# Benchmark
# Hill Climbing and Simulated Annealing
# exploring the minimum

Adrian-Valentin Panaintescu

October 29, 2021

**Abstract**

Heuristic search refers to a search strategy that attempts to optimize the solution by iteratively improving the solution based on a given heuristic function or a cost measure. A heuristic search method does not always guarantee to find the optimal solution, but may instead find an acceptable solution within a reasonable amount of time and memory space. The most common methods includes hill climbing methods, the best-first search, the A* algorithm, simulated-annealing, and genetic algorithms.

## 1 Introduction

In this article we will describe how these algorithms: Hill Climbing best or first improvement and Simulated Annealing handle the problem of finding the minimum. The tests are perform using 4 specific mathematical functions:

- De Jong 1 function
- Schwefel's function
- Rastrigin's function
- Michalewicz's function

Testing for each function all three methods and varying with dimensions: 5, 10, 30. Also in the experiment we took into account the accuracy of the numbers and the size of its representation.

## 2 Motivation

In this experiment we will analyze how one method is better than another and in what circumstances. And to what extent it influences various parameters such as: *dimension*, *method* or *precizion* result and computation time.

# 3 Methods

As a mention, the start point is generating into a binary representation by generating $c \cdot n$ *bits.* That $c$ reprezent minimul of bits used to reprezent o value and is computed in the begining by:

$$f(a, b, p) = \lfloor \frac{\ln(10^p \cdot (b - a))}{\ln(2)} \rceil \tag{1}$$

where:

$a$ = lower range limit of the domain
$b$ = upper range limit of the domain
$p$ = precision

A neighbor is recognized that the difference by binary representation is only a bit.

## 3.1 Hill Climbing First Improvement

In Hill Climbing technique, starting at the base of a hill, we walk upwards until we reach the top of the hill. In other words, we start with initial state and we keep improving the solution until its optimal. The improvement is made when you find an optimal neighbor.
As a mention, we start from a random point and we are looking through his neighbors and we move on to the first neighbor who is better.

## 3.2 Hill Climbing Best Improvement

Like Hill Climbing First Improvement, a gradient search is made only that it differs in the fact that all the neighbors are evaluated and the best is our jump.

## 3.3 Simulated Annealing

It is also a gradient search, only that the neighbors consider themselves a little closer step by step, so that approaching the optimal solution it will converge in the optimal.

# 4 Experimental

During the experiment each parameter configuration was run 30 times and taken in the minimum, minimum horse but especially the average together with the standard deviation.
The precision was 5 to maintain uniformity and facilitate comparison.
Hill Climbing has an 10000 internal iterations and Simulated Annealing run with start with temperature $T$ = *200* and accepted temperature $T$ ¿ *0.00001* and the function for actualization:

$$f(T) = T \cdot s, s = 0.95$$

## 4.1 Functions

Using the following functions, we can argue that the search space is one that does not allow a greedy approach. These functions are characterized by the fact that they have several local optimal and plateau.

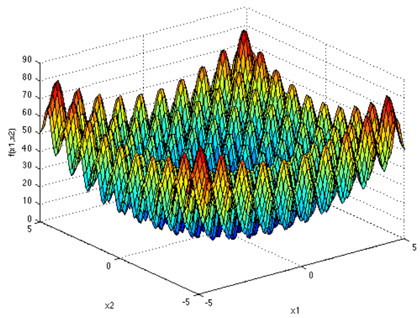


Figure 1: Rastrigin's Function

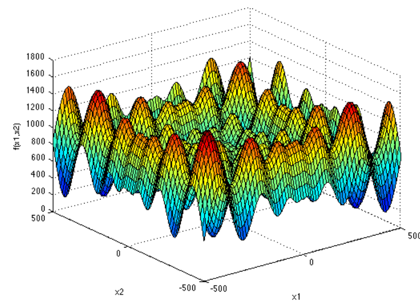


Figure 2: Schwefel's Function

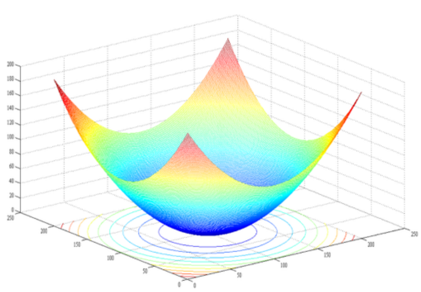

Figure 3: De Jong 1 Function

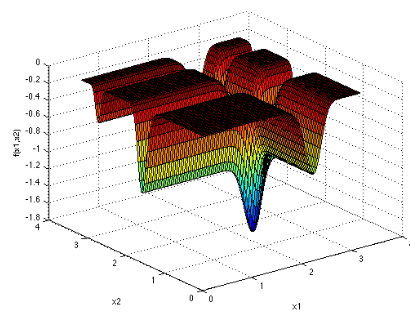


Figure 4: Michalewicz's Function

### 4.1.1 De Jong 1

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

### 4.1.2 Schwefel's

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

#### 4.1.3 Rastrigin's

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

#### 4.1.4 Michalewicz's

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

# 5 Results

## 5.1 De Jong 1

| Algorithm | Dim | Mean | Min | Max | Deviation | Time (s) |
|---|---|---|---|---|---|---|
| hc-first | 10 | 0 | 9.76682e-07 | 9.76682e-07 | 8.47033e-22 | 2.08212 |
| hc-best | 10 | 0 | 2.51984e-05 | 0.000133805 | 2.30341e-05 | 0.812047 |
| sa | 10 | 0 | 9.76682e-07 | 9.76682e-07 | 8.47033e-22 | 2.04219 |
| hc-first | 15 | 0 | 1.46502e-06 | 1.46502e-06 | 6.35275e-22 | 3.8621 |
| hc-best | 15 | 0.00014 | 4.91271e-05 | 0.0016712 | 0.000103252 | 0.705516 |
| sa | 15 | 0 | 1.46502e-06 | 1.46502e-06 | 6.35275e-22 | 3.95299 |
| hc-first | 30 | 0 | 2.93004e-06 | 2.93004e-06 | 1.27055e-21 | 0.14188 |
| hc-best | 30 | 0.00019 | 8.10646e-05 | 0.000568624 | 5.19716e-05 | 0.915175 |
| sa | 30 | 0 | 2.93004e-06 | 2.93004e-06 | 1.27055e-21 | 8.16611 |

Figure 5: De Jong 1 results table

## 5.2 Schwefel's

| Algorithm | Dim | Mean | Min | Max | Deviation | Time (s) |
|---|---|---|---|---|---|---|
| hc-first | 10 | -3073.32 | -3684.07 | -2624.63 | 193.153 | 2.63315 |
| hc-best | 10 | -3959.01 | -4189.83 | -3637.51 | 122.819 | 0.691972 |
| sa | 10 | -3319.24 | -3680.93 | -2730.96 | 221.817 | 2.6178 |
| hc-first | 15 | -4679.9 | -5291.33 | -3963.48 | 243.603 | 4.70815 |
| hc-best | 15 | -6027.99 | -6284.12 | -5654.59 | 118.773 | 0.709938 |
| sa | 15 | -4980.69 | -5510.63 | -4239.05 | 288.24 | 4.67842 |
| hc-first | 30 | -9342.27 | -9988.14 | -8395.22 | 294.608 | 14.0914 |
| hc-best | 30 | -11937.5 | -12347.7 | -11407.5 | 175.155 | 1.00358 |
| sa | 30 | -9888.68 | -10835.3 | -8710.03 | 419.934 | 14.3921 |

Figure 6: Schwefel's results table

## 5.3 Rastrigin's

| Algorithm | Dim | Mean | Min | Max | Deviation | Time (s) |
|---|---|---|---|---|---|---|
| hc-first | 10 | 23.8912 | 9.95389 | 40.9455 | 7.12647 | 1.6802 |
| hc-best | 10 | 13.2314 | 2.99668 | 26.3303 | 6.20286 | 0.584127 |
| sa | 10 | 20.7069 | 6.95289 | 38.183 | 5.5531 | 1.7013 |
| hc-first | 15 | 36.9367 | 18.8341 | 49.994 | 7.28582 | 3.00671 |
| hc-best | 15 | 15.7768 | 6.4644 | 27.418 | 4.98917 | 0.65536 |
| sa | 15 | 29.3345 | 15.6853 | 48.5941 | 7.20882 | 3.08207 |
| hc-first | 30 | 73.2062 | 40.4491 | 112.753 | 11.5703 | 8.68145 |
| hc-best | 30 | 35.0751 | 21.6035 | 54.4033 | 7.20532 | 0.899367 |
| sa | 30 | 56.7813 | 38.6183 | 75.9252 | 7.05237 | 8.7485 |

Figure 7: Rastrigin's results table

## 5.4 Michalewicz's

| Algorithm | Dim | Mean | Min | Max | Deviation | Time (s) |
|---|---|---|---|---|---|---|
| hc-first | 10 | -7.3794 | -8.82491 | -4.40354 | 0.744486 | 2.46225 |
| hc-best | 10 | -9.21779 | -9.88988 | –8.71383 | 0.284067 | 0.869014 |
| sa | 10 | -8.10464 | -9.21586 | -6.86494 | 0.598839 | 2.4336 |
| hc-first | 15 | -11.5333 | -13.6796 | -9.25147 | 0.920931 | 4.85392 |
| hc-best | 15 | -14.0174 | -14.648 | -13.0156 | 0.348111 | 1.08516 |
| hc-first | 30 | -22.1946 | -24.4272 | -18.7593 | 0.921606 | 16.4472 |
| hc-best | 30 | -27.8643 | -29.0349 | -25.9075 | 0.523017 | 1.72776 |
| sa | 30 | -23.6245 | -27.053 | -20.2618 | 1.36884 | 16.7719 |

Figure 8: Michalewicz's results table

## 5.5 Obeservation

It is clear that the computation time has increased with increasing size but what you could see is that this increase is not more pronounced for Simulated Annealing.
We can see that the number of dimensions plays a very important role in the purpose the better the number of variables of the function, the weaker the result will be, which does not detract from the the global minimum we want to approach.
As far as the result and the size are concerned, something like would have a relation:

$$r = dim * C$$

## 6 Conclusions

One method is better than another. I believe that not every method can be classified as the best. It simply depends on the context and we as humans must use our reasoning to decide what and when is best in context with the problem. In the future I could run the experiment using some characteristics being variable, such as computing machinery or different precision.

## References


[1] Wiley, De Jong 1's function figure
https://onlinelibrary.wiley.com/cms/asset/ab373f57-0fc8-41a3-99bb-a85c8931e035/cplx21560-fig-0012-m.png

[2] Sfu, Schwefel's function figure
https://www.sfu.ca/~ssurjano/schwef.html

[3] Sfu, Rastrigin's function figure
https://www.sfu.ca/~ssurjano/rastr.html

[4] Sfu, Michalewicz's function figure
https://www.sfu.ca/~ssurjano/michal.html

[5] Stackexchange.com
pgfplot: 3d function plot
https://tex.stackexchange.com/questions/194018/pgfplot-3d-function-plot

[6] Manual for Package pgfplots
http://pgfplots.sourceforge.net/pgfplots.pdf

[7] Wikipedia
Simulated Annealing
https://en.wikipedia.org/wiki/Simulated_annealing

[8] Simulated Annealing on Sciencedirect
https://www.sciencedirect.com/topics/materials-science/simulated-annealing

[9] Springer
Heuristic Search
https://link.springer.com/referenceworkentry/10.1007%2F978-1-4419-9863-7_875

[10] Baeldung
Example of Hill Climbing Algorithm in Java
https://www.baeldung.com/java-hill-climbing-algorithm

[11] Youtube
Hill Climbing Algorithm
https://www.youtube.com/watch?v=oSdPmxRCWws

[12] profs.info.uaic.ro
Teaching: Genetic Algorithms https://profs.info.uaic.ro/eugennc/teaching/ga/

[13] Stackexchange.com
Rounding to nearest integer symbol in Latex
https://tex.stackexchange.com/questions/433101/rounding-to-nearest-integer-symbol-in-latex

[14] Stackexchange.com
LaTeX figures side by side
https://tex.stackexchange.com/questions/37581/latex-figures-side-by-side

[15] Stackexchange.com
No natural log parentheses when typesetting
https://tex.stackexchange.com/questions/314609/no-natural-log-parentheses-when-typesetting

[16] Stackexchange.com
How to write a perfect equation parameters description
https://tex.stackexchange.com/questions/95838/how-to-write-a-perfect-equation-parameters-description