Comparison of Hill Climbing variants and Simulated Annealing algorithms on a set of functions

Tudose George-Daniel

4^{th} November, 2020

1 Introduction

We want to compare how fast and accurate these two heuristic algorithms can find the global optimum of a set of functions and how these measurements differ between them.

1.1 Motivation

For determining either the global optimum of a function the deterministic approach usually takes a huge amount of time. So we are using heuristic approaches to get approximate results. We want the compare the results of two heuristic algorithms on a set of functions.

2 Method

The two heuristic algorithms that have been chosen are the *Hill Climbing* and the *Simulated Annealing* algorithms.

2.1 Hill Climbing

The algorithm randomly selects a candidate solution as the starting point. It then tries to improve the candidate in an iterative manner by searching through the neighbours that are at a Hamming distance of 1 of it. If it couldn't find a better neighbour then the algorithm stops as it is in a local minimum.

The algorithm has two variations and the difference is how it improves the candidate solution. The two approaches are *Best Improvement* and *First Improvement*. The former iterates through all the neighbours and updates the candidate solution with the best neighbour whilst the latter updates the candidate with the first neighbour that represents a better solution.

The iterative version of the algorithm is usually used in practice to use a different initial candidate solution for the algorithm.

2.2 Simulated Annealing

The algorithm selects a candidate solution at random as the starting point. Compared to the *Hill Climbing* algorithm it can make uphill jumps based off a temperature component. Implementation-wise, the algorithm has two nested loops which we'll refer to as the *inner loop* and the *outer loop*. The former checks random neighbours in Hamming distance of 1 of the candidate solution for a set number of times and it tries to improve the candidate solution. The possibility of a jump to a worse candidate solution for a possibility to get a better result is based on a probability which is less likely to happen as the temperature cools off. The *outer loop* is related to the temperature as it cools down towards an equilibrium state where it would be an optimum.

3 Experiment

For the experiment, a set of functions was selected and the presented heuristic algorithms have been applied on them to find their global minimum. To measure the amount of time it takes for a function we've tested the algorithms for a different number of dimensions. The precision of the values that was used is 5 decimals. The sample size is 30.

The experiments have been benchmarked on an **AMD Ryzen 5 3600x**.

3.1 De Jong's function 1

It is also known as the sphere model. It is continuous, convex and unimodal. The known global minimum for all dimensions is $f(x) = 0\,; x(i) = 0\,; i = 1 : n$ [3]

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

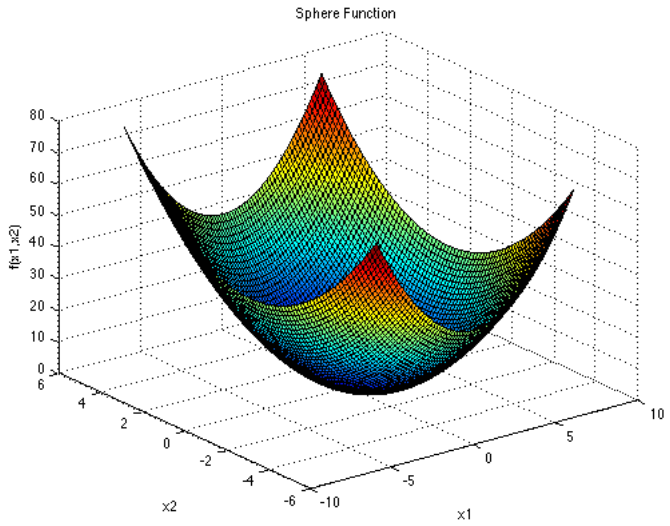


Figure 1: De Jong's function 1[4]

3.1.1 Best Improvement Hill Climbing

Table entrees equal to 0 were values that were smaller than 10^{-5} so they were approximated to 0.

Dimensions	μ	Minimum	Maximum	σ
5	0	0	0	0
10	0	0	0	0
30	0	0	0	0

Table 1: Values for HC Best Improvement for De Jong's function

The time values were measured in seconds. The observations made in this section apply onwards to the other tables as well.

Dimensions	μ	Minimum	Maximum	σ
5	25.752	24.909	26.263	0.278
10	152.425	145.887	229.413	14.78
30	404.555	2353.24	2477	25.019

Table 2: Time values for HC Best Improvement for De Jong's function

3.1.2 First Improvement Hill Climbing

Dimensions	μ	Minimum	Maximum	σ
5	0	0	0	0
10	0	0	0	0
30	0	0	0	0

Table 3: Values for HC First Improvement for De Jong's function

Dimensions	μ	Minimum	Maximum	σ
5	13.738	13.257	14.195	0.204
10	77.707	75.181	80.323	1.394
30	1231.297	1209.03	1259.03	12.413

Table 4: Time values for HC First Improvement for De Jong's function

3.1.3 Simulated Annealing

The starting temperature is 150 and cools off by 0.5% of its current value. This observation applies onwards to the other tables as well.

Dimensions	μ	Minimum	Maximum	σ
5	0	0	0	0
10	0	0	0	0
30	0	0	0	0

Table 5: Values for Simulated Annealing for De Jong's function

Dimensions	μ	Minimum	Maximum	σ
5	25.605	24.461	26.185	0.422
10	36.531	33.913	38.935	1.248
30	61.965	58.341	64.115	1.460

Table 6: Time values for Simulated Annealing for De Jong's function

3.1.4 Comparison

In this section we'll compare the three algorithms for the same number of dimensions between them and also to the known global minimum. This observation applies onward to the other tables as well.

Dimensions	Algorithm	Expected result	μ_{value}	σ_{value}	μ_{time}
5	HC Best	0	0	0	25.752
	HC First	0	0	0	13.738
	SA	0	0	0	25.605
10	HC Best	0	0	0	152.425
	HC First	0	0	0	77.707
	SA	0	0	0	36.53
30	HC Best	0	0	0	2404.555
	HC First	0	0	0	1231.297
	SA	0	0	0	61.965

Table 7: Comparison of the algorithms for De Jong's function

3.2 Schwefel's function 7

Schwefel's function is deceptive in that the global minimum is geometrically distant, over the parameter space, from the next best local minima. Therefore, the search algorithms are potentially prone to convergence in the wrong direction. The global minimum varies based on the number of dimensions:
$f(x) = -n \cdot 418.9829\ ; x(i) = 420.9687\ ; i = 1 : n.$ [3]

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

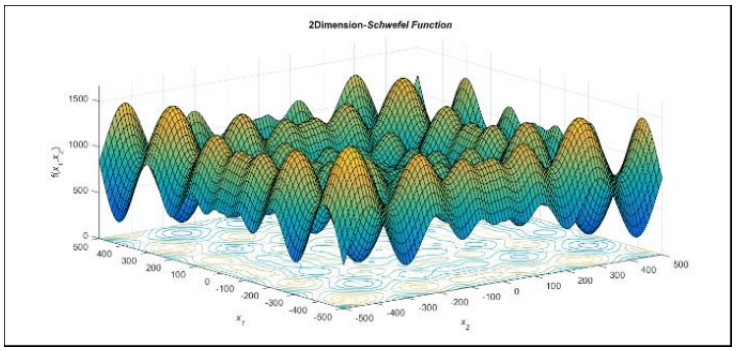


Figure 2: Schwefel's function 7[4]

3.2.1 Best Improvement Hill Climbing

For 5 dimentions, the Global minimum is -2094.9145. For 10 dimensions is -4189.829. For 30 is -12569.487.

Dimensions	μ	Minimum	Maximum	σ
5	-2094.89666	-2094.91	-2094.81	0.03457
10	-4161.61233	-4189.62	-4087.02	30.51827
30	-11582.5	-11854.2	-11416.4	124.10212

Table 8: Values for HC Best Improvement for Schwefel's function

Dimensions	μ	Minimum	Maximum	σ
5	48.813	45.403	50.417	1.151
10	282.125	275.495	288.562	2.64
30	5007.104	4881.24	5138.93	69.329

Table 9: Time values for HC Best Improvement for Schwefel's function

3.2.2 First Improvement Hill Climbing

Dimensions	μ	Minimum	Maximum	σ
5	-2094.78333	-2094.91	-2094.71	0.06396
10	-4009.70033	-4121.05	-3917.91	55.68046
30	-11050.50333	-11262	-10849.2	116.22008

Table 10: Values for HC First Improvement for Schwefel's function

Dimensions	μ	Minimum	Maximum	σ
5	26.827	26.256	27.37	0.293
10	157.386	152.366	164.987	2.878
30	2762.576	2701.58	2825.18	37.429

Table 11: Time values for HC First Improvement for Schwefel's function

3.2.3 Simulated Annealing

Dimensions	μ	Minimum	Maximum	σ
5	-2094.74166	-2094.91	-2094.5	0.09512
10	-4176.41466	-4189.72	-4071.18	26.67730
30	-12550.72666	-12569	-12450.5	31.07277

Table 12: Values for the Simulated Annealing for Schwefel's function

Dimensions	μ	Minimum	Maximum	σ
5	29.187	27.516	30.182	0.53
10	43.371	41.594	45.468	0.974
30	81.102	79.445	82.673	0.99239

Table 13: Time values for Simulated Annealing for Schwefel's function

3.2.4 Comparison

Dimensions	Algorithm	Expected result	μ_{value}	σ_{value}	μ_{time}
5	HC Best	-2094.9145	-2094.89666	0.03457	48.813
	HC First	-2094.9145	-2094.78333	0.06396	26.827
	SA	-2094.9145	-2094.74166	0.09512	29.187
10	HC Best	-4189.829	-4161.61233	30.51827	282.125
	HC First	-4189.829	-4009.70033	55.68046	157.386
	SA	-4189.829	-4176.41466	26.67730	43.371
30	HC Best	-12569.487	-11582.5	124.10212	5007.104
	HC First	-12569.487	-11050.50333	116.22008	2762.576
	SA	-12569.487	-12550.72666	31.07277	81.102

Table 14: Comparison of the algorithms for Schwefel's function

3.3 Rastrigin's function 6

This function has many local minima thus this test function is multimodal. The location of the minima are regularly distributed. The known global minimum is $f(x) = 0;\ x(i) = 0\ ;\ i = 1 : n$ [3]

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

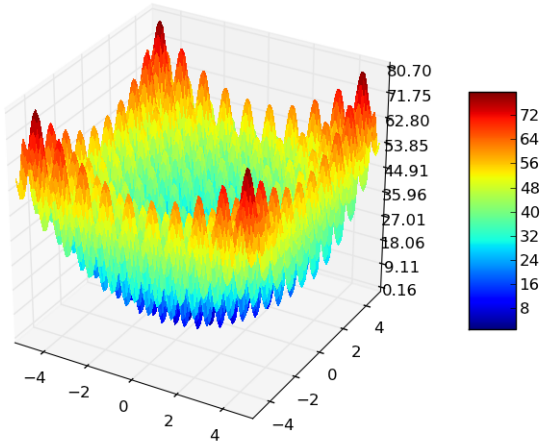


Figure 3: Rastrigin's function 6[4]

3.3.1 Best Improvement Hill Climbing

Dimensions	μ	Minimum	Maximum	σ
5	0	0	0	0
10	2.45184	1.23582	3.23079	0.58735
30	22.64780	17.0972	26.3293	2.62216

Table 15: Values for HC Best Improvement for Rastrigin's function

Dimensions	μ	Minimum	Maximum	σ
5	23.008	21.578	24.09	0.476
10	132.694	128.22	136.734	2.215
30	2226.845	2166.13	2300.53	32.270

Table 16: Time values for HC Best Improvement for Rastrigin's function

3.3.2 First Improvement Hill Climbing

Dimensions	μ	Minimum	Maximum	σ
5	0.19899	0	0.99495	0.40478
10	3.53777	1.98992	5.22576	0.88426
30	30.98477	26.067	35.0582	2.10824

Table 17: Values for HC First Improvement for Rastrigin's function

Dimensions	μ	Minimum	Maximum	σ
5	13.415	12.767	13.924	0.285
10	76.554	73.33	78.884	1.432
30	1272.325	1229.89	1312.69	21.476

Table 18: Time values for HC First Improvement for Rastrigin's function

3.3.3 Simulated Annealing

Dimensions	μ	Minimum	Maximum	σ
5	1.98318	0	6.15924	1.73951
10	4.01592	0.994959	6.91333	1.62065
30	15.36175	7.96473	27.9557	4.99550

Table 19: Values for Simulated Annealing for Rastrigin's function

Dimensions	μ	Minimum	Maximum	σ
5	27.888	26.326	28.98	0.696
10	40.753	37.806	43.056	1.46896
30	72.537	70.277	75.595	1.233

Table 20: Time values for Simulated Annealing for Rastrigin's function

3.3.4 Comparison

Dimensions	Algorithm	Expected result	μ_{value}	σ_{value}	μ_{time}
5	HC Best	0	0	0	23.008
	HC First	0	0.19899	0.40478	13.415
	SA	0	1.98318	1.73951	27.888
10	HC Best	0	2.45184	0.58735	132.694
	HC First	0	3.53777	0.88426	76.55433
	SA	0	4.01592	1.62065	40.75356
30	HC Best	0	22.64780	2.62216	2226.845
	HC First	0	30.98477	2.10824	1272.32566
	SA	0	15.36175	4.99550	72.537

Table 21: Comparison of the algorithms for Rastrigin's function

3.4 Michalewicz's function 12

The Michalewicz function is a multimodal test function (n! local optima). The parameter m defines the "steepness" of the valleys or edges. Larger m leads to more difficult search. For very large m the function behaves like a needle in the haystack (the function values for points in the space outside the narrow peaks give very little information on the location of the global optimum).[3]

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

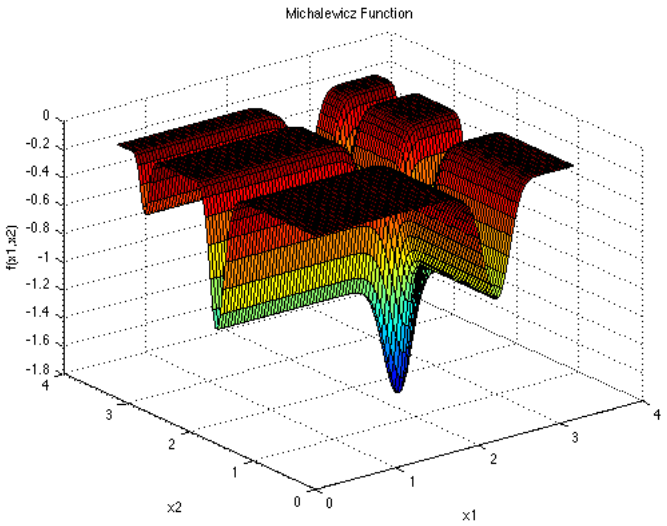


Figure 4: Michalewicz's function 12[4]

3.4.1 Best Improvement Hill Climbing

The global minimum varies based on the number of dimensions. For 5 dimensions is -4.68765. For 10 dimensions is -9.66015. For 30 is -29.63088.[3][9]

Dimensions	μ	Minimum	Maximum	σ
5	-4.68765	-4.68766	-4.68764	0
10	9.54036	-9.65456	-9.41135	0.06162
30	-27.41567	-27.7992	-27.15155	0.17220

Table 22: Values for HC Best Improvement for Michalewicz's function

Dimensions	μ	Minimum	Maximum	σ
5	30.285	29.824	30.775	0.219
10	184.405	179.455	188.525	2.338
30	3378.556	3321.03	3447.38	30.515

Table 23: Time values for HC Best Improvement for Michalewicz's function

3.4.2 First Improvement Hill Climbing

Dimensions	μ	Minimum	Maximum	σ
5	-4.68754	-4.68766	-4.687	0.00020
10	-9.44665	-9.53551	-9.33632	0.05505
30	-26.92771	-27.4403	-26.5327	0.23080

Table 24: Values for HC First Improvement for Michalewicz's function

Dimensions	μ	Minimum	Maximum	σ
5	6.924	16.244	17.429	0.280
10	102.669	99.438	107.323	1.428
30	1919.973	1875.25	1966.04	20.510

Table 25: Time for HC First Improvement for Michalewicz's function

3.4.3 Simulated Annealing

Dimensions	μ	Minimum	Maximum	σ
5	-4.59332	-4.68766	-4.4483	0.08229
10	-9.38573	-9.58097	-8.99229	0.12933
30	-28.63814	-29.2608	-27.7154	0.31448

Table 26: Values for Simulated Annealing for Michalewicz's function

Dimensions	μ	Minimum	Maximum	σ
5	37.624	35.827	38.811	0.670
10	62.784	59.175	91.004	5.415
30	137.332	130.593	139.677	1.788

Table 27: Time values for Simulated Annealing for Michalewicz's function

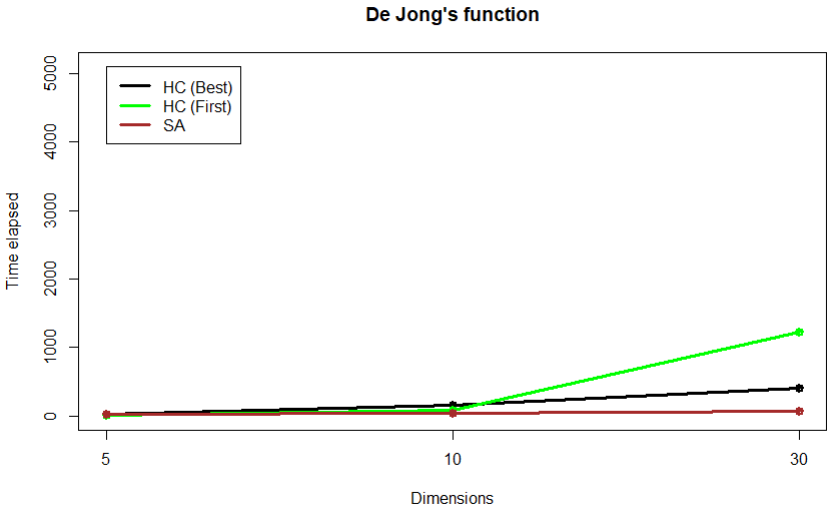
3.4.4 Comparison

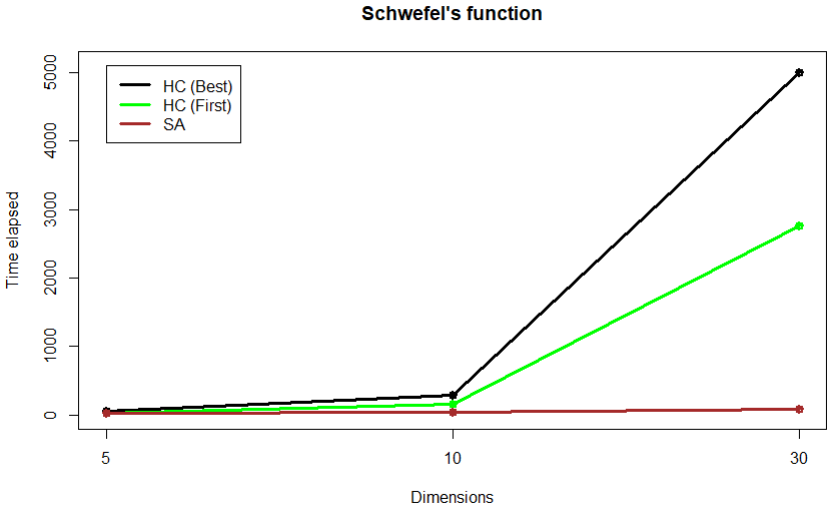
Dimensions	Algorithm	Expected result	μ_{value}	σ_{value}	μ_{time}
5	HC Best	-4.687	-4.68765	0	30.285
	HC First	-4.687	-4.68754	0.00020	16.924
	SA	-4.687	-4.59332	0.08229	37.624
10	HC Best	-9.66	-9.54036	0.06162	184.40516
	HC First	-9.66	-9.44665	0.05505	102.669
	SA	-9.66	-9.38573	0.12933	62.784
30	HC Best	-29.63088	-27.41567	0.17220	3378.556
	HC First	-29.63088	-26.92771	0.23080	1919.973
	SA	-29.63088	-28.63814	0.31448	137.332

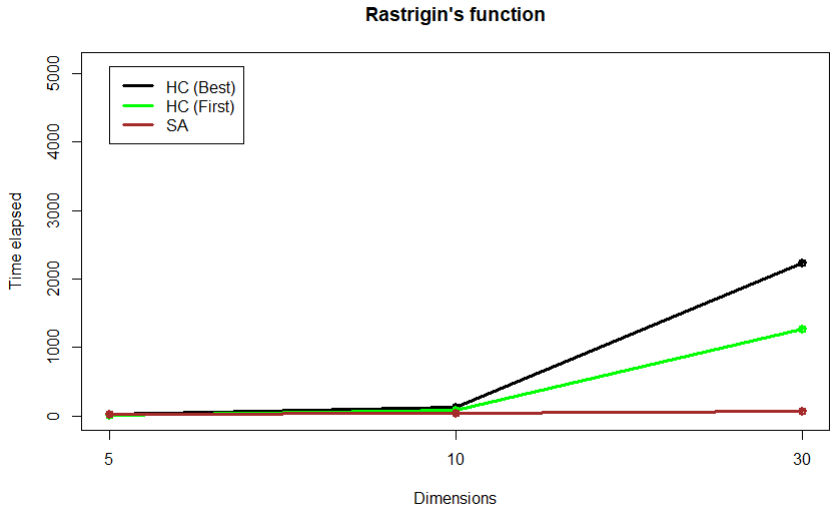
Table 28: Comparison of the algorithms for Michalewicz's function

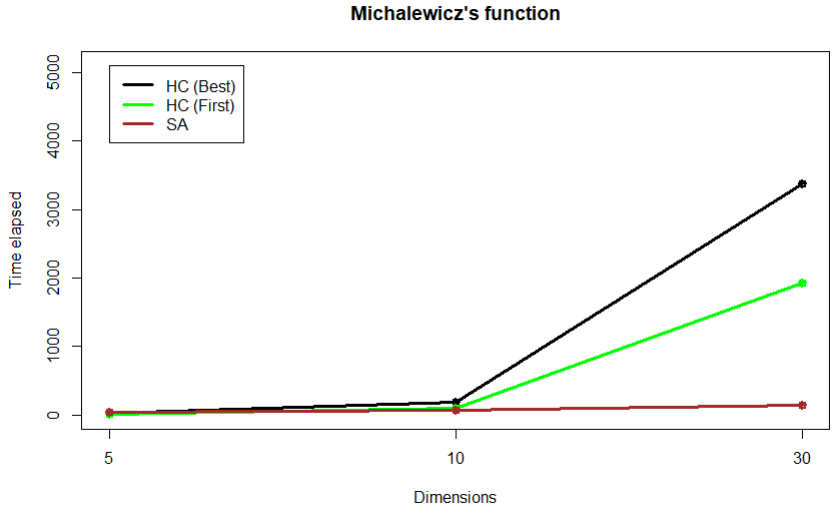
4 Observations

The first observation to be made is that the number of dimensions directly affects the time it takes for a result to be acquired. In the next plots, the values for the μ of the time have been considered.


De Jong's function
HC (Best)
HC (First)
SA
Time elapsed
0
1000
2000
3000
4000
5000
5
10
30
Dimensions


Schwefel's function
HC (Best)
HC (First)
SA
Time elapsed
0
1000
2000
3000
4000
5000
5
10
30
Dimensions





We can observe that the *Best Improvement* version of the *Hill Climbing* algorithm tends to take a lot of time to get a result as it needs to iterate through all the neighbours of the candidate solution in trying to improve it. Meanwhile, the *Simulated Annealing* algorithm takes a lot less time compared to the other two. For a function like De Jong's, we can observe in Table 7 that it has the same results while having a really good μ of time for a higher number of dimensions. Meanwhile, for functions with more local minima, it obtains results closer to the actual global minimum on average(μ) in a reasonable amount of time. By choosing a fitting temperature and cooling rate the *Simulated Annealing* can be tweaked to get better results in a faster manner compared to the *Hill Climbing Algorithm*.

5 Conclusions

Both algorithms are able to get good approximate solutions in a reasonable amount of time. The *Hill Climbing* algorithm tends to have a higher elapsed time compared to the *Simulated Annealing* one. Between the two variants of the former, the *First Improvement* one tended to get worse results than the *Best Improvement one* but not far from its results while also having almost half of its elapsed time. The

Simualted Annealing algorithm had the closest results to the the actual minima of the functions in the fastest time of the three algorithms. By tweaking the temperature and cooling rate it can be adjusted to have even better results.

References

[1] Eugen Croitoru, Teaching: Genetic Algorithms
https://profs.info.uaic.ro/~eugennc/teaching/ga/

[2] Eugen Croitoru, LaTeXexample
https://gitlab.com/eugennc/teaching/-/tree/master/GA

[3] GEATbx, Information about De Jong's, Schwefel's, Rastrigin's and Michalewicz's functions
http://www.geatbx.com/docu/fcnindex-01.html

[4] De Jong
https://www.sfu.ca/~ssurjano/spheref.html
Schwefel
https://www.researchgate.net/figure/Two-dimension-Schwefel-function_fig1_311993569
Rastrigin
https://www.cs.unm.edu/~neal.holts/dga/benchmarkFunction/rastrigin.html
Michalewicz
https://www.sfu.ca/~ssurjano/michal.html

[5] Value of π
https://en.wikipedia.org/wiki/Pi->PI

[6] Text merger
https://www.filesmerge.com/merge-text-files->textmerger

[7] R threads
https://www.math.ucla.edu/~anderson/rw1001/library/base/html/paste.html
https://stat.ethz.ch/R-manual/R-devel/library/base/html/write.html

[8] C++ assisting threads
Thread
http://www.cplusplus.com/reference/thread/thread/
http://www.cplusplus.com/reference/thread/thread/id/
http://www.cplusplus.com/reference/thread/this_thread/
https://www.geeksforgeeks.org/multithreading-in-cpp/

Random and Mersenne Twister 19937 generator
https://www.cplusplus.com/reference/random/
http://www.cplusplus.com/reference/chrono/
https://codeforces.com/blog/entry/61587
https://www.guyrutenberg.com/2014/05/03/c-mt19937-example/

Uniform real distribution
https://en.cppreference.com/w/cpp/numeric/random/uniform_real_distribution http://www.cplusplus.com/reference/random/uniform_real_distribution/

[9] Charlie Vanaret, Jean-Baptiste Gotteland, Nicolas Durand, Jean-Marc Alliot Certified Global Minima for a Benchmark of Difficult Optimization Problems 2014