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
Tiulkov Nikita
J4133c

Accepted by
Dr Petr Chunaev

St.Petersburg
2021

1. 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

2. Formulation of the problem

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

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

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

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

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

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 $\epsilon = 0.001$ 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.).

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.

3. Brief theoretical part

3.1. Nelder-Mead algorithm

The Nelder–Mead method is a commonly applied numerical method used to find the minimum or maximum of an objective function in a multidimensional space. 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.

The method approximates a local optimum of a problem with n variables when the objective function varies smoothly and is unimodal. Typical implementations minimize functions, and we maximize $f(x)$ by minimizing $-f(x)$. 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.

3.2. Differential Evolution method

Differential evolution is an metaheuristic algorithm that solves the optimization problem by maintaining a population of agents, i.e. candidate solutions, creating new agents by combining existing ones and further keeping the best one.

Choose $CR \in [0, 1]$, the crossover probability, $F \in [0, 2]$, the differential weight, and $NP \geq 4$, 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 NP agents x (i.e. the population).
- Pick three distinct agents a, b and c from the population, different from x.
- Compute the trial vector $y = (y_1, ..., y_n)$ as follows. For $i = 1, ..., n$, pick $r_i \in \mathbb{U}(0, 1)$. If $r_i < CR$, then $y_i = a_i + F(b_i - c_i)$, otherwise $y_i = x_i$.
- If $f(y) \leq f(x)$, 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.

3.3. Levenberg-Marquardt algorithm

The application of LMA is the least-squares curve fitting problem: given a set $(x_i, y_i)_{i=1}^m$, find the parameters β (column vector) of the model curve $f(x, \beta)$ so that the sum of the squares of the deviations $S(\beta)$ is minimized:

$$arg\min_{\beta} S(\beta) \equiv arg\min_{\beta} \sum_{i=1}^{m} [y_i - f(x_i, \beta)]^2$$

Start with an initial guess for β. In each iteration step, the parameter vector β is replaced by a new estimate $\beta + \Delta\beta$. To determine $\Delta\beta$, the function $f(x_i, \beta + \Delta\beta)$ is approximated by its linearization:

$$f(x_i, \beta + \Delta\beta) \approx f(x_i, \beta) + J_i \Delta\beta, J_i = (\nabla_{x_i} f(\beta))^T$$

The sum $S(\beta)$ has its minimum at a zero gradient with respect to β. The above first-order approximation of $f(x_i, \beta + \Delta\beta)$ gives

$$S(\beta + \Delta\beta) \approx \sum_{i=1}^{m} [y_i - f(x_i, \beta) - J_i \Delta\beta]^2$$

or in vector notation,

$$S(\beta + \Delta\beta) \approx [y - f(\beta)]^T [y - f(\beta)] - 2[y - f(\beta)]^T J \Delta\beta + \Delta\beta^T J^T J \Delta\beta,$$

where J is the Jacobian matrix, whose i-th row equals J_i , and where $f(\beta)$ and y are vectors with i-th component $f(x_i, \beta)$ and y_i , respectively.

Taking the derivative of $S(\beta + \Delta\beta)$ with respect to $\Delta\beta$ and setting to zero gives

$$(J^T J)\Delta\beta = J^T[y - f(\beta)],$$

that is in fact a system of linear equations with respect to $\Delta\beta$.

The system may be replaced by

$$(J^T J + \lambda I)\Delta\beta = J^T[y - f(\beta)],$$

where I is the identity matrix, giving the increment $\Delta\beta$ to the estimated parameter vector β.

3.4. Simulated Annealing

Simulated Annealing is a probabilistic technique for approximating the global optimum of a given function. Specifically, it is a metaheuristic to approximate global optimization in a large search space for an optimization problem. By modeling such a process, a point or a set of points is searched for at which the minimum of some numerical function is achieved $F(\overline{x})$, where $\overline{x} = (x_1, ..., x_m) \in X$. The solution is sought by sequential calculation of points $\overline{x_0}, \overline{x_1}$, ... from X. Each point starting from $\overline{x_1}$ ”pretends” to bring the solution closer to the previous ones. The algorithm takes the point $\overline{x_0}$ as the source data. At each step, the algorithm (which is described below) calculates a new point and lowers the value of the value (initially positive), understood as ”temperature”. The algorithm stops when it reaches a point that turns out to be at a temperature of zero.

3.5. Traveling salesman problem

The traveling salesman problem (also called the travelling salesperson problem or TSP) asks the following question: «Given a list of cities and the distances between each pair of cities, what is the shortest possible route that visits each city exactly once and returns to the origin city?» It is an NP-hard problem in combinatorial optimization.

4. Results

1. At Figure 1 we could see the calculation results for each of methods. The least number of iterations and small number of function evaluations has differential evolution method. Compare to it, the Levenberg-Marquardt method has the least function evaluations and also small number of iterations. Simulated Annealing showed the worst results in all cases.

	Method	num_iterations	func_evaluations
0	Nelder-Mead	447	800
1	Differential Evolution	2	195
2	Levenberg-Marquardt	20	20
3	Simulated Annealing	1000	8366

Figure 1: Result table

Visualization for each methods is presented at Figure 2.

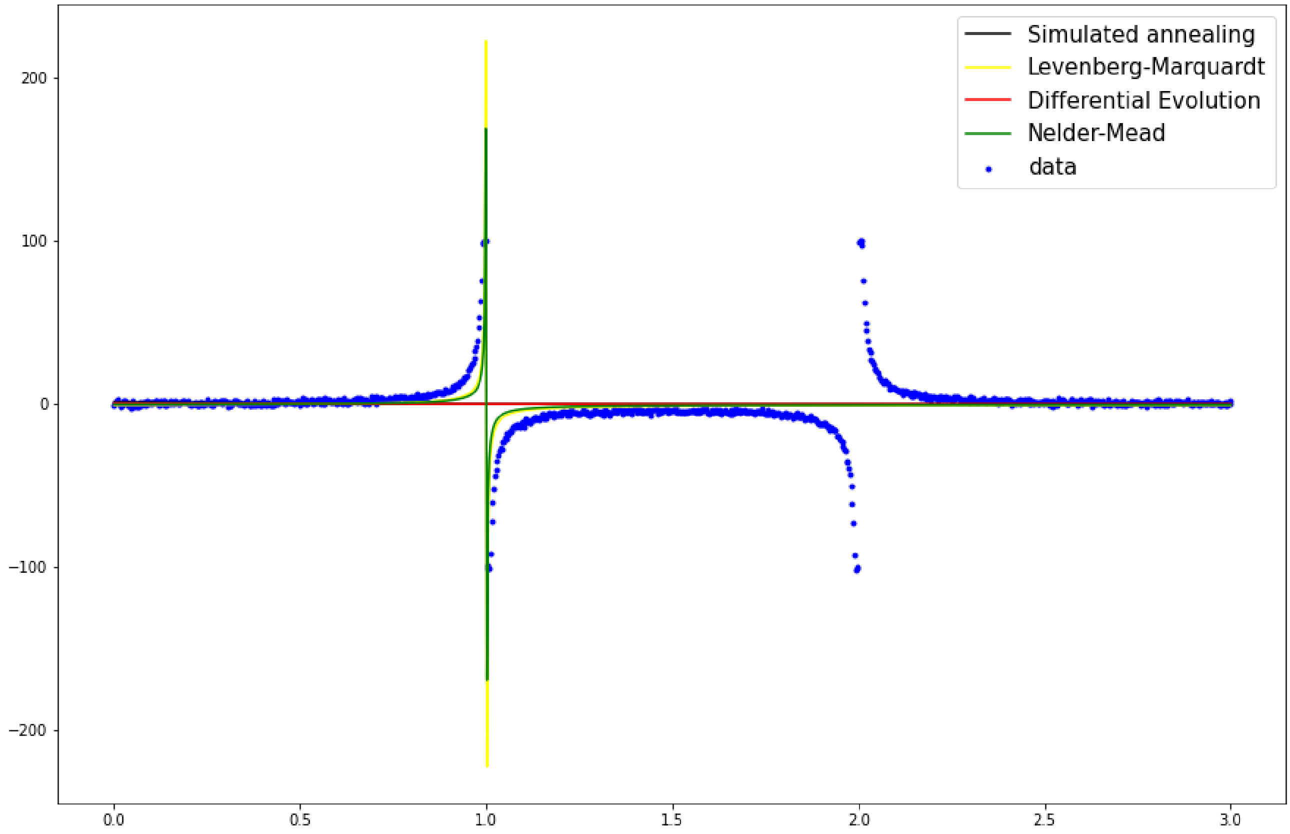
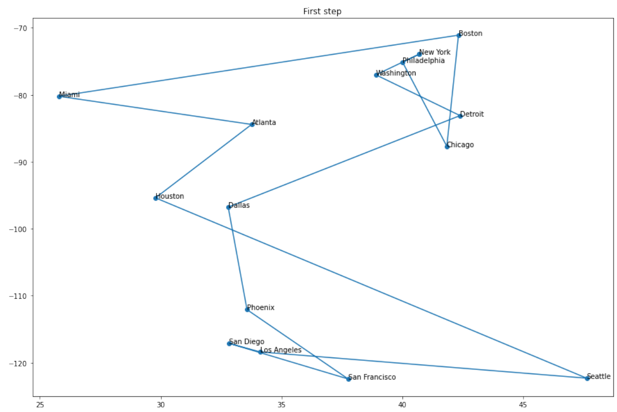
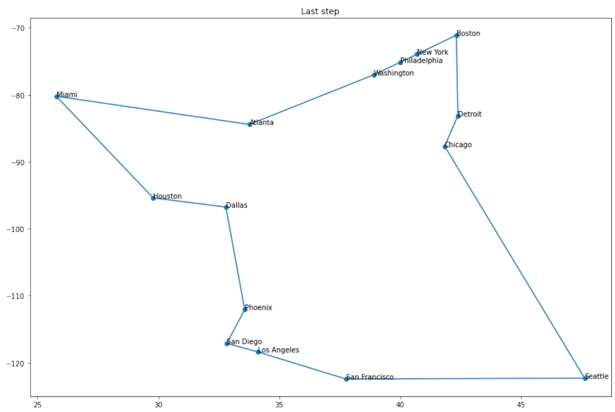


Figure 2: Methods visualization

II. For this task 15 USA cities was chose. For cities, Traveling salesman problem was set, the distance matrix was calculated and the Simulated Annealing method was applied. In the first iteration, the total length of the path was 11339 miles. On the last 7638 mile. Thus, the problem of finding the optimal path has been solved. The steps visualization is presented on figures below.





5. Conclusions

As a result of this work, stochastic and metaheuristic algorithms (Simulated Annealing, Differential Evolution) were investigated in the tasks of unconstrained nonlinear optimization and the experimental comparison of Nelder-Mead and Levenberg-Marquardt algorithms. Also, the Simulated Annealing algorithm was applied to solve the Traveling salesman problem posed on a dataset of 15 cities.

Appendix

The code of algorithms you can find on GitHub: https://github.com/FranticLOL/ITMO_Algorithms