# Finding the minimum of a benchmark numerical function:

# Hill-Climbing vs Simulated Annealing

Badarau Dragos

November 12, 2021


## Abstract

This paper compares two heuristic algorithms: Hill-Climbing and Simulated Annealing using 4 different functions: De Jong 1, Schwefel's, Rastrigin's, Michalewicz's to visualize the global minimum in multiple scenarios. This project aims to show that optimizing the solution is based on iteratively improving the performance of these algorithms within a reasonable amount of time and memory space.


## ❖ Introduction

This article refers to non-determination searching techniques that can be used for solving optimization problems. There will be described the Hill Climbing Best and First Improvement and Simulated Annealing algorithms and how none of these algorithms can't prove as being the best because the result depends on many factors such as time-consuming. The mathematical functions tested vary from the number of distribution of local minima to the range of values and are presented by their graphics and formulas in the next sections.

## ❖ Motivation

The goal is to compare how the 3 algorithms behave in different situations and to analyze their approximate trajectory in finding the optimal minimum as the dimension increases. The problem consists of finding the local optimum between the range of a function.

## ❖ Methods

A hill-climbing algorithm is a local search algorithm that moves continuously upward (increasing) until the best solution is attained. This algorithm comes to an end when the peak is reached. Taking into consideration a set of candidates which we will establish randomly we will look for an improvement. The improvement function has two ways for the evaluation process: taking the first better neighbor of the current solution or the best neighbor of the current solution and determining if or not producing a better overall solution. Both of them start by selecting a random candidate solution which will be compared and changed if found other better solutions.

Hill Climbing Best Improvement is a local search algorithm that selects a successor of the current assignment that improves the most evaluation function (in the current case the evaluation function searches for a minimum based on the 4 mathematical functions observed).

Hill Climbing First Improvement is based on the idea that the first successor is better than the current is selected. In this case, the successor is not necessarily the best neighbor. It may be considered incomplete at some points because it usually gets stuck in a local maximum; a state that is not the optimal one. Escaping the local maxima is based on restarting the algorithm so that it is run several times with a selected initial state that keeps changing. Random restart hill-climbing performs better because the random restart is the best out of successive regular hill climbing's.

Simulated Annealing is a technique that allows downward steps to escape a local maximum. Annealing emulates the concept in metallurgy; where metals are heated to very high temperatures and then gradually cooled so their structure is frozen at a minimum energy configuration. The idea behind annealing is that at high temperatures the algorithm should jump out of a local maximum which proves efficiency in some cases. There are two conditions mandatory: the decrease of the temperature after each iteration and the halting criterion (the temperature which is settled from the start and decreases at each iteration). In the inner loop, a random neighbor of the solution is evaluated. The solution is gradually chosen such that the probability of accepting a worse solution decreases with temperature.

## ❖ Implementation details

Hill-Climbing:

- The solution is generated and saved in the binary format
- The number of each hill-climbing iteration is 1000
- The neighbor notion is based on the Hamming space technique

Simulated Annealing:

- The initial temperature is set to 100
- The halting condition is temperature 0.00001
- Temperature decreases by 1%
- The solution is generated and saved in the binary format

## ❖ DeJong Function

$$f_1(x) = \sum_{i=1}^{n} x_i^2 \qquad -5.12 \le x_i \le 5.12$$

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

DeJong's function is also known as a sphere model.

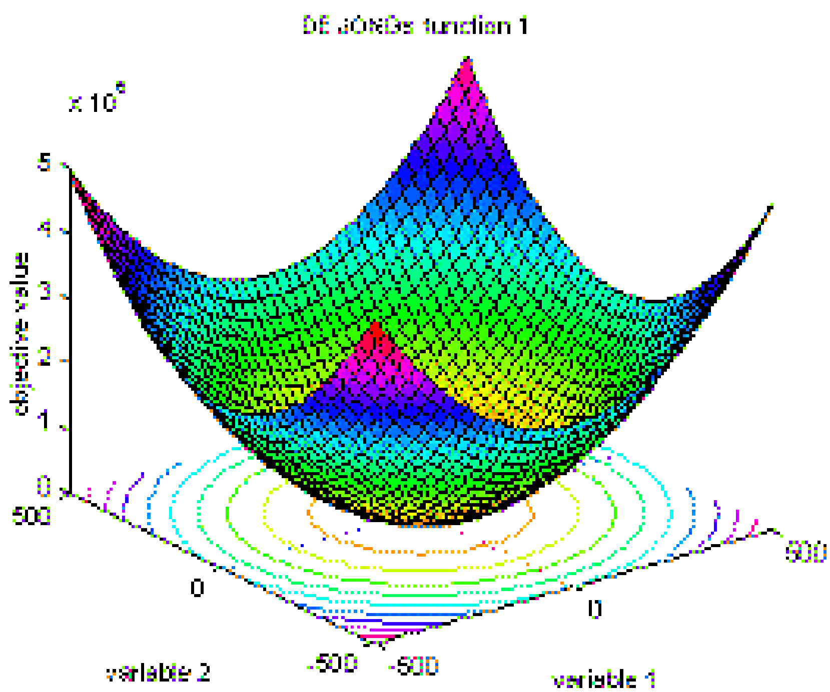



Figure 1: DeJong Function graph for n=2

- **First Improvement**

Because of the function shape, the first improvement saves a lot of time by picking the neighbor faster.

| Size | Mean | σ | Min | Max | Avg Time(s) | Min Time(s) | Max Time(s) |
|---|---|---|---|---|---|---|---|
| **5** | 0 | 0 | 0 | 0 | 25.315 | 22.3 | 27.85 |
| **10** | 0 | 0 | 0 | 0 | 165.875 | 162.8 | 171.98 |
| **30** | 0 | 0 | 0 | 0 | 4945.72 | 4720.31 | 5060.44 |

Values smaller than $10^{-5}$ were approximated to 0

- **Best Improvement**

Because this function has no local minima, the best improvement is slower than the first improvement.

| Size | Mean | σ | Min | Max | Avg Time(s) | Min Time(s) | Max Time(s) |
|---|---|---|---|---|---|---|---|
| **5** | 0 | 0 | 0 | 0 | 43.025 | 41.23 | 47.22 |
| **10** | 0 | 0 | 0 | 0 | 383.201 | 367.823 | 373.731 |
| **30** | 0 | 0 | 0 | 0 | 9436.78 | 9370.25 | 9746.33 |

Values smaller than $10^{-5}$ were approximated to 0

- **Simulated Annealing**

Simulated Annealing doesn't get to the local minima because the result is obtained approximately at the temperature stability point.

| Size | Mean | σ | Min | Max | Avg Time(s) | Min Time(s) | Max Time(s) |
|---|---|---|---|---|---|---|---|
| **5** | 0.00005 | 0.00005 | 0.00001 | 0.00024 | 9.9395 | 9.573 | 12.292 |
| **10** | 0.0005 | 0.00269 | 0.0002 | 0.0076 | 32.2395 | 31.891 | 35.747 |
| **30** | 6.4686 | 2.83332 | 1.98403 | 14.3654 | 214.055 | 209.55 | 224.59 |

Values smaller than $10^{-5}$ were approximated to 0

✓ Remarks

Taking into consideration the result, the fact that this function has no local minima implies that Hill Climbing First Improvement is the fastest solution and it matches the best results on each case.

## ❖ Schwefel's function

$$f_7(x) = \sum_{i=1}^{n} -x_i \cdot \sin\left(\sqrt{|x_i|}\right) \qquad -500 \le x_i \le 500$$

global minimum:
f(x)=-n·418.9829; x(i)=420.9687, i=1:n.

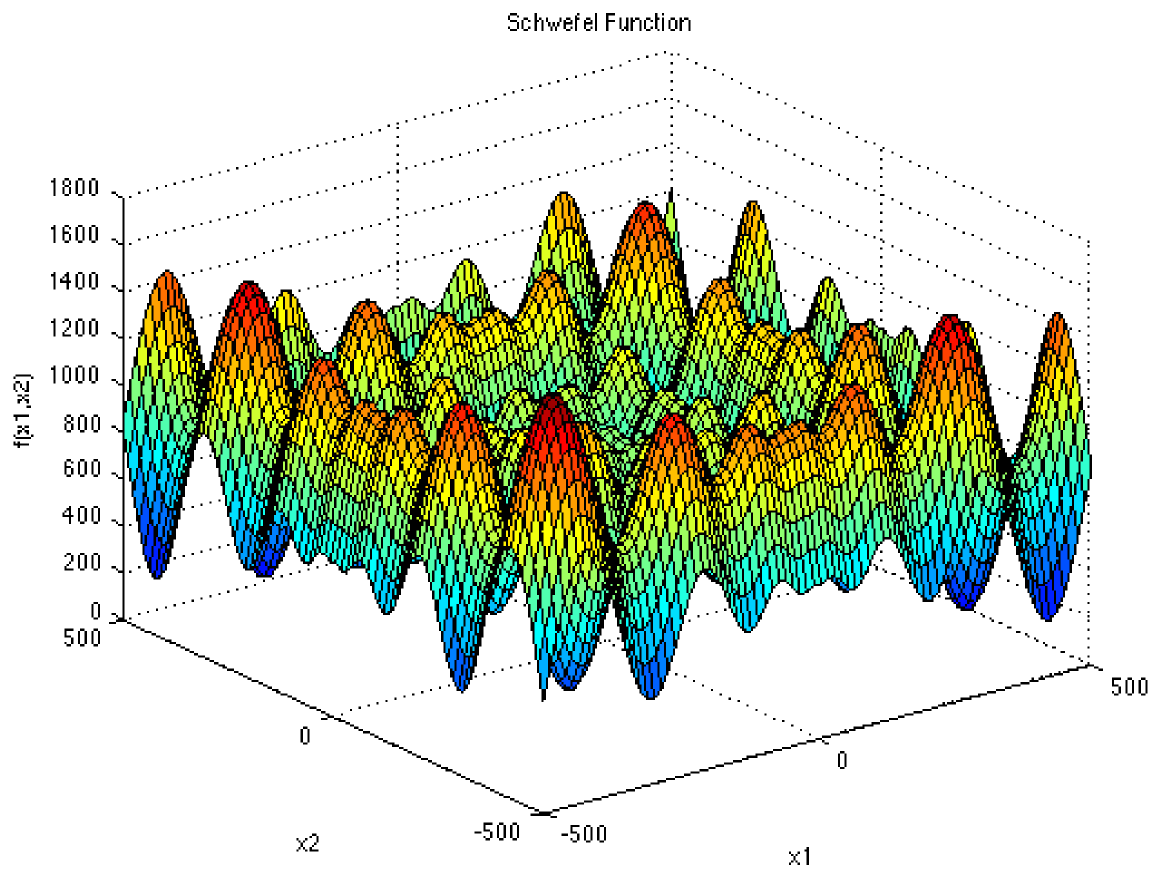



Figure 2: Schwefel's graph for n=2

- **First Improvement**

| Size | Mean | σ | Min | Max | Avg Time(s) | Min Time(s) | Max Time(s) |
|---|---|---|---|---|---|---|---|
| **5** | -2094.6 | 18.6512 | -2094.91 | -2026.34 | 54.323 | 51.134 | 61.615 |
| **10** | -3917.965 | 71.955 | -4076.2 | -3748.93 | 404.1695 | 388.6 | 429.581 |
| **30** | -10754.75 | 347.8381 | -11075.6 | -10005.2 | 11496.6 | 10168.2 | 13253.3 |

- **Best Improvement**

| Size | Mean | σ | Min | Max | Avg Time(s) | Min Time(s) | Max Time(s) |
|---|---|---|---|---|---|---|---|
| **5** | -2094.81 | 0.0879 | -2094.91 | -2094.6 | 103.51 | 90.021 | 109.302 |
| **10** | -4036.63 | 39.325 | -4091.53 | -3921.84 | 826.7645 | 730.627 | 862.864 |
| **30** | -11364.2 | 270.63 | -11525.3 | -11296.5 | 21117.01 | 200005.3 | 22843.3 |

- **Simulated Annealing**

| Size | Mean | σ | Min | Max | Avg Time(s) | Min Time(s) | Max Time(s) |
|---|---|---|---|---|---|---|---|
| **5** | -1606.715 | 179.61 | -1907.79 | -1150.34 | 14.885 | 14.253 | 16.427 |
| **10** | -3274.41 | 259.92 | -3783.27 | -2825.12 | 50.35 | 48.274 | 54.486 |
| **30** | -8630.26 | 1200.93 | -9244.47 | -6434.79 | 393.783 | 381.414 | 429.239 |

✓ Remarks

The fact that Schwefel’s function has a lot of local minimas makes it hard for the algorithms to find the solution in a reasonable amount of time.

## ❖ Rastrigin's function

$$f_6(x) = 10 \cdot n + \sum_{i=1}^{n} \left(x_i^2 - 10 \cdot \cos(2 \cdot \pi \cdot x_i)\right) \qquad -5.12 \leq x_i \leq 5.12$$

global minimum:
f(x)=0; x(i)=0, i=1:n.

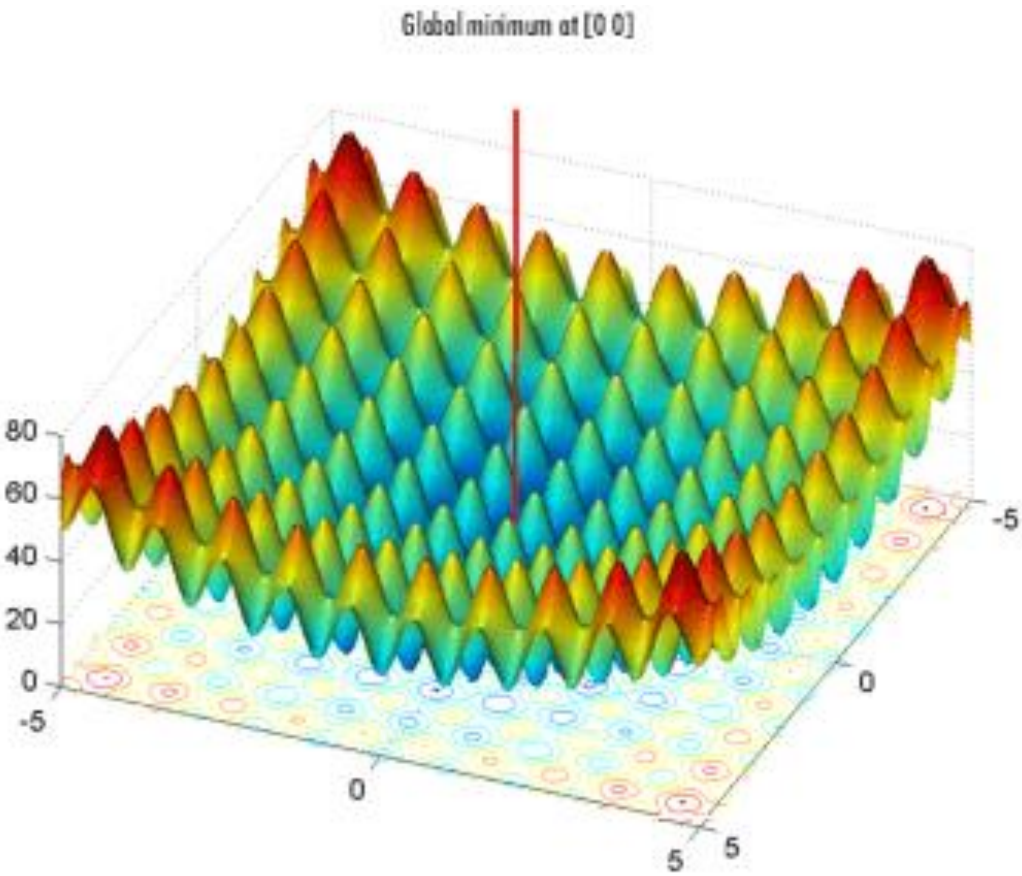



Figure 3: Rastrigin's function for n=2

- **First Improvement**

| Size | Mean | σ | Min | Max | Avg Time(s) | Min Time(s) | Max Time(s) |
|---|---|---|---|---|---|---|---|
| **5** | 0.99496 | 0.55344 | 0 | 1.98992 | 22.234 | 21.852 | 25.255 |
| **10** | 6.09019 | 1.05404 | 4.46159 | 8.17415 | 170.9445 | 163.361 | 176.389 |
| **30** | 35.69325 | 1.81602 | 32.626 | 39.6173 | 4250.575 | 4217.45 | 4263.3 |

- **Best Improvement**

| Size | Mean | σ | Min | Max | Avg Time(s) | Min Time(s) | Max Time(s) |
|---|---|---|---|---|---|---|---|
| **5** | 0 | 0.4854 | 0 | 0.9 | 39.689 | 38.543 | 41.144 |
| **10** | 4.46916 | 1.10244 | 2.23585 | 5.4616 | 295.139 | 286.763 | 316.014 |
| **30** | 27.6618 | 2.8946 | 22.0258 | 30.811 | 7693.25 | 7504.24 | 7893.21 |

- **Simulated Annealing**

| Size | Mean | σ | Min | Max | Avg Time(s) | Min Time(s) | Max Time(s) |
|---|---|---|---|---|---|---|---|
| **5** | 8.39012 | 6.4308 | 2.98494 | 25.4764 | 9.669 | 9.603 | 10.283 |
| **10** | 24.9224 | 9.9394 | 11.963 | 40.9022 | 34.162 | 32.189 | 35.449 |
| **30** | 109.188 | 23.793 | 65.8894 | 130.343 | 242.47 | 239.572 | 245.285 |

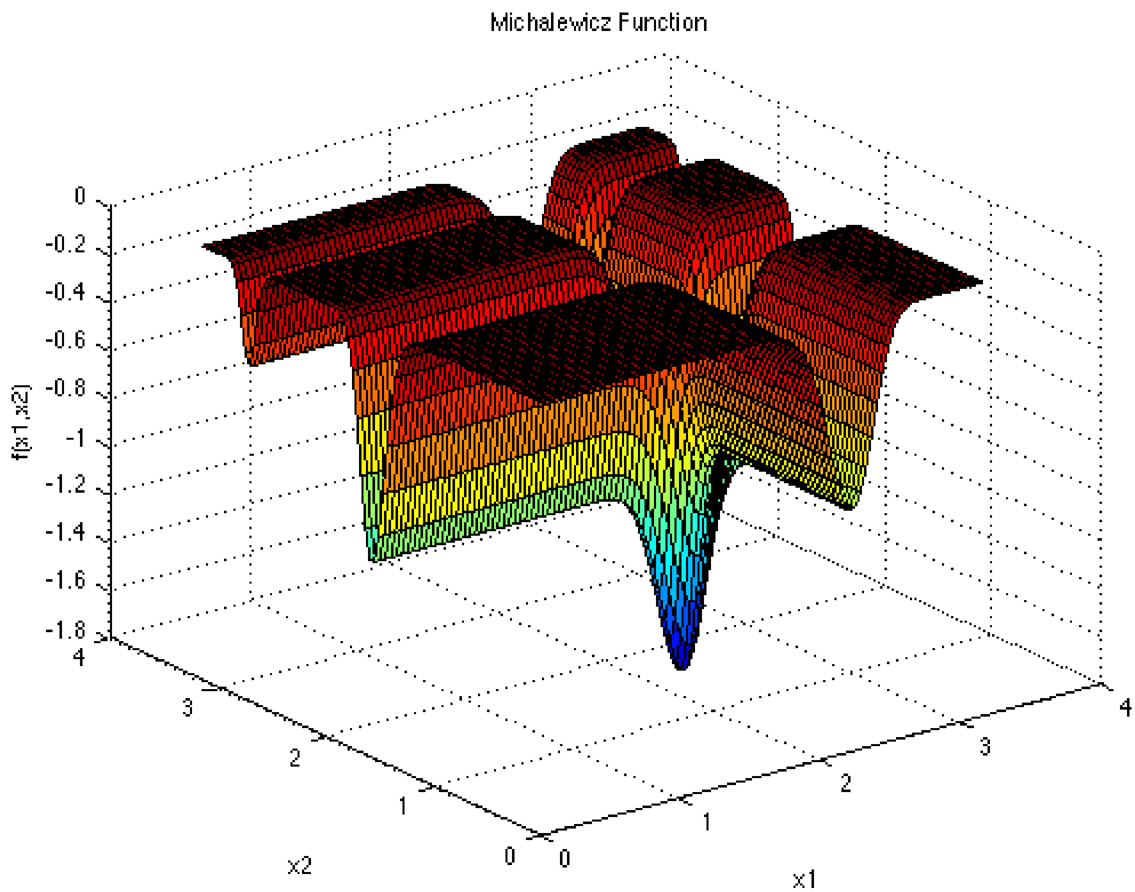
✓ Remarks

Simulated Annealing still proves to be faster at higher dimensions but it has poorer results compared to the Hill Climbing techniques.

## ❖ Michalewicz's function

$$f_{12}(x) = -\sum_{i=1}^{n} \sin(x_i) \cdot \left(\sin\left(\frac{i \cdot x_i^2}{\pi}\right)\right)^{2 \cdot m} \qquad i = 1:n,\ m = 10,\ 0 \le x_i \le \pi$$

global minimum:
f(x)=-4.687 (n=5); x(i)=???, i=1:n.
f(x)=-9.66 (n=10); x(i)=???, i=1:n.



- **First Improvement**

| Size | Mean | σ | Min | Max | Avg Time(s) | Min Time(s) | Max Time(s) |
|---|---|---|---|---|---|---|---|
| **5** | -4.6706 | 0.00171 | -4.68765 | -4.68199 | 21.907 | 20.735 | 22.869 |
| **10** | -9.23116 | 0.08205 | -9.43877 | -9.15703 | 150.690 | 144.312 | 160.413 |
| **30** | -26.0472 | 0.27051 | -26.5291 | -25.8651 | 3573.67 | 3437.17 | 3706.83 |

- **Best Improvement**

| Size | Mean | σ | Min | Max | Avg Time(s) | Min Time(s) | Max Time(s) |
|---|---|---|---|---|---|---|---|
| **5** | -4.68753 | 0.00151 | -4.68766 | -4.68297 | 38.641 | 37.317 | 39.515 |
| **10** | -9.40084 | 0.06715 | -9.60708 | -9.27461 | 272.634 | 260.449 | 287.492 |
| **30** | -26.5894 | 0.18572 | -27.1866 | -26.0415 | 6431.25 | 6222.04 | 6397.65 |

- **Simulated Annealing**

| Size | Mean | σ | Min | Max | Avg Time(s) | Min Time(s) | Max Time(s) |
|---|---|---|---|---|---|---|---|
| **5** | -3.97830 | 0.307938 | -4.47661 | -3.51984 | 9.409 | 9.367 | 9.699 |
| **10** | -7.67361 | 0.46761 | -8.86977 | -7.53411 | 31.8915 | 31.292 | 33.78 |
| **30** | -19.1731 | 1.78141 | -22.7272 | -17.8916 | 235.499 | 235.272 | 243.125 |

## ❖ Overall Comparison over the results

| **Algorithm result dimension 5** | | | |
|---|---|---|---|
| **Function** | HC First | HC BEST | SA |
| **De Jong** | 0 | 0 | 0.00005 |
| **Schwefel** | -2094.6 | -2094.81 | -1606.715 |
| **Rastrigin** | 0.9946 | 0 | 8.39012 |
| **Michalewicz** | -4.68796 | -4.68754 | -3.987305 |

| **Algorithm result dimension 10** | | | |
|---|---|---|---|
| **Function** | HC First | HC BEST | SA |
| **De Jong** | **0** | **0** | 0.0005 |
| **Schwefel** | -3917.965 | -4036.63 | -3274.41 |
| **Rastrigin** | 6.09019 | 4.46916 | 24.9224 |
| **Michalewicz** | -9.23116 | -9.40084 | -7.67361 |

| **Algorithm result dimension 30** | | | |
|---|---|---|---|
| **Function** | HC First | HC BEST | SA |
| **De Jong** | 0 | 0 | 6.4686 |
| **Schwefel** | -10754.75 | -11364.2 | -8630.26 |
| **Rastrigin** | 35.69325 | 27.6618 | 109.188 |
| **Michalewicz** | -26.0472 | -26.5894 | -19.1731 |

This is the MEDIAN values' evolution compared to the dimensions

## ❖ Discussion over the results

The problem of finding the minimum of a function concludes that the result is based on the iteration values and the dimensional input. Taking into consideration the differences between the approached functions, the distribution of local minima has a big impact over the overall results because it influenced the neighbor generation method. Observing the precision and time, allocating more time for a large number of iterations produces a closer outcome to the optimal minimum results. The Hill Climbing method proved to be slower but more precise using

1000 iterations, the most important aspect being the fact that the neighbors were chosen based on the most important bit at the time. Best Improvement Hill Climbing returns the best values with the most time spent in comparation with First Improvement Hill Climbing which gives decent results in half the time spent for Best Improvement algorithm. As things stand, Simulated Annealing approximates faster resulting in weaker outcome, the halting-criterion, termination condition and initial temperature being the decisive aspects of the method.

## ❖ Conclusion

This report points to the performance of the Hill Climbing and Simulated Annealing algorithms. In the context of 4 different numerical functions, there were significant differences and interpretations based on the time scaling, closeness to the minimum of each function, and the dimensions used. There were used two instances of the Hill Climbing: First and Best Improvement and a Simulated Annealing with a linear cooling time. The Best Improvement Hill Climbing proved to be the most time-consuming with better overall results than First Improvement Hill Climbing while the Simulated Annealing algorithm needs less time to approximate a result.

## ❖ Bibliography


- Implementation of the genetic algorithm
  **https://profs.info.uaic.ro/~eugennc/teaching/ga/**
- The graphs and functions used
  **http://www.geatbx.com/docu/fcnindex-01.html#P140_6155**
- Additional informations about hill-climbing
  **https://www.section.io/engineering-education/understanding-hill-climbing-in-ai/**
- Comparison between simulated annealing and hill climbing
  **https://academicjournals.org/article/article1379672838_Patrick.pdf**
- General aspects about the algorithms
  **https://www.youtube.com/watch?v=FyyVbuLZav8&t=278s**
  **https://www.youtube.com/watch?v=eBmU1ONJ-os&t=109s**
  **https://www.youtube.com/watch?v=oSdPmxRCWws**
- The convertor used
  https://www.ilovepdf.com/word_to_pdf