Report - Optimization Theory and Methods – Warsaw University of Technology

Bartłomiej Guś, PW, MM, WD, MN

Simplex method (Nelder-Mead)

1. Task

For n-variable functions, the code of the optimization algorithm using the simplex (Nelder-Mead) method should be developed in MATLAB.
The algorithm should be tested for selected functions of one, two and three variables, with constraints.
[Please select 2-3 functions].

2. Introduction

Simplex method, standard technique in linear programming for solving an optimization problem, typically one involving a function and several constraints expressed as inequalities. The inequalities define a polygonal region, and the solution is typically at one of the vertices. The simplex method is a systematic procedure for testing the vertices as possible solutions.

3. Selected functions

In order to test implemented algorithm the following objective functions has been selected:

- Function A:
$$f(x) = 2\cos(0{,}4x) - 2\sin(1.5x) + 40;$$
Multimodal; several local minima in connection with the sin() and cos() components

- Function B (Himmelblau's function):
$$f(x,y) = (x^2 + y - 11)^2 + (x + y^2 - 7)^2$$
Multimodal; function has four identical local minima:
    - $f(3.0, 2.0) = 0.0$
    - $f(-2.805, 3.1313) = 0.0$
    - $f(-3.7793, -3.2832) = 0.0$
    - $f(3.5844, -1.8481) = 0.0$

- Function C:
$$f(x,y,z) = x^2 + y^2 + z^2$$
Unimodal; minimum at $f(0,0,0) = 0$

4. Implementation

Implemented algorithm with following examples in the attached .zip file. The entire implementation of the solution consists of the following .m files:

- *Simplex_ProjectB_Group1.m* - The file allows you to choose one of the 3 functions that were used to test the algorithm. It allows you to define constraints, as well as enter the parameters of the simplex algorithm: $\alpha, \beta, \gamma, \varepsilon, \sigma$. The script also includes functions that plot function graphs and algorithm steps. The following functions are also called in this file.
- *Function_draw_ProjectB_Group1* - The function is used to draw the starting point of the algorithm.
- *Function_to_optimize_ProjectB_Group1* - The function returns the patterns of functions selected to test the algorithm depending on the user's choice (var which_function_you_want_to_optimize).
- *Function_check_constraints* – The function that checks whether the coordinates of an input point are within the assumed constraints of the domain.
- *Function_get_distance_between_points* – The function that checks the distance between two given points.
- *Function_get_min_max_and_pG* - The function that implements the first step of the simplex algorithm - returns from the set of points those for which the function takes the minimum and maximum value and the center of gravity of the set of points.

- *Function_simplex* - Base implementation of the simplex method algorithm.

The algorithm was implemented based on the formulas and block diagram presented in the lecture, shown below.

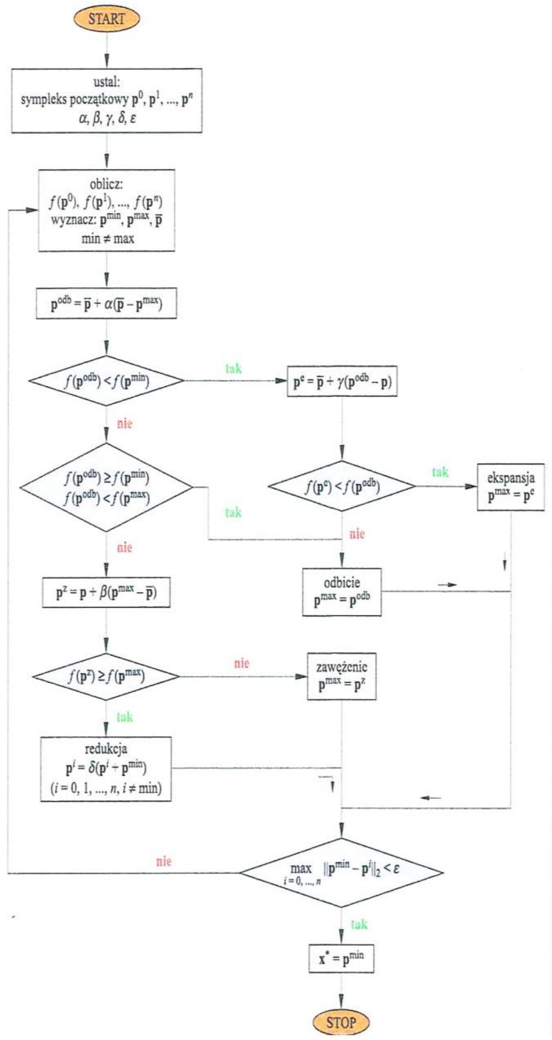

Figure 1 Block diagram of the Simplex algorithm

5. **Implementation test**

   a. **Function A**

   The following domain constraints are assumed for function A:

   - Case 1: $x \in < -2; 8 >$

     Tested on the following algorithm parameters:
     - alpha = 0.1;
     - beta = 0.5;
     - gamma = 2;
     - sigma = 0.5;
     - epsilon = 0.1;
   - Case 2: $x \in < 2; 5 >$

     Tested on the following algorithm parameters:
     - alpha = 0.5;
     - beta = 0.1;
     - gamma = 1;
     - sigma = 0.1;
     - epsilon = 0.1;

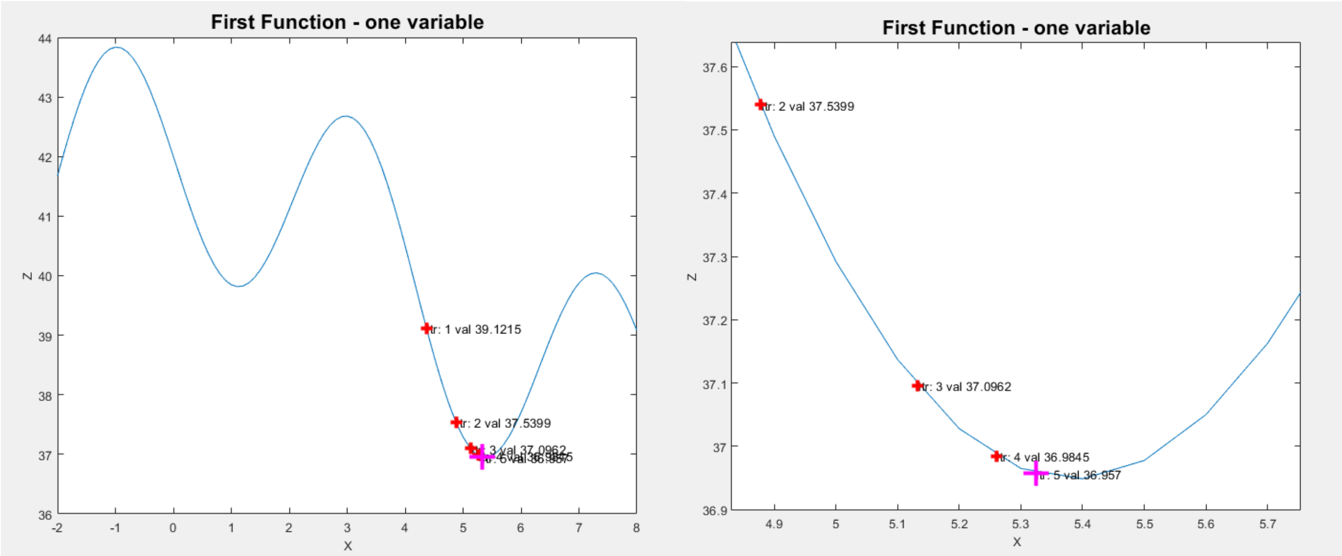

Figure 2 Simplex optimization results for function A (one variable); case 1

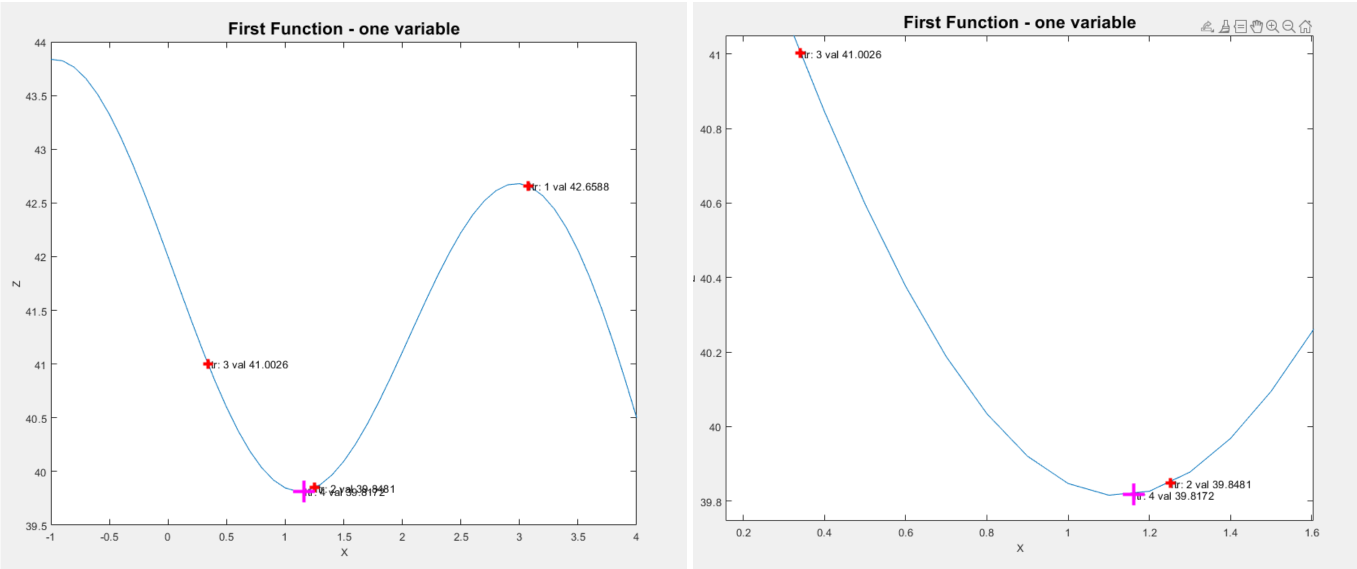

Figure 3 Simplex optimization results for function A (one variable); case 2

Based on the graph of the function, it can be seen that with the simplex method the minimum was determined with assumed accuracy (epsilon parameter).

b. Function B

The following domain constraints are assumed for function B:

- Case 1: $x \in < -4; 4 >; y \in < -4; 4 >$
- Case 2: $x \in < -4; -2 >; y \in < -4; -1 >$

Tested on the following algorithm parameters:

- alpha = 0.1;
- beta = 0.5;
- gamma = 1;
- sigma = 0.5;
- epsilon = 0.05;

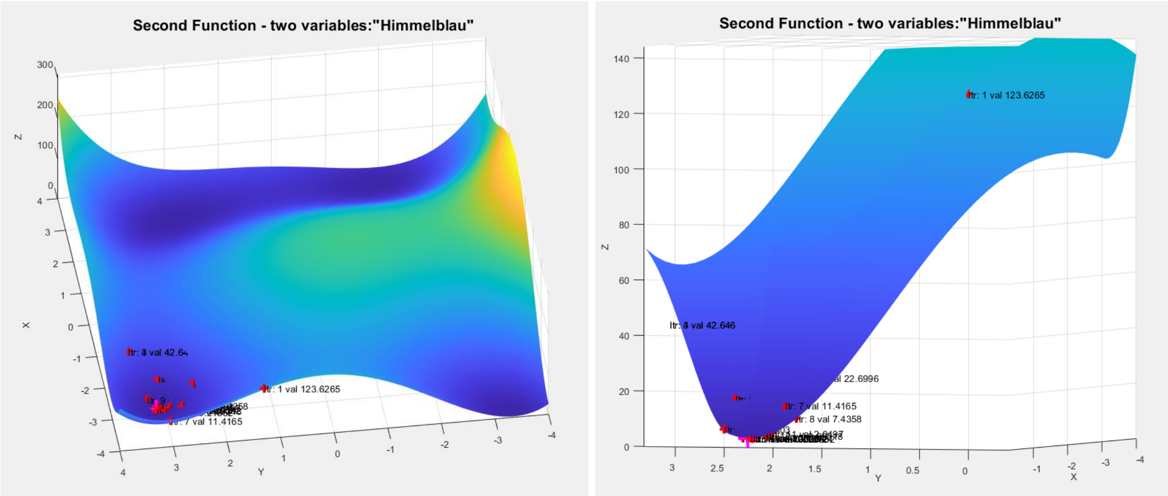

Figure 4 Simplex optimization results for function B (2 variables); case 1

Results:

- min x: -3.7709
- min y: -3.2636
- min f(x, y): 0.016306

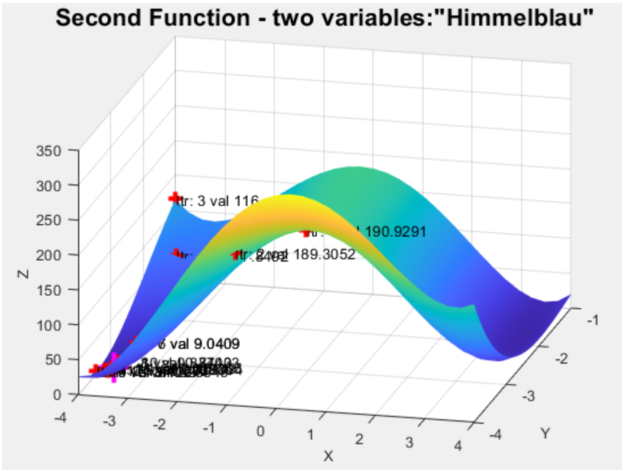

Figure 5 Simplex optimization results for function B (2 variables); case 2

Results:

- min x: -3.7564
- min y: -3.2764
- min f(x, y) = 0.028094

Based on the mathematically determined minima of the function, it can be seen that using the simplex method, one of the minima of the function was determined with assumed accuracy (the epsilon parameter). Which of the 4 minima is found depends on the drawn initial simplex.

c. Function C

The following domain constraints are assumed for function C:

- Case 1: $x \in < -4; 4 >; y \in < -4; 4 >; z \in < -4; 4 >$
- Case 2: $x \in < -10; 4 >; y \in < -2; 4 >; z \in < -1; 10 >$

Tested on the following algorithm parameters:

- alpha = 0.1;
- beta = 0.5;
- gamma = 1;
- sigma = 0.5;
- epsilon = 0.05;

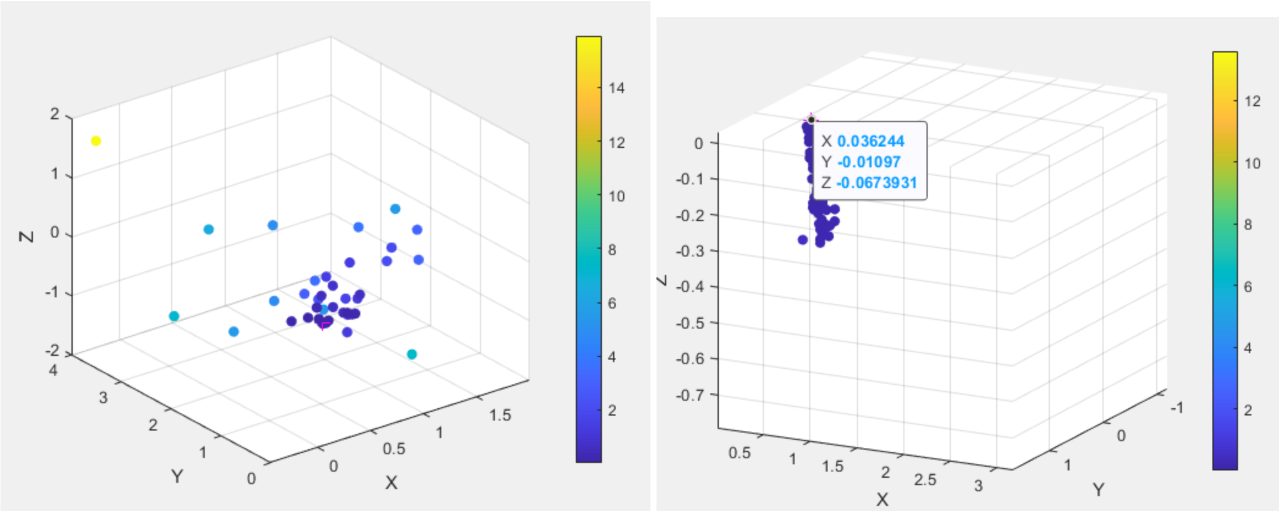

Figure 6 Simplex optimization results for function C (3 variables); case 1

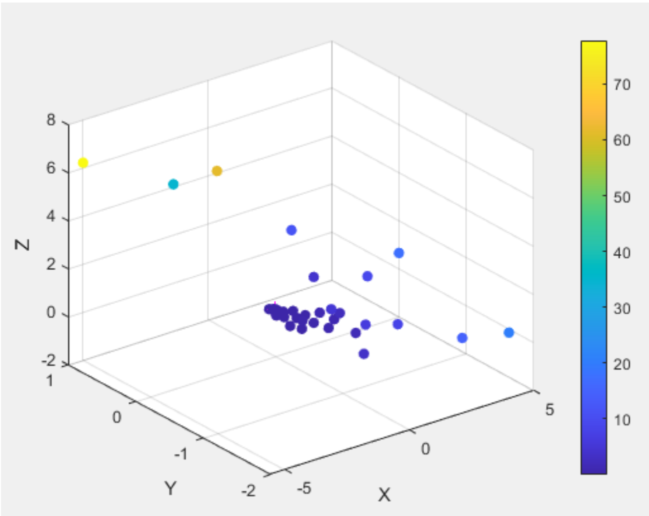

Figure 7 Simplex optimization results for function C (3 variables); case 2

Due to the fact that the C-function is a 4-dimensional function, there has been plotted only following steps of algorithm. For the simplex method parameters adopted above, the function returns the following x, y, z coordinates of the found minimum:

Case 1 results:

- Min x: 0.036244
- Min y: -0.01097
- Min z: -0.067393
- Min f(x, y, z) = 0.0059758

Case 2 results:

- Min x: -0.010627
- Min y: -0.0032329
- Min z: -0.0024956
- Min f(x, y, z) = 0.00012961

Which is within the assumed accuracy.

6. Conclusion

Based on the optimization results of the 3 sample functions, it can be concluded that the algorithm was implemented correctly. The solution allows for simplex optimization of multivariable functions. The fully parameterized script allows the user to test the algorithm for various key parameters of the simplex method, as well as various function constraints. Implemented plotting functions showing the successive steps of the algorithm allow the user to monitor the progress of the algorithm and interpret its performance.