# Finding the function minimum of standard benchmark functions using Trajectory Methods

Pasat Tudor Cosmin

October 26, 2021

## 1 Abstract

Presented in this report are two main methods, Hill Climbing, which also has two different approaches of searching through the neighbours to find the next candidate solution, and Simulated Annealing. We are going to explain how they work in finding the minimum of a function and in the end show the results of some experiments performed on four standard benchmark functions, tested on 3,10 and 30 dimensions. These experiments led to various results and in the end we discuss a more detailed comparison.

## 2 Introduction

In numerical analysis, Hill Climbing is a mathematical optimization technique which belongs to the family of local search. It is an iterative algorithm that starts with an arbitrary solution to a problem, then attempts to find a better solution by making an incremental change to the solution. If the change produces a better solution, another incremental change is made to the new solution, and so on until no further improvements can be found.
Simulated annealing (SA) 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. It is often used when the search space is discrete (for example the traveling salesman problem, the boolean satisfiability problem, protein structure prediction, and job-shop scheduling).
We tested the two trajectory methods Hill Climb and Simulated Annealing, which are described in more detailed in the "Methods" section.
In the "Functions" section you are going to find the definition and all the global minimums of the test functions, four standard benchmark functions, De Jong's function 1, Schwefel's function 7, Rastrigin's function 6 and Michalewicz's function 12.

Moving forward we presented in the section "Experiments" four tables, one for each function, containing the dimensions and all the results of the tests we performed, and in the end a comparison between between the trajectory methods.

# 3 Method

For both of these methods we used to represent the solution vectors of bits of length $L = D \cdot l$, where $l = \lceil log_2 10^p \cdot (b - a) \rceil$ is the length of one dimension, D is the number of dimensions, p is the precision and $[a, b]$ is the interval on which the function is defined.

## 3.1 Hill Climbing

The Hill Climbing is an iterative technique that does a local search starting from a random chosen candidate solution and compares it to his neighbours to find the best one. Every time a solution is found the algorithm restarts in order to maximise the exploration space.
The neighbours can be found in two different ways, the first improvement, that chooses the first neighbour that has a better value than the current solution and the best improvement that searches through all the neighbours and finds the best one.
To generate the neighbours we use a Hamming distance of 1 and negate a random bit until we find a better solution in the case of First Improvement Hill Climb or we go through all the neighbours to find the best one for Best Improvement Hill Climb.
In Hill Climb we first start by generating a random variable with a random vector of bits of length L in which we are going to remember the best solution and then we enter two loops. With the first one we generate different random solutions and starting from it we use the second loop to find the local optimum. This way we can make sure that global optimum is found.

## 3.2 Simulated Annealing

The Simulated Annealing is a metaheuristic method that uses almost the same principles as Hill climbing but it gives the possibility of getting out of a local minimum which expands the exploration space.
We used the initial value for the temperature of T = 1000 and is updating after every iteration of the firs loop by multiplying the current temperature. The first loop stops when the temperature is smaller than 0.00005. To search through the neighbours we use a first improvement method, but different from the Hill climbing one because we also have a probability to get out of a local minimum. To do this search we shuffled through all the neighbours. The inner loop has two stoping conditions, one stops it after all the neighbours were visited but still wasn't found another solution and the second one stops it after a number of iteration introduced by the user, that way the it is more time efficient.

# 4 Functions tested

To test the methods presented above we used four standard benchmark functions, defined as seen below.

## 4.1 De Jong's function 1

$$f(x) = \sum_{i=1}^{n} x_i^2, i = 1 : n, x_i \in [-5.12, 5.15]$$

Global minimum: f(x) = 0, x(i) = 0, i = 1:n

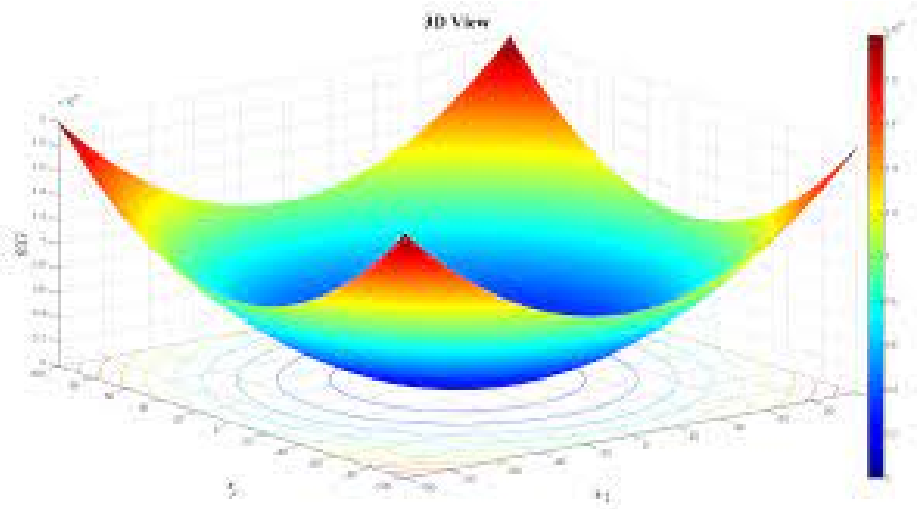

Figure 1: De Jong's function

### 4.2 Schwefel's function 7

$$f(x) = \sum_{i=1}^{n} -x_i \cdot sin\left(\sqrt{|x_i|}\right), x_i \in [-500, 500]$$

Global minimum:
f(x) = -2094.91, x(i) = 420.9687, i = 1:5
f(x) = -4189.82, x(i) = 420.9687, i = 1:10
f(x) = -12569.48, x(i) = 420.9687, i = 1:30

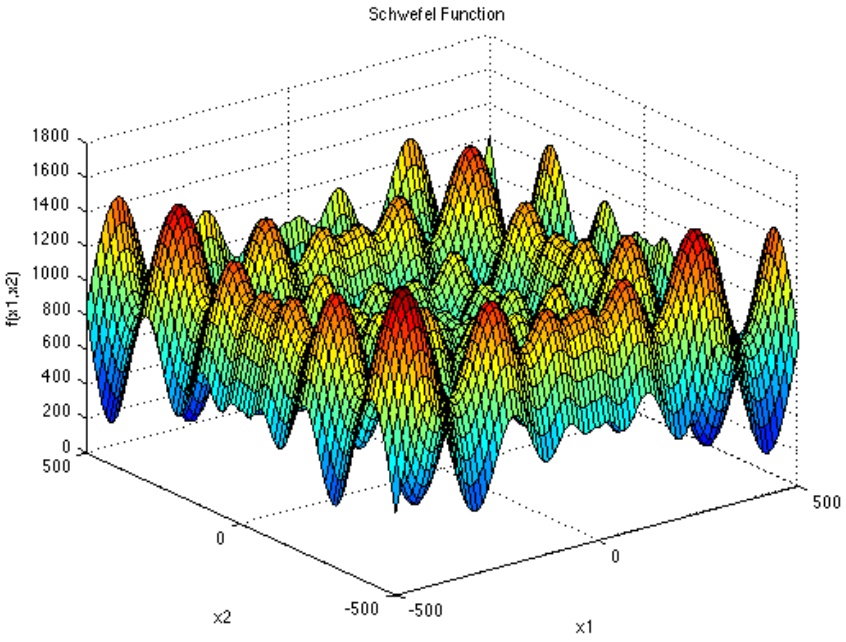


Figure 2: Schwefel's function

### 4.3 Rastrigin's function 6

$$f(x) = A \cdot n + \sum_{i=1}^{n} \left[ x_i^2 - A \cdot cos(2\pi x_i) \right], i = 1 : n, A = 10, x_i \in [-5.12, 5.15]$$

Global minimum: f(x) = 0, x(i) = 0, i = 1:n

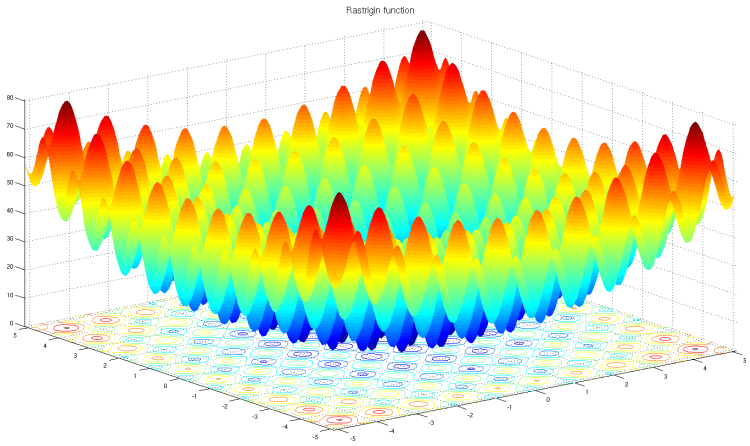


Figure 3: Rastrigin's Function.

### 4.4 Michalewicz's function 12

$$f(x) = -\sum_{i=1}^{n} sin(x_i) \cdot \left( sin\left(\frac{i \cdot x_i^2}{\pi}\right)\right)^{2 \cdot m}, i = 1 : n, m = 10, x_i \in [0, \pi]$$

Global minimum:
f(x) = -4.687, x(i) = ???, i = 1:5
f(x) = -9.66, x(i) = ???, i = 1:10
f(x) = -28.98, x(i) = ???, i = 1:30

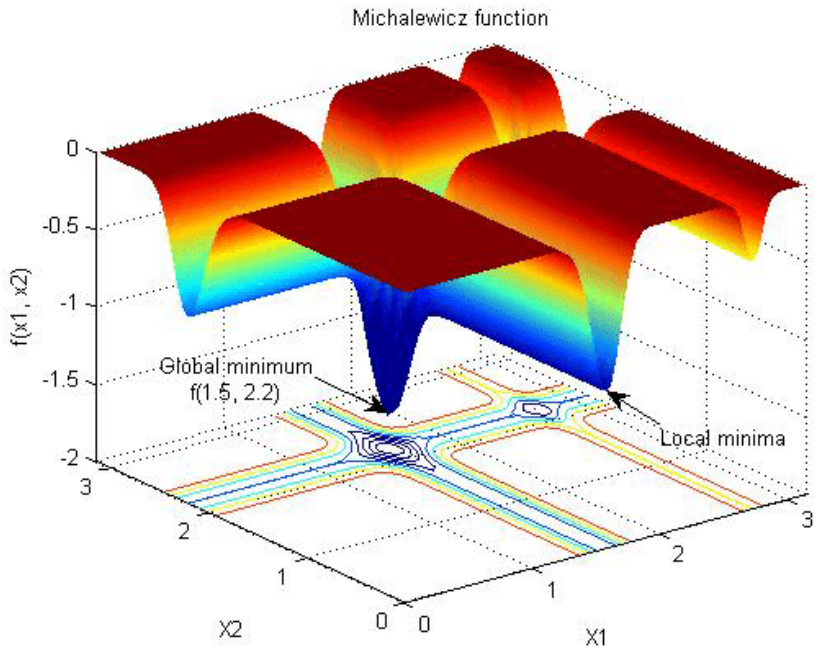


Figure 4: Michalewicz's function.

# 5 Experiment

The experiments were performed on an Intel i5 processor.
All three methods were tested 5,10 and 30 dimensions, a precision of $10^{-5}$ and 10000 iterations for First Improvment Hill climbing, 5000 for Best Improvment Hill climbing and the temperature T = 1000 for Simulated Annealing.

## 5.1 De Jong's function 1

| | FIHC | | | BIHC | | | SA | | |
|---|---|---|---|---|---|---|---|---|---|
| D | Solution | Average | Time | Solution | Average | Time | Solution | Average | Time |
| 5 | 0 | 0 | 53s | 0 | 0 | 54s | 0 | 0.00006 | 4s |
| 10 | 0 | 0 | 411s | 0 | 0 | 319s | 0 | 0.00005 | 6s |
| 30 | 0 | 0 | 7245s | 0 | 0 | 6651s | 0.00005 | 0.00026 | 13s |

## 5.2 Schwefel's function 7

| | FIHC | | | BIHC | | | SA | | |
|---|---|---|---|---|---|---|---|---|---|
| D | Solution | Average | Time | Solution | Average | Time | Solution | Average | Time |
| 5 | -2094.81 | -2094.29 | 182s | -2094.81 | -2094.36 | 146s | -2094.71 | -2094.38 | 4s |
| 10 | -4132.01 | -4103.54 | 1137s | -4158.39 | -4098.97 | 978s | -4155.28 | -4141.62 | 6s |
| 30 | -11958.5 | -11958.5 | 11898s | -12063 | -12063 | 20264s | -11526.8 | -11323.7 | 14s |

## 5.3 Rastrigin's function 6

| | FIHC | | | BIHC | | | SA | | |
|---|---|---|---|---|---|---|---|---|---|
| D | Solution | Average | Time | Solution | Average | Time | Solution | Average | Time |
| 5 | 0 | 0.0002 | 83s | 0 | 0.0412 | 65s | 1.995 | 2.6679 | 3s |
| 10 | 3.4794 | 3.9351 | 521s | 3.2348 | 4.09 | 417s | 8.20602 | 9.5568 | 5s |
| 30 | 34.3296 | 35.2302 | 1283s | 28.2675 | 29.2769 | 8327s | 32.2833 | 33.971 | 11s |

## 5.4 Michalewicz's function 12

| | FIHC | | | BIHC | | | SA | | |
|---|---|---|---|---|---|---|---|---|---|
| D | Solution | Average | Time | Solution | Average | Time | Solution | Average | Time |
| 5 | -3.6914 | -3.53 | 70s | -3.694 | -3.68 | 49s | -3.65706 | -3.59781 | 4s |
| 10 | -8.60521 | -8.47 | 463s | -9.07 | -8.94 | 350s | -8.1407 | -8.04701 | 6s |
| 30 | -24.31329 | -23.511 | 9081s | -26.0712 | -25.985 | 6829s | -26.215 | -26.0086 | 12s |

### 5.5 Comparison

Now we start comparing the results displayed above. Each table contains the statistics for the four benchmark functions , De Jong's function 1, Schwefel's function 7, Rastrigin's function 6 and Michalewicz's function 12. On the firs column we find the dimension tested, on the second the First Improvement Hill Climb, than Best Improvement Hill Climb and on the last the Simulated Annealing.
We start with the 5-dimensional version of the functions, as we can see all the methods gave a similar result, except the Simulated Annealing, which gave a worst solution for the Rastrigin's function, but also it has the smallest running time.
We continue with the 10-dimensional version, the results are also similar and this time the Simulated Annealing gave a solution two time higher the the other ones for the Rastrigin's function, but again has a better running time for all functions.
On the 30-dimensional version we see the biggest difference in time between the methods. Best Improvement Hill Climb gives a better solution, but also has the highest running time. If on Hill Climb the time can get up to 8 hours for finding the function minimum, in the case of Schwefel's function, Simulated Annealing finds a close solution in 14 seconds.
Concluded from our test is that the number of iterations has an important role in finding the optimum solution, higher it gets a more precise solution results. Due to the way the neighbourhood is searched through, we can see a big difference in time between First Improvement and Best Improvement Hill Climbing. While we tested the FIHC with 10000 iterations, on BIHC we only used 5000 and the running time for them is almost the same, but BIHC has more precise solutions.
Overall even though the Best Improvement Hill Climb gives the most accurate results, Iterated Hill Climb gives a close solution but with a almost 50% better time. Simulated Annealing doesn't always give an optimum solution but the one it gives it's close from the others and is a lot more time efficient.

## 6 Conclusions

This is all we have to say about the two trajectory methods, Hill Climbing and Simulated Annealing, tested on the standard benchmark functions , De Jong's function 1, Schwefel's function 7, Rastrigin's function 6 and Michalewicz's function 12. As we could see, the results were various, Hill climbing gave more accurate solutions, the Best Improvement version to be exact , but Simulated Annealing had way better running times.

# References


[1] Course Page
Genetic Algorithms: https://profs.info.uaic.ro/~eugennc/teaching/ga

[2] Wikipedia Commons
Hill Climbing: https://en.wikipedia.org/wiki/Hill_climbing
Simulated annealing: https://en.wikipedia.org/wiki/Simulated_annealing

[3] Functions
Function definitions:
De Jong's function 1: http://www.geatbx.com/docu/fcnindex-01.html#P89_3085
Schwefel's function 7: http://www.geatbx.com/docu/fcnindex-01.html#P150_6749
Rastrigin's function 6: http://www.geatbx.com/docu/fcnindex-01.html#P140_6155
Michalewicz's function 12: http://www.geatbx.com/docu/fcnindex-01.html#P204_10395

[4] Others
Geeksforgeeks:https://www.geeksforgeeks.org/simulated-annealing/
Teachingsite:https://www.u-aizu.ac.jp/~qf-zhao/TEACHING/MH/Lec04.pdf
Simulated Annealing article:
https://machinelearningmastery.com/simulated-annealing-from-scratch-in-python/