Greedy algorithms. The idea of this algorithm is to fix all function arguments except one and solve one-dimensional optimization problem on the function section which corresponds to this varying argument.

Let $f = f(x, y)$ is a function of two variables and (x_0, y_0) is an initial approximation for point of minimum.

1. Fix the y-argument and optimize f with respect to x: $x_1 = \arg\min_x f(x, y_0)$.
2. Then fix the x-argument and optimize f with respect to y: $y_1 = \arg\min_y f(x_1, y)$.

3. Further we repeat these steps until one of the following conditions is met:

$$1)\ |x_{i+1} - x_i| < \varepsilon \ \&\ |y_{i+1} - y_i| < \varepsilon \quad \text{or} \quad 2)\ |f(x_{i+1}, y_{i+1}) - f(x_i, y_i)| < \varepsilon.$$

3.5. Nelder-Mead method

Nelder-Mead method is a heuristic approach and is built around simplexes. Simplex in N-dimensional space is a $(N+1)$ connected vertices. For example, in the case of 2 dimensional space it is triangle. First it is needed to choose 3 points for the initial simplex. Then sorting is performed by values of the function for chosen vertices. Further gravity center x_c is found for every point except point with biggest function value in it (x_h). Then point x_h is reflected with respect to the gravity center. Then algorithm performs either shinkage or expansion. And after the previous step convergence check is performed. If the mutual closeness of the vertices in simplex is small enough (i.e., required precision is achieved) to minimum point, then the solution is found. Otherwise, we continue the algorithm from the step of sorting the vertices.

The Nelder-Mead algorithm typically requires only one or two function evaluations at each step, while many other direct search methods use more function evaluations. But on the other hand Nelder-Mead algorithm can stuck in a local minimum or converge to non-stationary point.

4. Results

4.1. Subtask *I*

Consider the three aforementioned functions. Let the precision parameter $\varepsilon = 10^{-3}$.

1) $f(x) = x^3,\ x \in [0, 1]$

This function is the well-known cubic parabola which is convex on the given segment and has minimum value at the point $x^* = 0$ with the value $f(x^*) = 0$. Because of the convexity, all considered methods are applicable.

The exhaustive search method with $n = \frac{1}{\varepsilon} + 1 = 1001$ provides the exact solution $\widehat{x} = 0$ but the costs for the obtained result are 1000 iterations and 1001 function evaluations. These are demonstrate the slowness of this algorithm.

As for the dichotomy method is concerned, its result is $\widehat{x} \approx 9.88 * 10^{-4}$. To obtain this result, this method required 10 iterations during which it evaluated function 21 times. This method is approximating, so it does not have to find exact solution but it finds approximate solution more quickly than exhaustive search (The former method took ≈ 50 times less iterations than the latter in this case).

The method of golden section found the approximate solution $\widehat{x} \approx 3.67 * 10^{-4}$ in 15 iterations with 18 evaluations of function. Although this method must do significantly less function evaluations than the dichotomy method (in this case the numbers are practically the same), it found more accurate solution than the latter.

2) $f(x) = |x - 0.2|, \; x \in [0, 1]$

This function also belongs to the class of well-known functions. This function is also convex It is evident that the minimum of this function is situated at the point $x^* = 0.2$. Here the application of the considered methods is correct since, as has been mentioned, this function is convex. Let us consider the results of these methods.

The exhaustive search method with the same $n = \frac{1}{\varepsilon} + 1 = 1001$ gives the exact solution $\widehat{x} = 0.2$ as this point belongs to the grid generated by ε on the segment $[0, 1]$ but the costs for the obtained result are 1000 iterations and 1001 function evaluations because of the method construction.

As for the dichotomy method, its result is $\widehat{x} \approx 0.20001$. To obtain this result, this method required 10 iterations during which it evaluated function 21 times. Here we again see the dramatic difference in f-evaluations and number of iterations between this method and the exhaustive search.

The latter considered method, the golden section method found a bit worse solution $\widehat{x} \approx 0.20007$ than the dichotomy method in 15 iterations with 18 evaluations of function.

3) $f(x) = x \sin\left(\frac{1}{x}\right), \; x \in [0.01, 1]$

Let us have a look at the graph, which is depicted in the figure 1, of this function on the given segment.

As one can see, this function is not convex and not even concave and has a number of local minima (and maxima) but its global minimum is located near the point $x = 0.22$. The application of the dichotomy and the golden section methods is not justified: these methods are not guaranteed to converge to the point of global minimum. Nevertheless, let us apply all these methods in experimental purposes.

The exhaustive search method with $n = \frac{0.99}{\varepsilon} + 1 = 991$ gives the solution $\widehat{x} = 0.223$

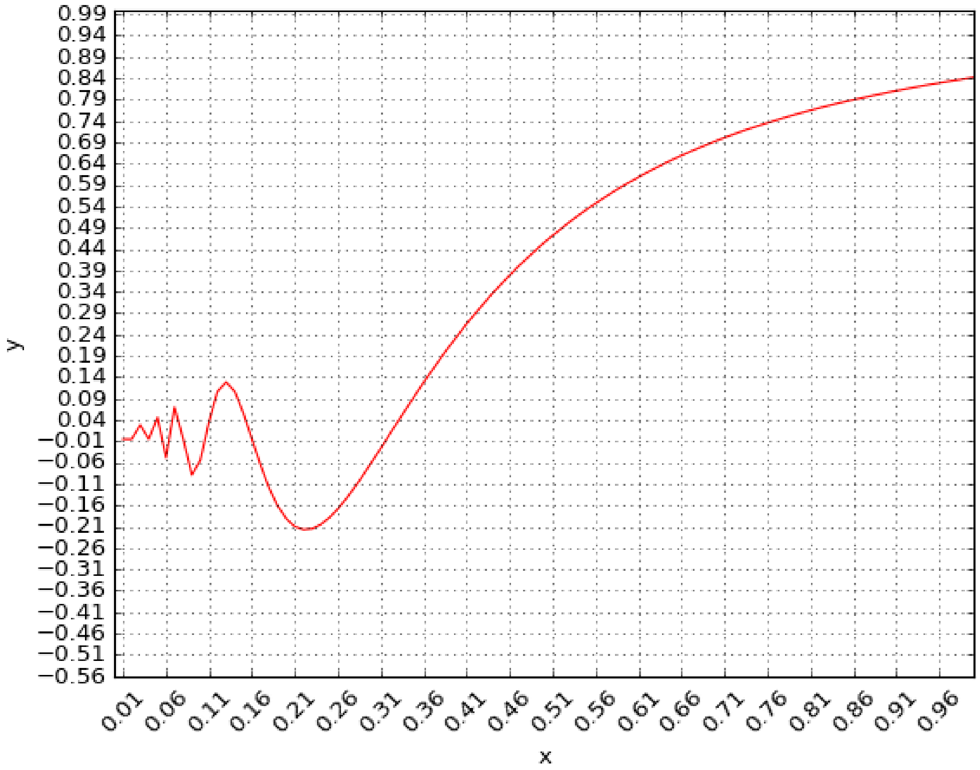

Figure 1: The graph of f(x)

in 990 iterations with 991 function evaluations. The solution is in fact close to the optimal point but the method again required a lot of iterations and function evaluations.

The dichotomy approach's solution is $\widehat{x} \approx 0.2225$ which was obtained in 10 iterations with 21 f-calculations (these indicators are also 50 times less than the exhaustive search's ones as for the two previous functions).

At last, the application of the golden section method provides a bit different solution $\widehat{x} \approx 0.2227$ which was obtained in 15 iterations with 18 evaluations of function.

To sum up the results of this section, we obtained 3 solutions for 3 functions. The exhaustive search method needs a lot of iterations to give the answer as opposed to the dichotomy method and the golden section method each of which obtain its result in ≈ 50 times less iterations with the same times less function evaluations.

4.2. Subtask *II*

We have the data as described in the problem formulation section 2. Let $\alpha = 0.785,\ \beta = 0.31$.

1) Approximation with $F(x,a,b) = ax + b$

The target function $D(a,b) = \sum_{k=0}^{100}(F(x_k,a,b) - y_k)^2$ has to be optimized by the exhaustive search, Gauss and Nelder-Mead methods with precision $\varepsilon = 10^{-3}$. The results of optimization are presented in the figure 2.

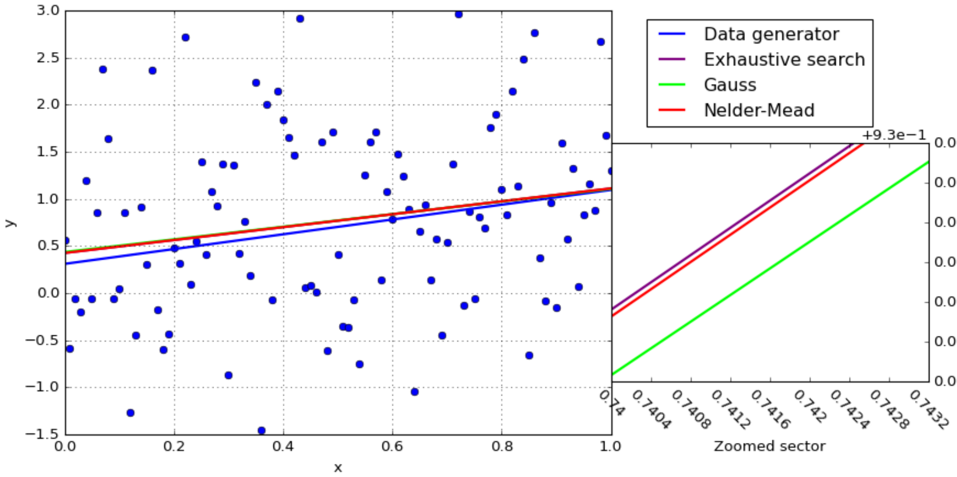

Figure 2: The data, the line which generated these data and fitted lines. Since the lines are too close, the additional plot with zoomed sector is provided on the right size of the figure.

As has been seen from the figure, the resulted lines are too close to each other so all methods converged successfully. Nevertheless, each of the methods required its own number of iterations, number of function evaluations and demonstrated its own precision in term of euclidian distance to the exact solution α. To be short, these indicators are provided in the table 1.

	Iterations	f-evaluations	precision
Exhaustive search	1002001	1002001	≈ 0.153
Gauss	286	546	≈ 0.171
Nelder-Mead	69	127	≈ 0.153

Table 1: Algorithms' indicators in the case of linear approximation.

From this table we can see that the most expensive in terms of iterations and function evaluation method is the exhaustive search what is the consequence from its construction. By the same time, the most cheap methods is the Nelder-Mead which required approximately 14000 times less iterations as opposed to the exhaustive search and approximately 4 times less iterations than Gauss method of coordinate descent. At the same time, this method provides the best precision among all considered methods.

2) Approximation with $F(x, a, b) = \frac{a}{1+bx}$

The target function $D(a, b) = \sum_{k=0}^{100}(F(x_k, a, b) - y_k)^2$ has to be optimized by the exhaustive search, Gauss and Nelder-Mead methods with precision $\varepsilon = 10^{-3}$ in order to find the best fitted rational function for given data. The results of optimization in this case are presented in the figure 3.

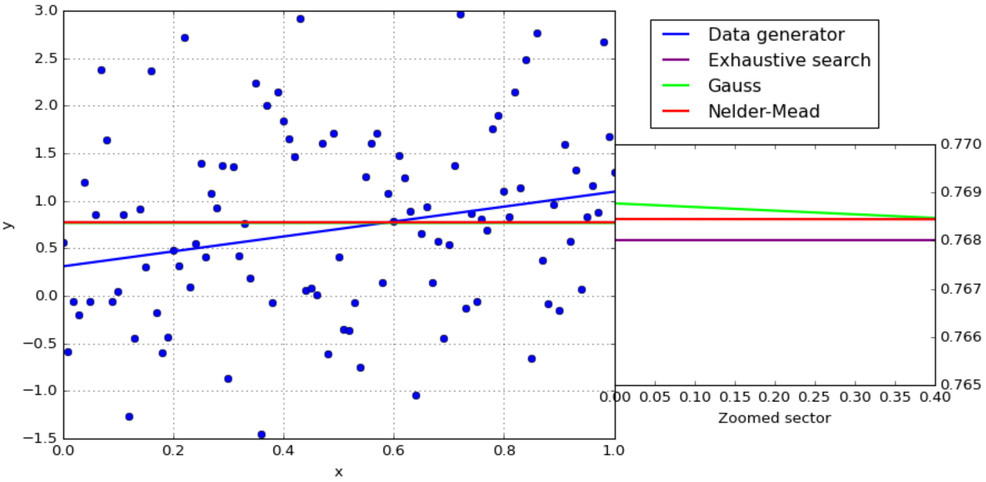

Figure 3: The data, the line which generated these data and fitted lines of rational functions. Since the lines are too close, the additional plot with zoomed sector is provided on the right size of the figure.

It follows from the figure that the resulted lines are too close to each other here again so all methods converged successfully. Also it is worth mentioning that all resulted hyperbolas are too close to the line $y = 0.768$ which right side of the equation approximately coincides with the mean of y. It looks logical since in the class of rational functions the curve with $a = 0$ corresponds to the mean which minimizes the mean square of bias:

$$\mathbb{E}X = \arg\min_{a\in\mathbb{R}} \mathbb{E}(X - a)^2.$$

As for the indicators which were enumerated above, they are presented in the table 2 for this case.

As has been seen, the exhaustive search again is the most expensive method. As for other two methods, they demonstrate practically the same both number of iterations, number of function evaluations and precision.

	Iterations	f-evaluations	precision
Exhaustive search	1002001	1002001	≈ 0.311
Gauss	77	147	≈ 0.309
Nelder-Mead	65	125	≈ 0.310

Table 2: Algorithms' indicators in the case of rational approximation.

5. Data structures and design techniques used in algorithms

In the implementation of the algorithms and for random variables generation the Python's package Numpy is used since it allows to vectorize calculations what accelerates the evaluations. The implementation of the Nelder-Mead algorithm is taken from the Scipy package.

6. Conclusion

As the result of this work, we considered direct methods of optimization in the cases of one and two variables. The methods were compared by the number of iterations required to obtain the solution with given precision, the number of function evaluations and precision. All of the considered methods except Nelder-Mead method were implemented in Python programming language. Discussion for data structures and design techniques which were used in the implementations is provided. The work goals were achieved.

7. Appendix

Algorithms implementation code is provided in [2].

Bibliography

1. Luenberger D. G., Ye Y. Linear and Nonlinear Programming. — Springer Publishing Company, Incorporated, 2015. — ISBN: 3319188410.
2. Grigorev D., Golovach M. Code repository. — `https://github.com/dmitry-grigorev/AlgoAnalysisDevelopment`. — 2022.