Finding the minimum of optimization benchmarking functions using Hill Climbing and Simulated Annealing methods.

University "Alexandru Ioan Cuza" of Iași

Ion Zmeu E4

October 31 2022

1 Abstract

In this article are described the results of approximation the minimum of the Rastrigin,Michalewicz,De Jong 1,Schwefel functions using the Hill Climbing and Simulated Annealing methods by running the algorithms with 1000 iterations and with dimensions 5,10 and 30.

2 Introduction

Finding the minimum of a function is hard and takes a lot of time when done manually that is why in this article I used the Hill Climbing and Simulated Annealing algorithms to compute or approximate the global minimum of the Rastrigin,Michalewicz,De Jong ,Schwefel functions in a reasonable amount of time with decent results by monopolizing the computation power of a computer;

3 Methods

3.1 Solution Representation

The solution is represented as a vector of bites of size B*D where B is the number of bites needed to represent our solution and D is the number of dimensions set.This vector is translated into floating point numbers with a precision of 5.

3.2 Neighbor

I am flipping one bit from current solution to get a neighbor at distance hamming of 1;

3.2.1 First Improvement

This improvement chooses a neighbor from the current solution and replaces the current solution if it is better than it .

3.2.2 Best Improvement

This improvement checks all the neighbors and replaces the current solution with the best neighbor which has the best improvement.

3.2.3 Worst Improvement

This improvement checks a number of random neighbors and replaces the current solution with the best neighbor (which may not be the best solution but it lets the algorithm to check more potentially better solutions).

3.3 Algorithms

3.3.1 Hill Climbing

This algorithm starts with an arbitrary solution to a problem, then attempts to find a better solution by making an incremental change to the solution.

3.3.2 Simulated Annealing

This algorithm is a metaheuristic to approximate global optimization in a large search space for an optimization problem it uses temperature to make a change to the solution which can be decremental but this way it explores a larger space.

3.4 Stop condition

3.4.1 Hill Climbing

The Hill Climbing algorithm stops after going through all the iterations set.

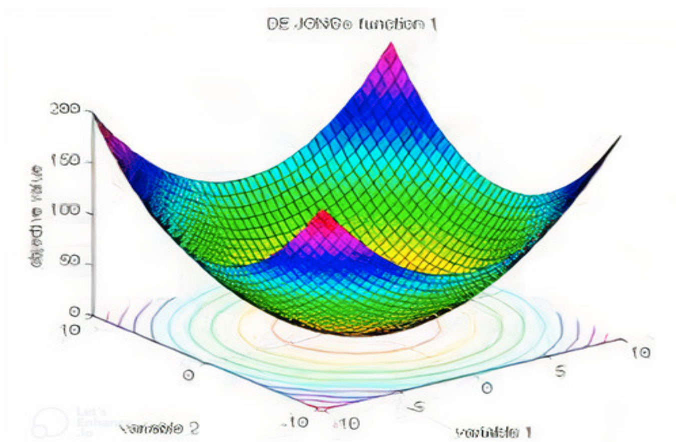
3.4.2 Simulated Annealing

The temperature decreases and The Simulated Annealing algorithm stops after its temperature reaches a low enough value.

4 Functions

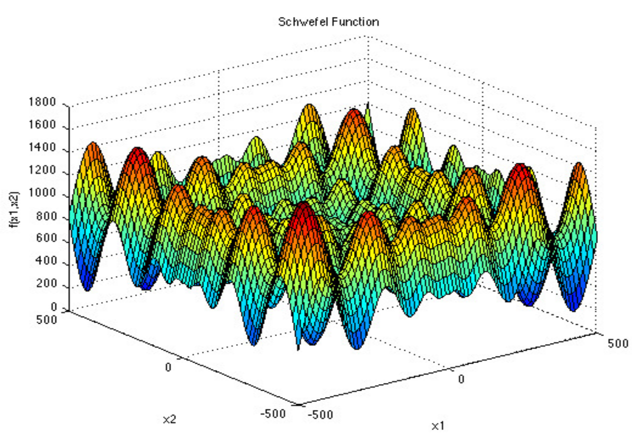
4.1 De Jong's function

$$f_1(x) = \sum_{i=1}^{n} x_i^2, x_i \in [-5.12, 5.12]$$



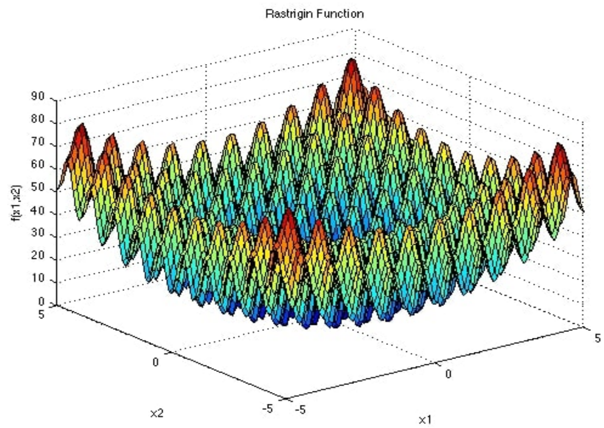
4.2 Schwefel's function

$$f_7(x) = \sum_{i=1}^{n} -x_i \cdot sin(\sqrt[2]{|x_i|}), x_i \in [-500, 500]$$



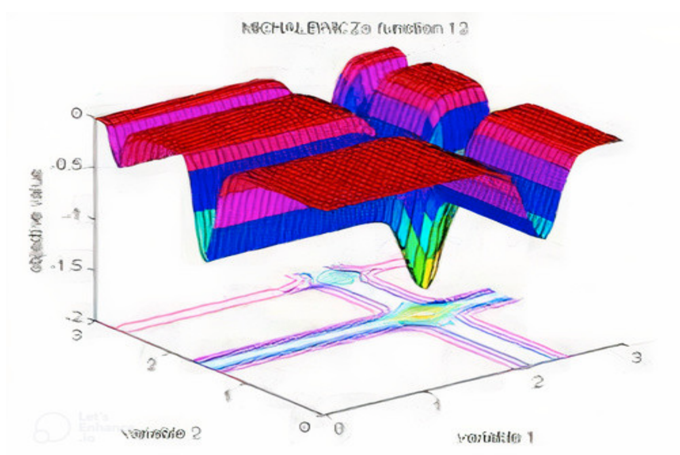
4.3 Rastrigin's function

$$f_6(x) = 10 \cdot n + \sum_{i=1}^{n} x_i^2 - 10 \cdot cos(2 \cdot \pi \cdot x_i), x_i \in [-5.12, 5.12]$$



4.4 Michalewicz's function

$$f_{12}(x) = -\sum_{i=1}^{n} sin(x_i) \cdot \left(sin\left(\frac{i \cdot x_i^2}{\pi}\right)\right)^{2 \cdot m}, x_i \in [0, \pi]$$



5 Results

5.1 De Jong's function

	SA	First	Best	Worst	SA	First	Best	Worst	SA	First	Best	Worst
dimensions	5	5	5	5	10	10	10	10	30	30	30	30
average	0	0	0	0	0	0	0	0	0	0	0	0
best	0	0	0	0	0	0	0	0	0	0	0	0
avg. time	7.23	15.96	10.4	15.93	9.8	55.9	34.46	56.86	19.33	350.53	629.83	635.8
deviation	0	0	0	0	0	0	0	0	0	0	0	0

5.2 Schwefel's function

	SA	First	Best	Worst	SA	First	Best	Worst	SA	First	Best	Worst
dimensions	5	5	5	5	10	10	10	10	30	30	30	30
average	-2017.06	-2071.02	-2093.9	-2094.78	-4018.37	-3910.2	-4082.39	-4058.55	-11588.03418	-10760.51667	-11312.33021	-11291.15928
best	-2094.91	-2094.91	-2094.91	-2094.91	-4189.61	-4128.29	-4189.21	-4155.28	-12097.9	-10937.8	-11873.2	-11604.5
avg. time	5.1	14.03	18.83	19.5	6.63	49.4	71.66	74.33	12.63	586.23	852.16	873.16
deviation	78.43734	21.84317	4.84271	0.09603	147.09506	80.13129	50.38330	59.25302	322.24088	109.32534	152.28449	151.73394

5.3 Rastrigin's function

	SA	First	Best	Worst	SA	First	Best	Worst	SA	First	Best	Worst
dimensions	5	5	5	5	10	10	10	10	30	30	30	30
average	3.94587	0.89387	0.23232	0.36499	10.22801	5.73265	3.90917	4.44261	34.54481	36.60717	28.043	27.94346
best	0.00006	0	0	0	4.22076	2.47168	1.99499	2.23587	23.30365	31.21634	18.1026	21.37
avg. time	5.56	9.9	14.13	9.7	6.93	35.53	52.76	53.8	13.1	489.23	710.83	717.36
deviation	3.01231	0.41261	0.42112	0.47968	3.97924	1.22616	0.80615	0.84973	6.73113	2.39853	2.63904	2.69739

5.4 Michalewicz's function

	SA	First	Best	Worst	SA	First	Best	Worst	SA	First	Best	Worst
dimensions	5	5	5	5	10	10	10	10	30	30	30	30
average	-4.53503	-4.68508	-4.686407	-4.68572	-9.03964	-9.30231	-9.40138	-9.329308	-27.20538	-26.20239	-26.963764	-26.88795
best	-4.68764	-4.68764	-4.68766	-4.68766	-9.51256	-9.55196	-9.65427	-9.4484	-28.26675	-26.75873	-27.59027	-27.83535
avg. time	4.46	7.73	12.33	12.83	6.96	32.33	52.03	53.46	14.6	420.16	697.13	703.33
deviation	0.12561	0.00489	0.00102	0.00418	0.27974	0.11747	0.10284	0.07114	0.58748	0.27563	0.00010	0.33875

6 Conclusion

From this data we can deduce that the Hill Climbing algorithm is better when searching smaller spaces with a smaller local minimum and the Simulated Annealing algorithm produces good results when searching a large space , both of them are good for computing good results in reasonable time.

7 References

[1]https://en.wikipedia.org/wiki/Simulated_annealing
[2]https://en.wikipedia.org/wiki/Hill_climbing
[3]http://www.geatbx.com/docu/fcnindex-01.html#P140_6155
[4]https://profs.info.uaic.ro/~eugennc/teaching/ga/
[5]https://www.calculator.net/standard-deviation-calculator.html
[6]https://www.youtube.com/watch?v=_ThdI0A9Lbk
[7]https://www.youtube.com/watch?v=XNMGq5Jjs5w