FEDERAL STATE AUTONOMOUS EDUCATIONAL
INSTITUTION
OF HIGHER EDUCATION
ITMO UNIVERSITY

Report
on the practical task No. 4
Algorithms for unconstrained nonlinear optimization. Stochastic and metaheuristic algorithms.

Performed by

Anastasiia Pokasova

J4133c+

Accepted by

Dr Petr Chunaev

St. Petersburg
2020

Goal

The use of stochastic and metaheuristic algorithms (Simulated Annealing, Differential Evolution, Particle Swarm Optimization) in the tasks of unconstrained nonlinear optimization and the experimental comparison of them with Nelder-Mead and Levenberg-Marquardt algorithms

Problems and methods

Generate the noisy data $\{x_k, y_k\}$, where $k = 0, ..., 1000$, according to the following rule:

$$D_{it} = \begin{cases} -100 + \delta_k, & f(x_k) < -100 \\ f(x_k) + \delta_k, & -100 \leq f(x_k) \leq 100 \\ 100 + \delta_k, & f(x_k) > 100 \end{cases} \quad (1)$$

where $\delta_k \sim N(0, 1)$ are values of a random variable with standard normal distribution,

$$f(x) = \frac{1}{x^2 + -3x + 2}$$

Approximate the data by the rational function:

$$F(x, a, b, c, d) = \frac{ax + b}{x^2 + cx + d}$$

by means of least squares through the numerical minimization (with precision $\varepsilon = 0.001$ of the following function:

$$D(a, b) = \sum_{k=0}^{1000} (F(x_k, a, b, c, d) - y_k)^2. \quad (2)$$

To solve the minimization problem, use Nelder-Mead algorithm, Levenberg-Marquardt algorithm and at least two of the methods among Simulated Annealing, Differential Evolution and Particle Swarm Optimization. If necessary, set the initial approximations and other parameters of the methods. Use $\epsilon = 0.001$ as the precision; at most 1000 iterations are allowed. Visualize the data and the approximants obtained in a single plot. Analyze and compare the results obtained (in terms of number of iterations, precision, number of function evaluations, etc.)

Brief theoretical part

Partical Swarm Optimization Particle swarm optimization (PSO) is a computational method that optimizes a problem by iteratively trying to improve a candidate solution with regard to a given measure of quality. It solves a problem by having a population of candidate solutions named particles, and moving these particles around in the search-space according to simple mathematical formulae over the particle's position and velocity. In PSO, the new location of each particle is determined by a velocity term, which reflects the attraction of global best (g_b) and its own best (o_b) during the history of the particle and random coefficients.

Differential Evolution A basic variant of the Differential Evolution algorithm works by having a population of candidate solutions (called agents). These agents are moved around in the search-space by using simple mathematical formulae to combine the positions of existing agents from the population. If the new position of an agent is an improvement then it is accepted and forms part of the population, otherwise the new position is simply discarded. The process is repeated and by doing so it is hoped, but not guaranteed, that a satisfactory solution will eventually be discovered.

The choice of DE parameters e.g. differential weight, crossover region, population size, can have a large impact on optimization performance.

Simulated Annealing The simulated annealing algorithm is an optimization method which mimics the slow cooling of metals, which is characterized by a progressive reduction in the atomic movements that reduce the density of lattice defects until a lowest-energy state is reached. In a similar way, at each virtual annealing temperature, the simulated annealing algorithm generates a new potential solution (or neighbour of the current state) to the problem considered by altering the current state, according to a predefined criterion. The acceptance of the new state is then based on the satisfaction of the Metropolis criterion, and this procedure is iterated until convergence.

Nelder-Mead The method uses the concept of a simplex, which is a special polytope of n + 1 vertices in n dimensions. Examples of simplices include a line segment on a line, a triangle on a plane, a tetrahedron in three-dimensional space and so forth.

Nelder–Mead in n dimensions maintains a set of n + 1 test points arranged as a simplex. It then extrapolates the behavior of the objective function measured at each test point in order to find a new test point and to replace one of the old test points with the new one, and so the technique progresses. The simplest approach is to replace the worst point with a point reflected through the centroid of the remaining n points. If this point is better than the best current point, then we can try stretching exponentially out along this line. On the other hand, if this new point isn't much better than the previous value, then we are stepping across a valley, so we shrink the simplex towards a better point.

Levenberg-Marquardt The Levenberg-Marquardt algorithm combines two numerical minimization algorithms: the gradient descent method and the Gauss-Newton method. In the gradient descent method, the sum of the squared errors is reduced by updating the parameters in the steepest-descent direction. In the Gauss-Newton method, the sum of the squared errors is reduced by assuming the least squares function is locally quadratic in the parameters, and finding the minimum of this quadratic. The Levenberg-Marquardt method acts more like a gradient-descent method when the parameters are far from their optimal value, and acts more like the Gauss-Newton method when the parameters are close to their optimal value.

Results

Number of iterations and function evaluations for each method:

Nelder-Mead	Levenberg-Marquardt	Simulated Annealing	Differential Evolution
432, 743	null, 2	1000, 9071, 2	123, 7495

As we can see, Nelder-Mead and Levenberg-Marquardt methods require significantly less function computations. Simulated annealing and Differential Evolution methods give almost similar to Nelder-Mead results.

Results of approximation using rational function are shown below.

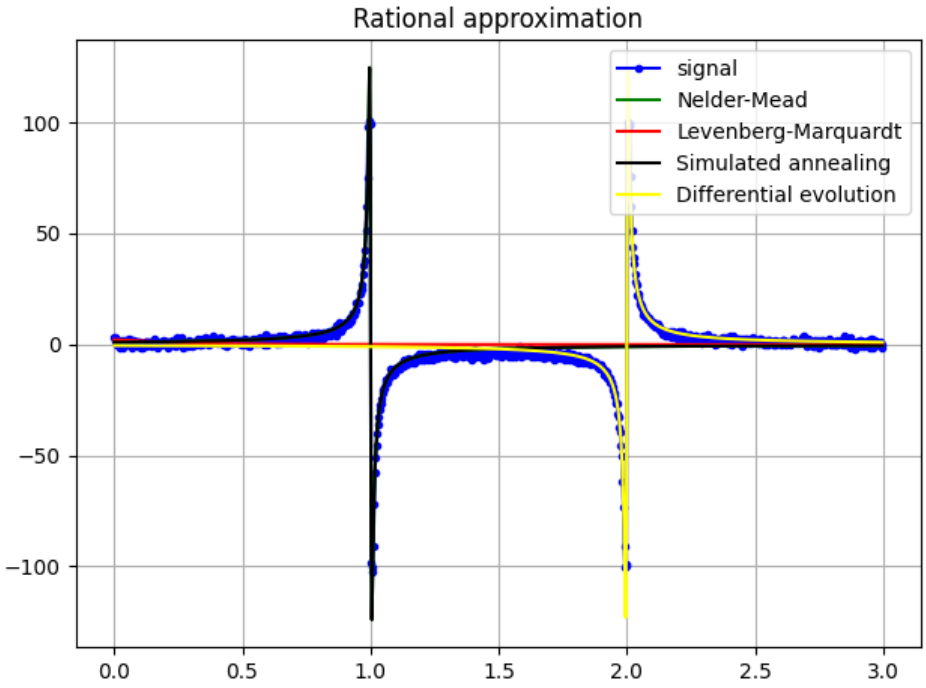

Figure 1: Results of approximation

Conclusion

As a part of a laboratory work, stochastic and metaeuristic methods were compared to direct methods of solving the unconstrained optimization problem. A program was implemented in order to analyze given methods.

Source code

Source code for the implementation is located here: here: `https://github.com/NeoIsALie/Algorithm_Analysis/tree/lab4`.