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
Carlos Andres Daza Rachen

Academic group
Accepted by
Dr Petr Chunaev

St. Petersburg
2022

Goal

The use of direct methods (one-dimensional methods of exhaustive search, dichotomy, golden section search; multidimensional methods of exhaustive search, Gauss (coordinate descent), Nelder-Mead) in the tasks of unconstrained nonlinear optimization

Formulation of the problem

1. Compare one-dimensional methods
2. Compare exhaustive search, Gauss and Nelder-Mead methods

Brief theoretical part

Brute-force algorithms like **exhaustive search** are these processes that reach the perfect solution to a problem by analysing all the possible candidate solutions. There are advantages and disadvantages to adopting such kind of approach. Usually, a brute-force approach is simple to implement, and it will always find a solution to the computational problem by considering iteratively all the possible solutions one by one [1].

Dichotomic search is a search algorithm that operates by selecting between two distinct alternatives (dichotomies) at each step. It is a specific type of divide and conquer algorithm. A well-known example is binary search [2].

Golden section search uses the golden ratio to approximate Fibonacci search o find an extremum value with narrowing the searching interval in a golden ratio (φ) range [3].

The **Nelder-Mead simplex method** uses a simplex to traverse the space in search of a minimum. A simplex is a generalization of a tetrahedron to n-dimensional space. A simplex in one dimension is a line, and in two dimensions it is a triangle (see figure 7.8). The simplex derives its name from the fact that it is the simplest possible polytope in any given space.

The **Newton-Gauss** method consists of linearizing the model equation using a Taylor series expansion around a set of initial parameter values b0, also called preliminary estimates, whereby only the first-order partial derivatives are consider.

Results

1. One-dimensional methods

1.1 x^3

	Iterations	Calls	x minimun	value
Exhaustive Search	1000	1000	0.00	0.00
Dichotomy Search	13	26	0.00	0.00
Golden Section	15	16	0.00	0.00

1.2 $|x - 0.2|$

	Iterations	Calls	x minimun	value
Exhaustive Search	1000	1000	0.2000	0.0000
Dichotomy Search	11	22	0.1997	0.0003
Golden Section	15	16	0.1997	0.0003

1.3 $sin(1/x)$

	Iterations	Calls	x minimun	value
Exhaustive Search	999	999	0.2230	-0.2172
Dichotomy Search	12	24	0.0419	-0.0403
Golden Section	15	16	0.2222	-0.2172

In the first two equations the results do not differ that much, the exhaustive search has the maximum number of iterations and the dichotomous search the least, the golden section is in the middle.

The golden section shows the most accurate results after the exhaustive search.

2.

2.1 Linear function

	Iterations	Value	x minimun	y
Exhaustive Search	32	113.038531	1.4398645	0.393673
Gauss Search	4	113.038640	1.4390630	0.393043
Nelder-Mead	29	113.038531	1.4400729	0.393611

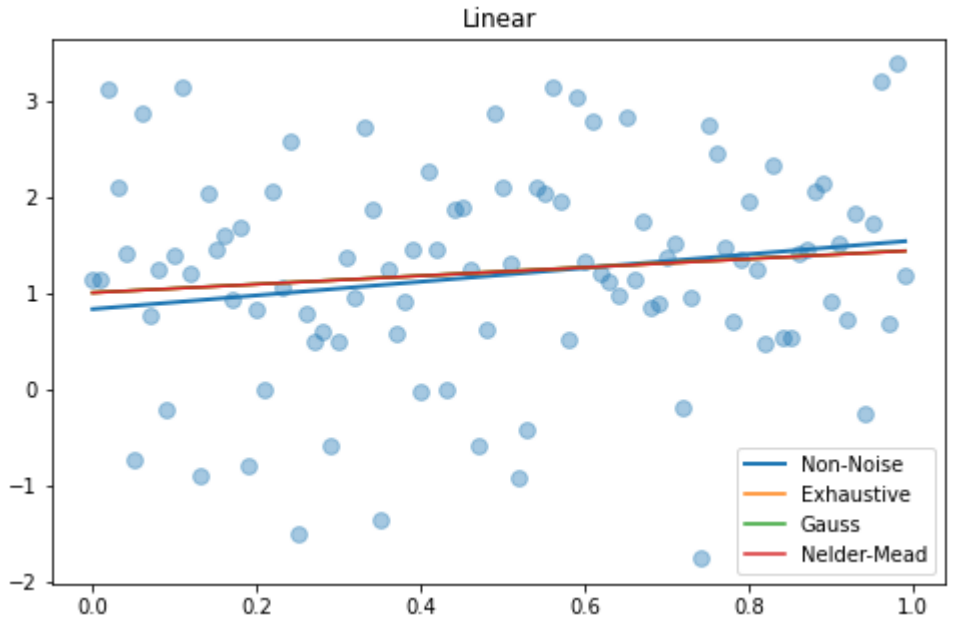
2.2 Rational function

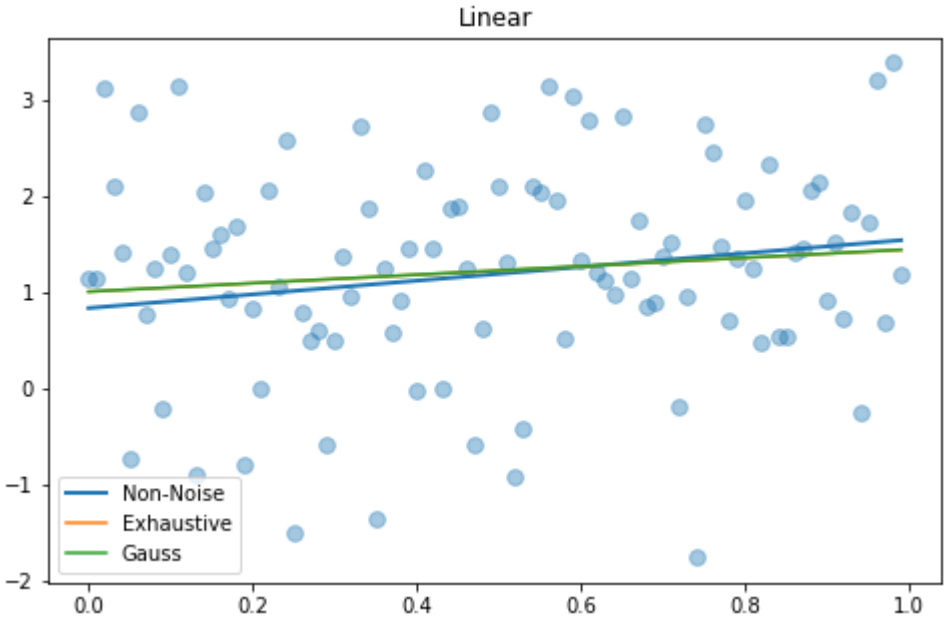
	Iterations	Value	x minimun	y
Exhaustive Search	57	121.496914	0.8310769	-0.513800
Gauss Search	5	121.496983	0.8305283	-0.514796
Nelder-Mead	38	121.496915	0.8312510	-0.513634

As expected the exhaustive search has the highest number of iterations, the Gaussian search has one seventh of the iterations compared to Nelder-mead, having similar results.

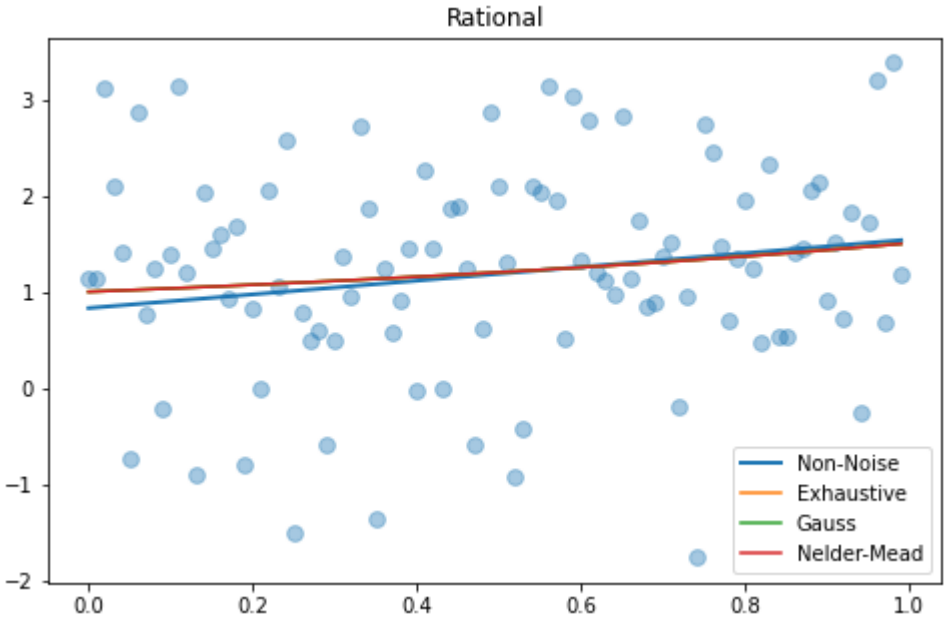
3. Plot

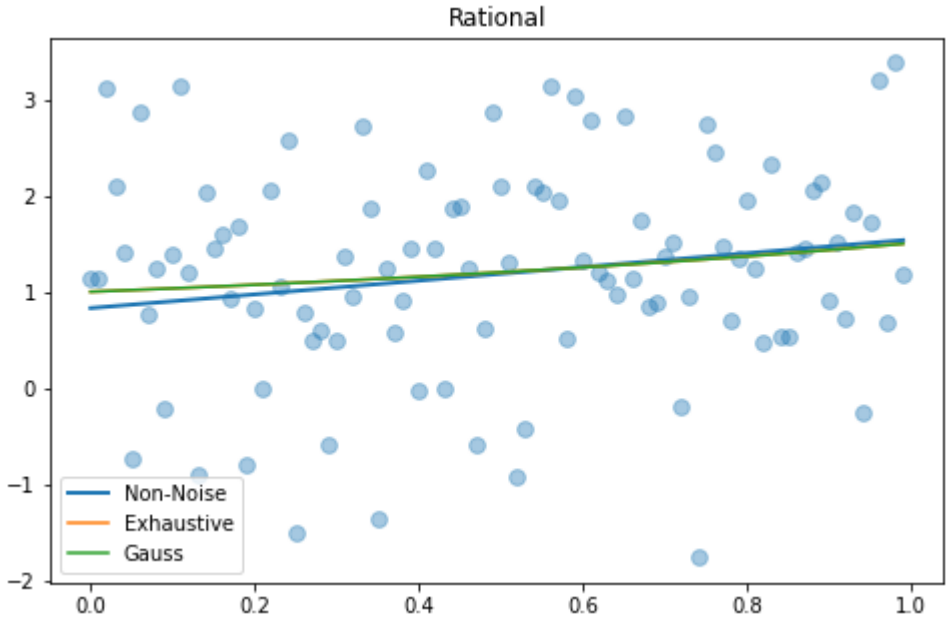
3.1 Linear





3.2 Rational


Rational
Non-Noise
Exhaustive
Gauss
Nelder-Mead


Rational
Non-Noise
Exhaustive
Gauss

Conclusions

- The exhaustive search is the most accurated but the most resource and time demanding
- The dichotomy has less time execution but it's not the most accurate
- The golden section is in the middle on time execution and is a little more accurated than dichotomy
- Nelder-Mead and Gauss has similar results, this due to the ε value used, giving good approximations

Bibliography

[1] Peroni S, Brute-force algorithms , https://comp-think.github.io/book/06.pdf
[2] Dichotomic search, https://en.wikipedia.org/wiki/Dichotomic_search
[3] Kochenderfer, Algorithms for optimization