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
Alexandra Matveeva
J4134c
Accepted by
Dr Petr Chunaev

St. Petersburg
2021

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

I. Generate the noisy data (x_k, y_k), where $k = 0, \ldots, 1000$, according to the rule:

$$y_k = \begin{cases} -100 + \delta_k, f(x_k) < -100, \\ f(x_k) + \delta_k, -100 \le f(x_k) \le 100, \\ 100 + \delta_k. \end{cases} x_k = \frac{3k}{1000},$$

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

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

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

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 $\varepsilon = 0.001$ 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.).

II. Choose at least 15 cities in the world having land transport connections between them. Calculate the distance matrix for them and then apply the Simulated Annealing method to solve the corresponding Travelling Salesman Problem. Visualize the results at the first and the last iteration. If necessary, use the city dataset from https://people.sc.fsu.edu/~jburkardt/datasets/cities/cities.html

Brief theoretical part

Metaheuristic algorithms are a class of stochastic algorithms using a combination of randomization and local search. They are often based on learning from nature or biological systems.

Simulated annealing is a metaheuristic algorithm that solves the optimization problem similar to the process of annealing in metallurgy. Simulated Annealing is a stochastic global search optimization algorithm. This means that it makes use of randomness as part of the search process. This makes the algorithm appropriate for nonlinear objective functions where other local search algorithms do not operate well.

$f: \mathbb{R}^n \to \mathbb{R}$ is an energy. $T = \{T_k\}$ is a decreasing non-negative sequence. T is called cooling schedule of temperature.

Algorithm: Let a_0 be an initial approximation. At each iteration $k \in \mathbb{N}_0$:

- choose $a^* \in Neighbours(a_k)$, where Neighbours is a certain rule;
- if $f(a^*) \le f(a_k)$, then $a_{k+1} = a^*$; if $f(a^*) > f(a_k)$, then $a_{k+1} = a^*$ with probability

$$eps\left(-\frac{f(a^*)-f(a_k)}{T_k}\right),$$

- stop if $T_k \equiv 0$.

Differential evolution (DE) is a population-based metaheuristic search algorithm that optimizes a problem by iteratively improving a candidate solution based on an evolutionary process.

Choose $p \in [0,1]$, the *crossover probability*, $w \in [0;\ 2]$, the *differential weight*, and N $N \geq 4$, the *population size*. Let $x \in \mathbb{R}^n$ denote an agent in the population.

Algorithm: Until a termination criterion is met (e.g. the number of iterations performed):

- Randomly pick *N* agents **x** (i.e. the population).

- Pick three distinct agents **a**, **b** and **c** from the population, different from **x**.

- Compute the trial vector $y = (y_1, \dots, y_n)$ as follows. For $i = 1, \dots, n$, pick $r_i \in U(0,1)$. If $r_i < p$, then $y_i = a_i + w(b_i - c_i)$, otherwise $y_i = x_i$.

- If $f(\boldsymbol{y}) \leq f(\boldsymbol{x})$, then replace **x** with the trial vector **y**, otherwise keep **x**. Pick the best agent from the population and return it as the best found solution.

Particle swarm optimization (PSO) is a population-based stochastic optimization algorithm motivated by intelligent collective behavior of some animals such as flocks of birds or schools of fish.

A basic variant of the PSO algorithm works by having a population (called a swarm) of candidate solutions (called particles). These particles are moved around in the search-space according to a few simple formulae. The movements of the particles are guided by their own best-known position in the search-space as well as the entire swarm's best-known position. When improved positions are being discovered these will then come to guide the movements of the swarm. The process is repeated and by doing so it is hoped, but not guaranteed, that a satisfactory solution will eventually be discovered.

Formally, let $f: \mathbb{R}^n \to \mathbb{R}$ be the cost function which must be minimized. The function takes a candidate solution as an argument in the form of a vector of real numbers and produces a real number as output which indicates the objective function value of the given candidate solution. The gradient of f is not known. The goal is to find a solution a for which $f(a) \leq f(b)$ for all b in the search-space, which would mean a is the global minimum.

Travelling Salesman Problem (TSP): Given a set of cities and distance between every pair of cities, the problem is to find the shortest possible route that visits every city exactly once and returns to the starting point.

Results

I. To solve the minimization problem, Nedler-Mead algorithm, Levenberg-Marquardt algorithm, Simulated Annealing and Differential Evolution were used.

```
Nelder-Mead method: [-0.77635305  0.77673564 -2.00095739  1.00096738] function evaluations: 493 iterations: 282
Levenberg-Marquardt method: [-1.0015689   1.00204634 -2.00088087  1.00089717] function evaluations: 183
Simulated Annealing: [-0.99865943  0.99915876 -2.00097221  1.00098838] function evaluations: 9416 iterations: 1000
Differential Evolution: [-0.99878812  0.99928753 -2.00097149  1.00098767] function evaluations: 1260 iterations: 4
```

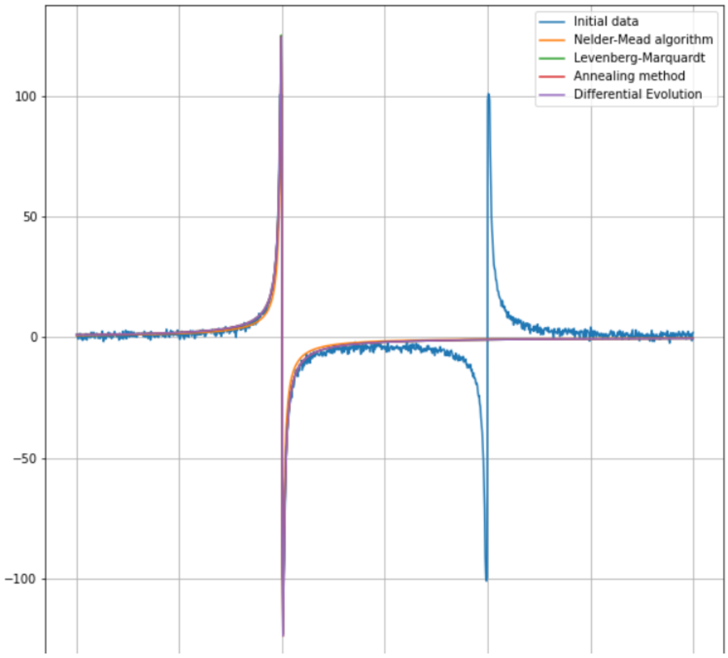

Fig. 1 - The results of the application of the Nelder-Mead algorithm, Levenberg-Marquardt algorithm, Simulated Annealing, Differential Evolution to solve the specified problem of rational approximation

In figure 1 we can see, that all methods detected first break and also showed approximately about the same accuracy. Differential evolution algorithm show faster performance for given task.

II. In the second part of the task we solved the Travelling Salesman Problem applying the Simulated Annealing method. For this assignment city dataset (https://people.sc.fsu.edu/~jburkardt/datasets/cities/kn57_xy.txt) was used.

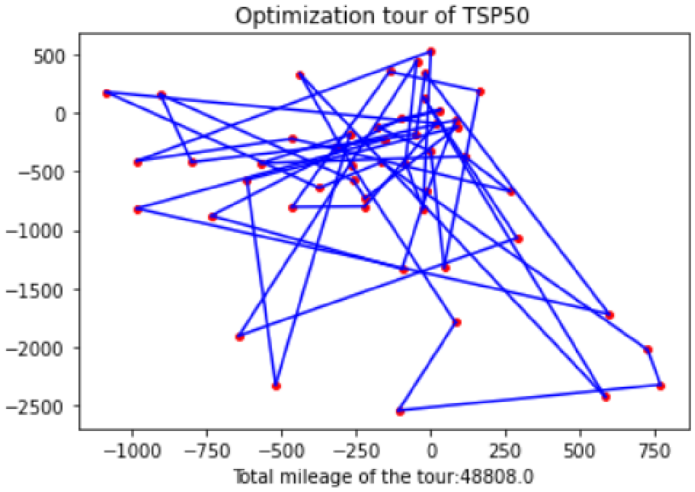

Fig. 2 – First iteration

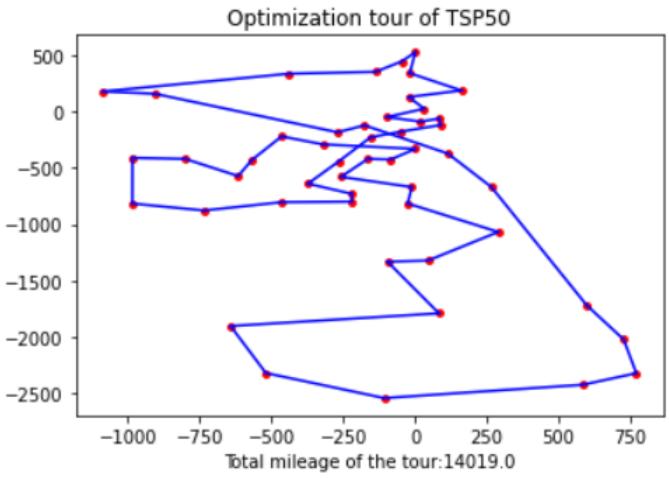

Fig. 3 – Last iteration

As you can see in figure 1 and figure 2, the algorithm find optimal solution: 14019.0 (instead of 48808.0).

Conclusion

During the execution of laboratory work, we solve the task of unconstrained nonlinear optimization using stochastic and metaheuristic algorithms (Simulated Annealing, Differential Evolution) and experimentally compare them with experimental comparison of them with. In the second part of work Travelling Salesman Problem was solved.

Appendix

GitHub Link: https://github.com/alex-mat-s/Algorithms/blob/main/Lab4.ipynb