Homework T1

David-Andrei Viziteu

November 1, 2020

Abstract

Comparing the accuracy and execution time of Hill Climbing algorithm against Simulated Annealing algorithm they are being used to find the minimum value of some mathematical functions on a closed interval.

1 Introduction

This report studies the algorithmic optimization problem of finding the minimum value of some mathematical functions on their domain of definition using Hill Climbing (first improvement and best improvement) and Simulated Annealing approaches.

The implementation of the methods is described in the next section.

The results that show the differences between accuracy, precision and execution time of the algorithms are outlined in the "Experiment" section of this document.

2 Methods (algorithms' descriptions):

The functions have been tested for $n \in \{5, 10, 30\}$, 30 times (30 samples) for each dimension. To avoid double precision errors (e.g. -5.12 + 0.1 will output -5.0200000000000005 in Python 3.8), the vector containing the x_i values (the solutions) is represented as an array of bits as follows [7]:

- We choose d, a decimal precision. In our case we have $d = 2$.
- The function's domain $[a, b]$ is divided in $N = (b - a) * 10^d$ equal sub domains. To be able to represent an x_i value ($x_i \in [a, b]$), we will need $\lceil \log_2(N) \rceil$ bits. To convert the binary representation of x_i to a human-readable value, we will use the following formula:

$$x_i = \frac{a + decimal(x_{i(binary)}) * (b - a)}{(2^n - 1)}$$

- As a candidate solution (array of $x_i, i \in [0, n]$) is represented as an array of bits, we will generate a neighbour of the solution by negating exactly one bit.

2.1 Hill Climbing - general description: [6]

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.

2.1.1 Iterated Hill Climbing - first improvement (pseudocode):

$GlobalMinimum \leftarrow \infty$
Repeat steps 1-4 10.000 times:

1. Generate a candidate solution $initialSolution$ (array of bits) randomly.

2. Compute and store the value of $f(initialSolution)$.

3. For every bit of the candidate solution do:

    - Negate that bit, thus obtaining a new solution.
    - Compute $f(newSolution)$ and store the result.
    - if $f(initialSolution) > f(newSolution)$ then $initialSolution \leftarrow newSolution$. Go to step 2.
    - Negate again that bit, thus obtaining the initial solution.

4. if $GlobalMinimum > f(initialSolution)$ then $GlobalMinimum \leftarrow initialSolution$.

return $GlobalMinimum$ (therefore obtaining one sample)

2.1.2 Iterated Hill Climbing - Best improvement (pseudocode):

$GlobalMinimum \leftarrow \infty$
Repeat steps 1-5 10.000 times:

1. Generate a candidate solution $initialSolution$ (array of bits) randomly.

2. $smallestResult \leftarrow f(initialSolution)$.

3. For every bit of the candidate solution do:

    - $bestSolution \leftarrow None$
    - Negate that bit, thus obtaining a new solution.
    - Compute $f(newSolution)$ and store the result.
    - if $smallestResult > f(newSolution)$:
      then $smallestResult \leftarrow f(newSolution); bestSolution \leftarrow newSolution$
    - Negate again that bit, thus obtaining the initial solution.

4. if $bestSolution \neq None$:
   then $initialSolution \leftarrow bestSolution$; go to step 3.

5. if $smallestResult < GlobalMinimum$ then $GlobalMinimum \leftarrow smallestResult$.

return $GlobalMinimum$ (therefore obtaining one sample)

2.2 Simulated Annealing:[5]

The name of the algorithm comes from annealing in metallurgy, a technique involving heating and controlled cooling of a material to increase the size of its crystals and reduce their defects. Both are attributes of the material that depend on their thermodynamic free energy.

This notion of slow cooling implemented in the simulated annealing strategy is interpreted as a slow decrease in the probability of accepting worse solutions as the solution space is explored. Accepting worse solutions allows for a more extensive search for the global optimal solution. In general, simulated annealing algorithms work as follows. The temperature progressively decreases from an initial positive value to zero. At each time step, the algorithm randomly selects a solution close to the current one, measures its quality, and moves to it according to the temperature-dependent probabilities of selecting better or worse solutions, which during the search respectively remain at 1 (or positive) and decrease towards zero.

2.2.1 Cooling Schedule:

For the experiment I have tried using a probabilistic cooling scheme to utilise the features of both logarithmic and linear schedules:

$$random[0,1) < \beta \begin{cases} T_i \leftarrow \frac{1}{1+i} * \frac{T_0}{\log_{10} 1+i}(\text{logarithmic cooling schedule}) & \text{if TRUE} \\ T_i \leftarrow 0.967 * T_{i-1}(\text{linear cooling schedule}) & \text{if FALSE} \end{cases}$$

for an arbitrary chosen $\beta \in [0, 1)$.

Due to the nature of our test functions, the closer to 0 β was, the better the results of the algorithm. Furthermore, I chose $\beta = 0$, thus making the probability of the algorithm to chose the linear cooling schedule every iteration equally to 1.

3 Experiment

3.1 Schwefel's Function 7[3]:

The Schwefel function is complex, with many local minima. It's 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.

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

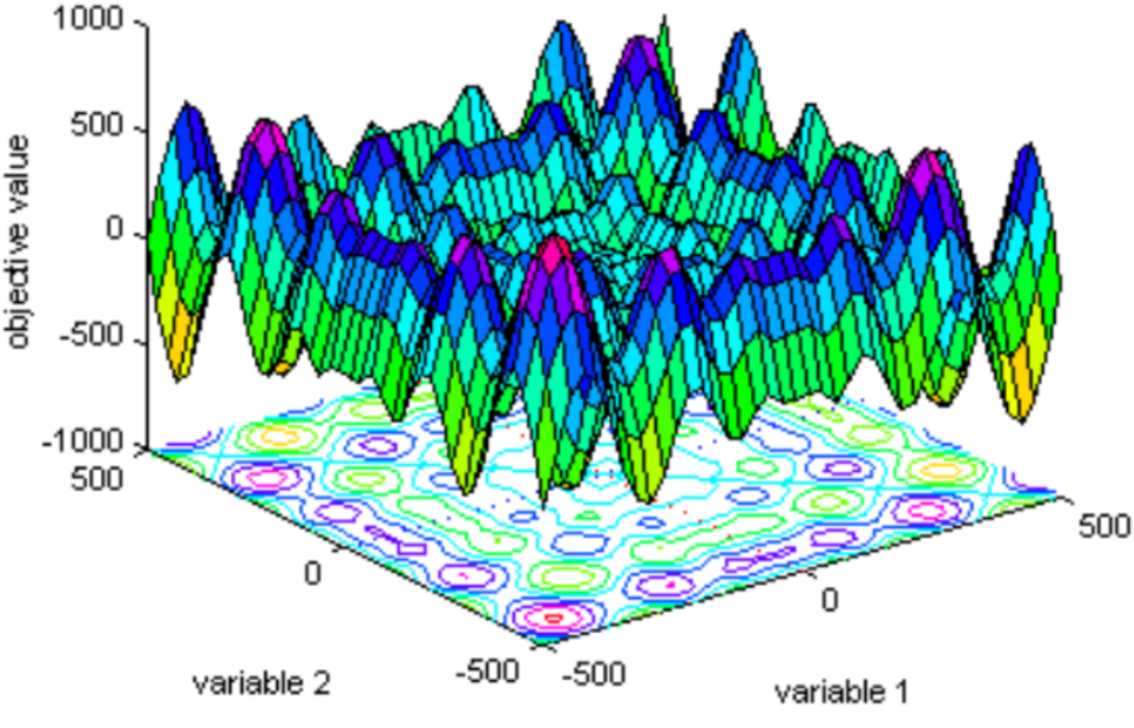

Figure 1: Schwefel's function 7's two-dimensional form for n=2

3.2 Rastrigin's Function 6[2]:

This test function is highly multimodal. However, the location of the minima are regularly distributed.

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

Global minimum:
$f(x) = 0; x_i = 0, \forall n \in N$

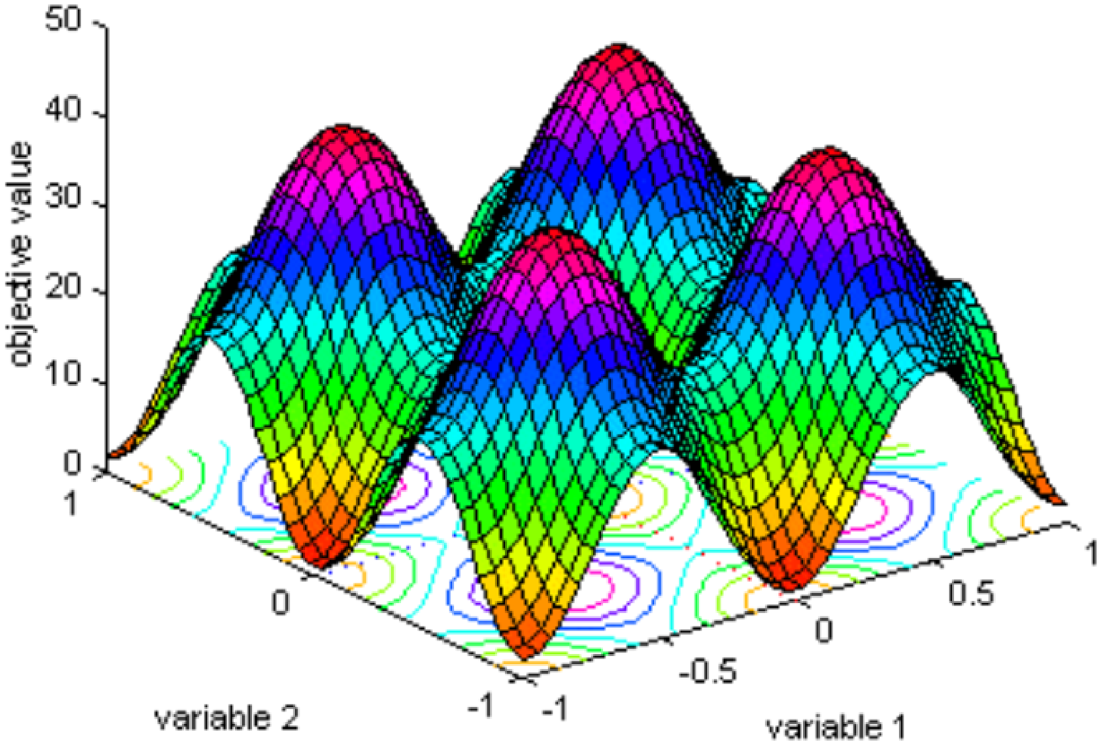

Figure 2: Rastrigin's Function 6's two-dimensional form for n=2

3.3 Dejong's Function 1[4]:

So known as sphere model. It is continuos, convex and unimodal.

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

Global minimum:
$f(x) = 0, x_i = 0, i \in [1, n], \forall n \in N.$

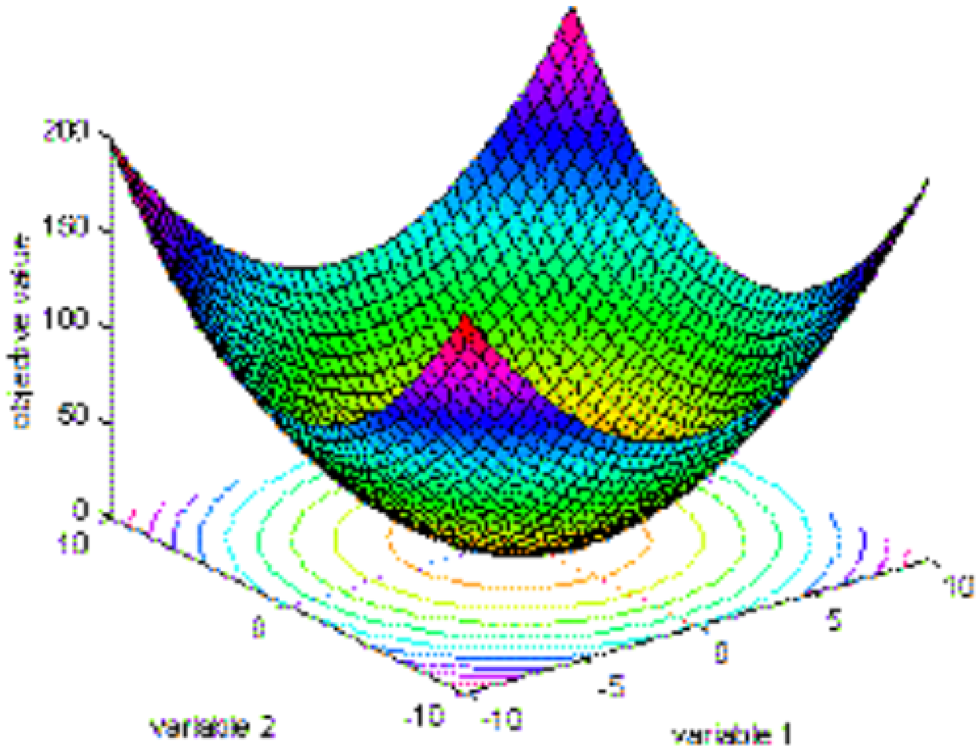

Figure 3: Dejong's function's two-dimensional form for n=2

3.4 Michalewicz's Function[1]:

The Michalewicz function has d! local minima, and it is multimodal. The parameter m defines the steepness of they valleys and ridges. The recommended value of m is m = 10.

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

Global minimum:
at n = 5: $f(x) = -4.6876$,
at n = 10: $f(x) = -9.6601$.
at n = 30: $f(x) \approx -28.24$.

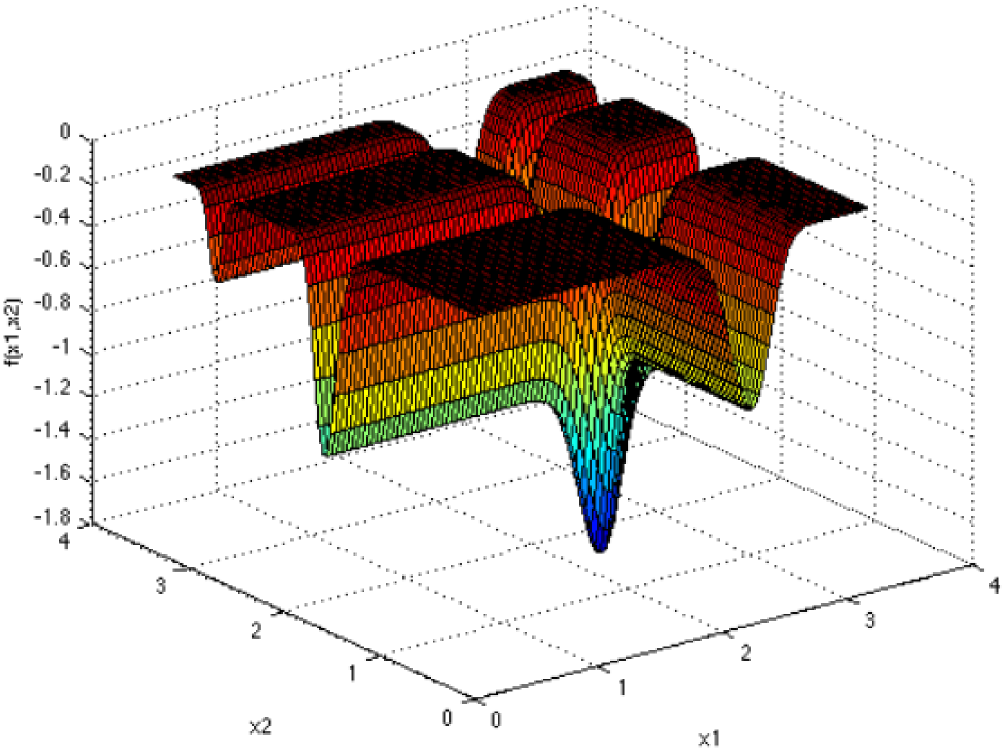

Figure 4: Michalewicz's function's two-dimensional form for n=2, m=10

3.5 Results

3.5.1 Rastrigin's Function:

n	T-min(s)	T-max(s)	T-med(s)	Wrst res	Bst res	Avg res	Std dev
5	71.78	76.282	74.095	11.2	2.097	6.405	2.592
10	240.56	249.313	244.394	30.583	10.237	19.161	5.406
30	1825.487	1861.259	1847.734	80.453	36.883	60.905	11.254

Figure 5: Results for Rastrigin's function - Hill Climbing Best Improvement

n	T-min(s)	T-max(s)	T-med(s)	Wrst res	Bst res	Avg res	Std dev
5	42.976	48.519	46.517	12.29	0.099	6.162	2.853
10	146.369	180.565	161.681	38.592	12.595	25.583	7.605
30	1093.541	1185.78	1129.22	104.813	48.977	75.026	13.978

Figure 6: Results for Rastrigin's function - Hill Climbing First Improvement

n	T-min(s)	T-max(s)	T-med(s)	Wrst res	Bst res	Avg res	Std dev
5	69.349	84.569	74.108	8.216	1.079	3.33	1.618
10	103.654	109.619	106.118	21.154	2.645	7.78	3.717
30	248.273	255.544	251.302	47.304	17.052	28.976	6.804

Figure 7: Results for Rastrigin's function - Simulated Annealing

3.5.2 Michalewicz's function:

n	T-min(s)	T-max(s)	T-med(s)	Wrst res	Bst res	Avg res	Std dev
5	68.052	70.028	69.15	-3.691	-4.674	-4.256	0.27
10	218.732	223.204	220.933	-6.554	-8.751	-7.584	0.478
30	1640.926	1657.942	1648.905	-21.006	-24.788	-22.953	0.937

Figure 8: Results for Michalewicz's function - Hill Climbing Best Improvement

n	T-min(s)	T-max(s)	T-med(s)	Wrst res	Bst res	Avg res	Std dev
5	40.046	46.411	42.688	-3.616	-4.632	-4.295	0.268
10	131.109	138.428	134.567	-5.546	-9.053	-7.273	0.809
30	968.738	993.888	980.674	-18.144	-23.798	-20.896	1.479

Figure 9: Results for Michalewicz's function - Hill Climbing First Improvement

n	T-min(s)	T-max(s)	T-med(s)	Wrst res	Bst res	Avg res	Std dev
5	68.871	74.042	71.109	-4.279	-4.685	-4.561	0.111
10	105.307	112.336	107.921	-8.554	-9.593	-9.119	0.29
30	245.439	256.733	250.09	-26.306	-28.24	-27.236	0.446

Figure 10: Results for Michalewicz's function - Simulated Annealing

3.5.3 Dejong's function:

n	T-min(s)	T-max(s)	T-med(s)	Wrst res	Bst res	Avg res	Std dev
5	70.547	72.195	71.374	0.0	0.0	0.001	0.0
10	235.284	238.309	236.608	0.001	0.001	0.001	0.0
30	1834.855	1857.042	1843.965	0.003	0.003	0.003	0.0

Figure 11: Results for Dejong's function - Hill Climbing Best Improvement

n	T-min(s)	T-max(s)	T-med(s)	Wrst res	Bst res	Avg res	Std dev
5	41.573	46.613	43.815	0.0	0.0	0.001	0.0
10	139.724	147.902	142.971	0.001	0.001	0.001	0.0
30	1079.307	1099.366	1087.239	0.003	0.003	0.003	0.0

Figure 12: Results for Dejong's function - Hill Climbing First Improvement

n	T-min(s)	T-max(s)	T-med(s)	Wrst res	Bst res	Avg res	Std dev
5	64.152	68.153	65.599	0.0	0.0	0.001	0.0
10	100.039	107.563	102.812	0.001	0.001	0.001	0.0
30	243.677	253.625	247.973	0.003	0.003	0.003	0.0

Figure 13: Results for Dejong's function - Simulated Annealing

3.5.4 Schwefel's function:

n	T-min(s)	T-max(s)	T-med(s)	Wrst res	Bst res	Avg res	Std dev
5	131.843	134.865	133.6	-1789.479	-2094.705	-1958.314	84.478
10	446.404	452.49	448.839	-2779.399	-3927.321	-3306.999	278.089
30	3541.236	3588.811	3566.44	-8337.504	-10697.619	-9912.658	500.548

Figure 14: Results for Schwefel's function - Hill Climbing Best Improvement

n	T-min(s)	T-max(s)	T-med(s)	Wrst res	Bst res	Avg res	Std dev
5	76.333	81.456	78.909	-1739.378	-2094.704	-1933.545	96.778
10	254.859	266.192	259.295	-2723.476	-3603.993	-3195.169	249.982
30	1977.767	2015.49	1990.104	-8391.742	-10392.161	-9386.266	466.654

Figure 15: Results for Schwefel's function - Hill Climbing First Improvement

n	T-min(s)	T-max(s)	T-med(s)	Wrst res	Bst res	Avg res	Std dev
5	69.045	74.717	70.55	-1976.051	-2094.914	-2080.579	35.788
10	109.249	114.13	111.522	-4024.298	-4189.617	-4140.928	52.48
30	265.903	277.646	271.854	-11860.798	-12499.045	-12242.26	158.573

Figure 16: Results for Schwefel's function - Simulated Annealing

4 Conclusions

After fine tunes to the cooling schedule, we can clearly observe in the results section that the Simulated Annealing algorithm provided better results overall when tested on highly multimodal functions (Rastrigin's, Michalewicz's and Schwefel's) despite the fact that it's medium execution time is faster than the one of the Hill Climbing approaches. Hill Climbing - Best Improvement produced better results when compared to the First Improvement procedure at the expense of execution time. When used on one-local minimum functions (Dejong), every technique reached and returned the global minimum value of those functions in every sample, Simulated Annealing stratagem being the fastest whilst Hill Climbing - Best Improvement being the slowest of the three.

Unfortunately the research time for the appropriate cooling schedule and cooling schedule parameters of the Simulated Annealing algorithm is almost on-pair with the execution time of both the Hill Climbing methods presented in this report.

References

[1] Michalewicz `https://www.sfu.ca/~ssurjano/michal.html`

[2] Rastrigin `http://www.geatbx.com/docu/fcnindex-01.html#P140_6155`

[3] Schwefel `http://www.geatbx.com/docu/fcnindex-01.html#P150_6749`

[4] Dejong 1 `http://www.geatbx.com/docu/fcnindex-01.html#P89_3085`

[5] Wikipedia - Simulated annealing `https://en.wikipedia.org/wiki/Simulated_annealing#Pseudocode`

[6] Wikipedia - Hill climbing `https://en.wikipedia.org/wiki/Hill_climbing`

[7] Eugen Croitoru - GA `https://profs.info.uaic.ro/~eugennc/teaching/ga/`