Genetic Algorithms in Minimization

Georgiana Diana Ojoc

November 26, 2019

Abstract

Genetic Algorithms have become popular as a means of solving hard combinatorial optimization problems. The first part of this paper briefly traces their history, explains the basic concepts and discusses a detailed implementation. The second part concentrates on the comparison with the three trajectory methods studied in the previous paper, namely Next Ascent Hill Climbing, Steepest Ascent Hill Climbing and Simulated Annealing, using four benchmark functions with variable number of arguments, namely Ackley, Michalewics, Rastrigin and Schwefel, in order to establish the best approach of a minimization problem.

1 Introduction

Genetic Algorithms are search and optimization algorithms based on the principles of natural evolution, which were first introduced by John Holland in 1970 and based on the concept of Charles Darwin's theory of evolution. Genetic Algorithms also implement the optimization strategies by simulating evolution of species through natural selections.

- solution - best candidate in the last generation
- generation - one iteration of the evolutionary process
- population - group of all individuals
- chromosome (individual) - a sequence of genes
- gene - the atomic information in a chromosome
- locus - the position of a gene on a chromosome
- allele - all possible values of a locus
- mutation - elementary random change of a gene
- crossover - exchange of genetic information between parents
- selection - random process of choosing parents or survivors

The input to the Genetic Algorithm is a set of potential solutions to that problem and a metric called a fitness function that allows each candidate to be quantitatively evaluated. These candidates are generated at random very often. Genetic Algorithms are generally composed of two processes. The first process is represented by the manipulation of individuals from the current generation by Mutation and Crossover techniques and the second process consists of the selection of individuals for the production of the next generation. The selection mechanism evaluates each candidate according to the fitness function activity, in a way that a candidate with a good fitness has higher chances to get selected than one with an average fitness.

2 Motivation

Genetic Algorithms have proven to be a successful problem-solving strategy, because they reflect the process of natural selection, which is present in all biological systems, the latter being considered the most highly optimized, adaptive systems known. As opposed to Hill Climbing and Simulated Annealing, which move in the space of a string by a single bit mutation from an original string, each new information sample is completely independent of the previous one.

3 Method

Mutation

In order to ensure that the individuals are not exactly the same, loop through all the alleles of all the individuals and replace the values of the selected alleles for mutation. I chose to represent the chromosomes as bit strings and to use a probability of Mutation that equals **0.01%**.

Crossover

I chose the Single Point Crossover and a probability of Crossover that equals **30%**. This technique selects randomly a locus at which it swaps the remaining alleles from one parent to another.

Selection

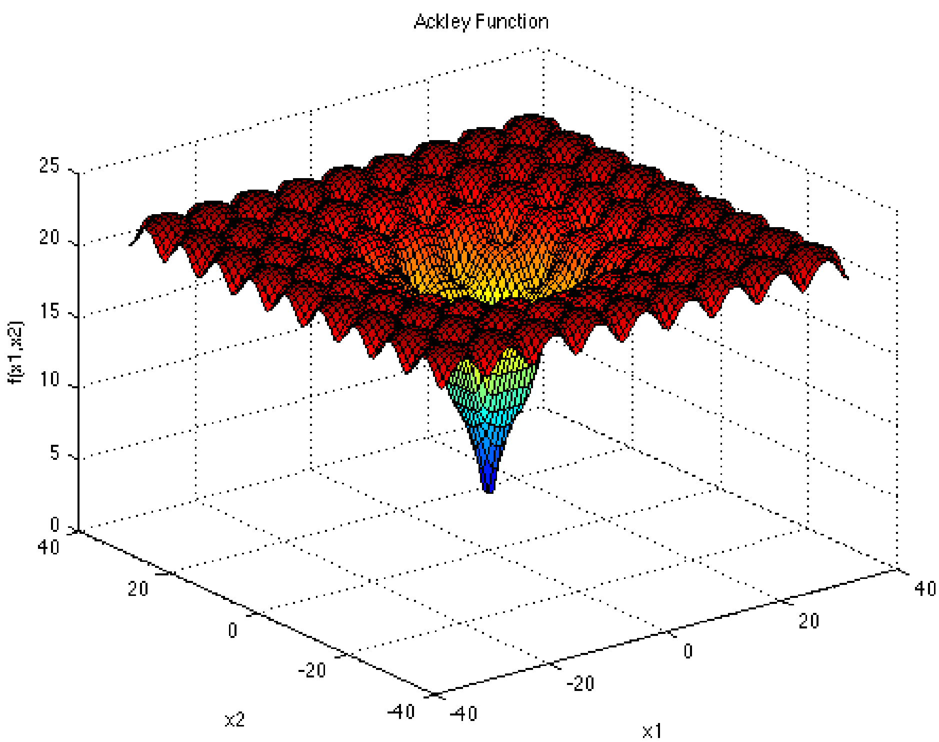
I chose a fitness function that is represented by $fitness(individual) = 1.1 * max(fitness) - fitness(individual)$ and the Roulette Wheel technique, in which the chance of an individual to be selected is proportional to the amount by which its fitness is greater or less than its competitors' fitness. I also tried Elitist and Tournament techniques, but Roulette Wheel provides the best results.

4 Results

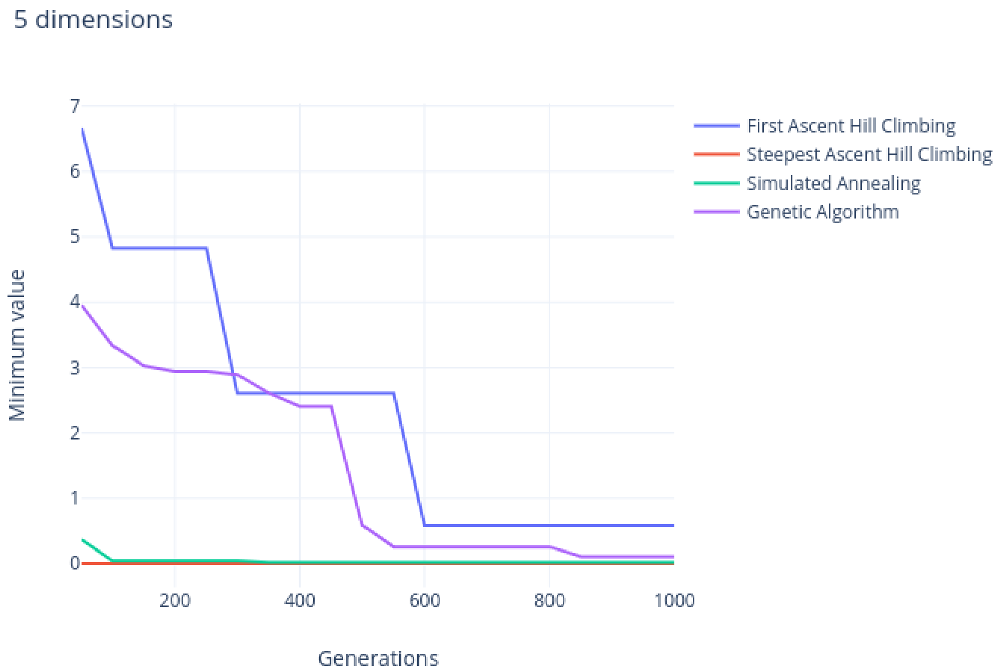
Ackley Function

Ackley's function is a widely used multimodal test function.

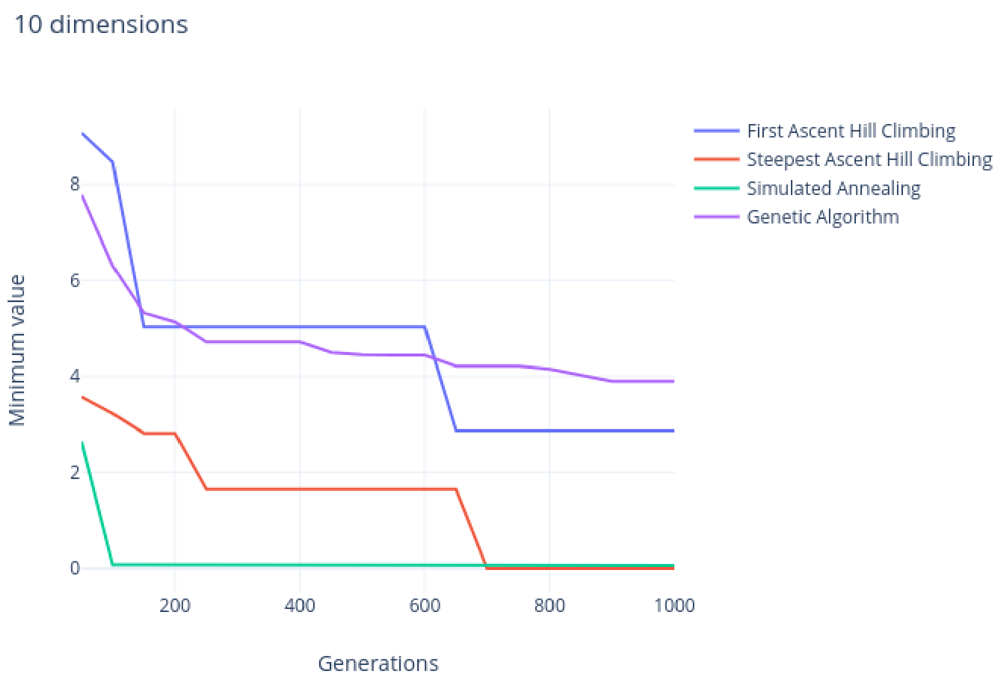
- Number of variables: n variables
- Definition: $f(\mathbf{x}) = f(x_1, ..., x_n) = 20 + e - 20e^{-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}} - e^{\frac{1}{n}\sum_{i=1}^{n} cos(2\pi x_i)}$
- Search domain: $-15 \leqslant x_i \leqslant 30,\ i = 1, 2, ..., n$
- Global minimum: $x^* = (0, 0, ..., 0),\ f(x^*) = 0$
- Function graph: for $n = 2$



5 dimensions *Minimum value*	**Next Ascent Hill Climbing**	**Steepest Ascent Hill Climbing**	**Simulated Annealing**	**Genetic Algorithm**
Lowest value	0.271644	0.000000	0.006228	0.105677
Highest value	3.259373	1.646224	0.031405	3.528669
Mean	1.753228	0.102889	0.016966	2.462089
Median	2.122435	0.000000	0.016935	2.585520
Standard deviation	1.076154	0.404864	0.006094	1.027170



10 dimensions *Minimum value*	**Next Ascent Hill Climbing**	**Steepest Ascent Hill Climbing**	**Simulated Annealing**	**Genetic Algorithm**
Lowest value	3.326332	0.000000	0.028927	3.896640
Highest value	6.775652	2.814350	0.409925	8.767346
Mean	4.814584	1.630285	0.097009	7.203360
Median	4.980963	1.646224	0.082380	7.950652
Standard deviation	0.841747	0.775751	0.070873	1.656964

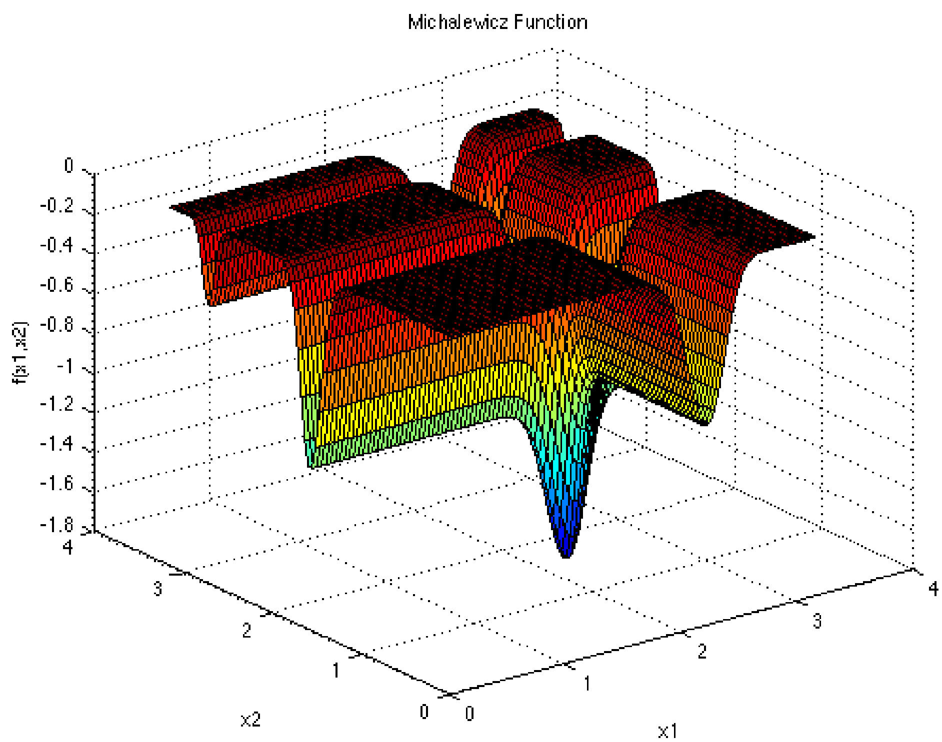


30 dimensions *Minimum value*	**Next Ascent Hill Climbing**	**Steepest Ascent Hill Climbing**	**Simulated Annealing**	**Genetic Algorithm**
Lowest value	5.678198	2.738609	2.504724	8.251607
Highest value	9.314037	3.682008	4.958828	9.885166
Mean	8.045001	3.114515	3.831233	9.059375
Median	8.290626	3.026937	3.671377	9.055507
Standard deviation	0.975008	0.202957	0.577180	0.474584

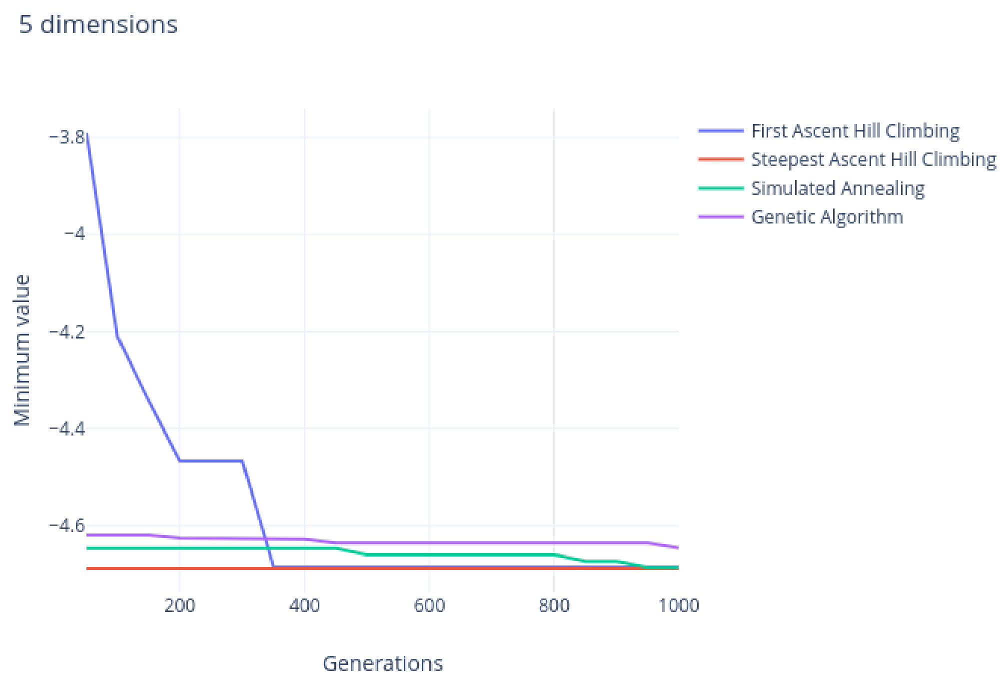
Michalewics Function

Michalewics' function is a multimodal test function. The exponent of the second sinus defines the "steepness" of the valleys or edges. Larger exponent leads to more difficult search. For very large exponent 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).

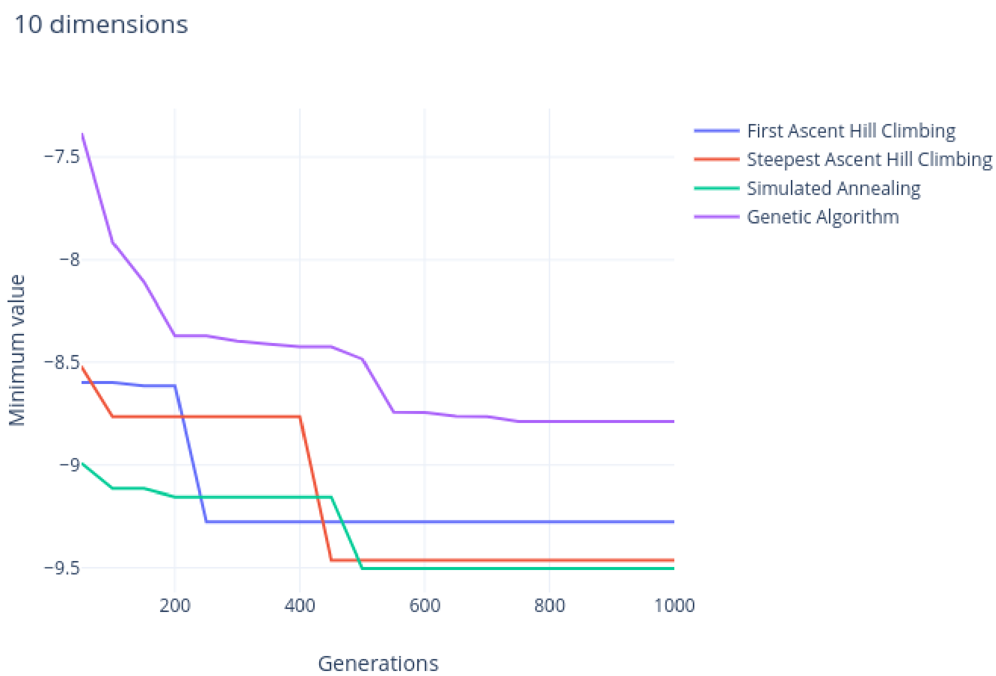
- Number of variables: n variables
- Definition: $f(\mathbf{x}) = f(x_1, ..., x_n) = -\sum_{i=1}^{n} \sin(x_i) \sin^{20}(\frac{ix_i^2}{\pi})$
- Search domain: $0 \leqslant x_i \leqslant \pi$, $i = 1, 2, ..., n$
- Global minimum: $n = 5 : f(x^*) = -4.687658$, $n = 10 : f(x^*) = -9.660150$
- Function graph: for $n = 2$



5 dimensions *Minimum value*	**Next Ascent Hill Climbing**	**Steepest Ascent Hill Climbing**	**Simulated Annealing**	**Genetic Algorithm**
Lowest value	-4.687506	-4.687658	-4.683960	-4.645770
Highest value	-4.516009	-4.645895	-4.640778	-4.352910
Mean	-4.644585	-4.682438	-4.670241	-4.522150
Median	-4.667276	-4.687658	-4.675162	-4.520190
Standard deviation	0.047689	0.014033	0.012620	0.111128



10 dimensions *Minimum value*	**Next Ascent Hill Climbing**	**Steepest Ascent Hill Climbing**	**Simulated Annealing**	**Genetic Algorithm**
Lowest value	-9.343045	-9.577276	-9.540572	-8.877780
Highest value	-8.685749	-9.015225	-9.231948	-8.036360
Mean	-9.017809	-9.239426	-9.394646	-8.574890
Median	-9.024617	-9.231762	-9.398126	-8.565210
Standard deviation	0.188814	0.147499	0.071194	0.262389

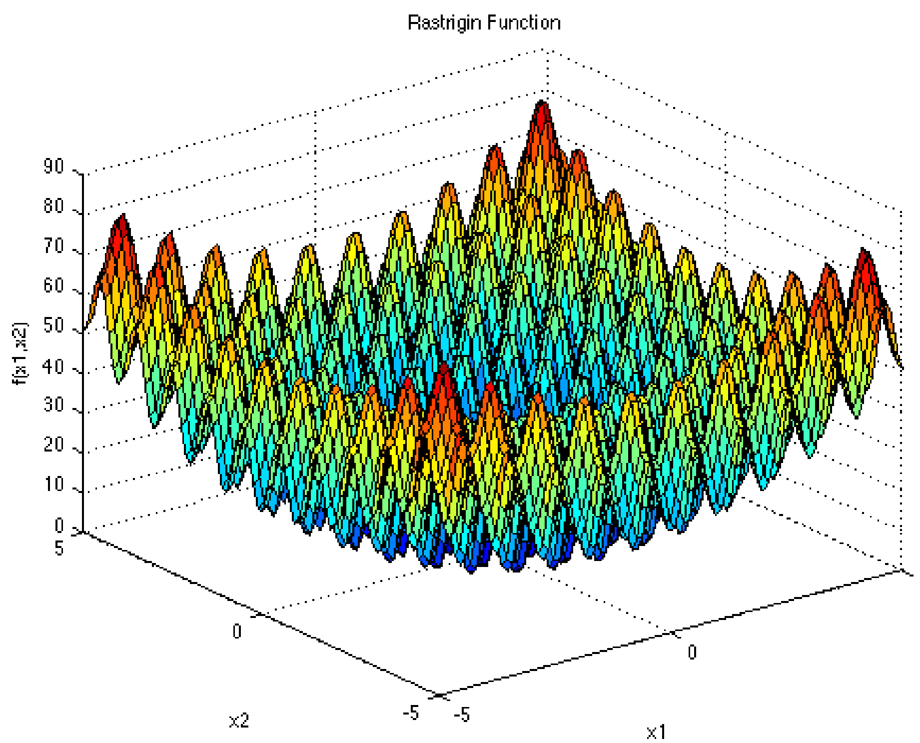


30 dimensions Minimum value	**Next Ascent Hill Climbing**	**Steepest Ascent Hill Climbing**	**Simulated Annealing**	**Genetic Algorithm**
Lowest value	-26.907732	-27.205175	-28.470696	-22.218600
Highest value	-24.848757	-25.629206	-27.711190	-20.042100
Mean	-25.615436	-26.468908	-27.992412	-21.483600
Median	-25.585226	-26.524710	-27.984600	-21.716700
Standard deviation	0.485646	0.438445	0.190435	0.645084

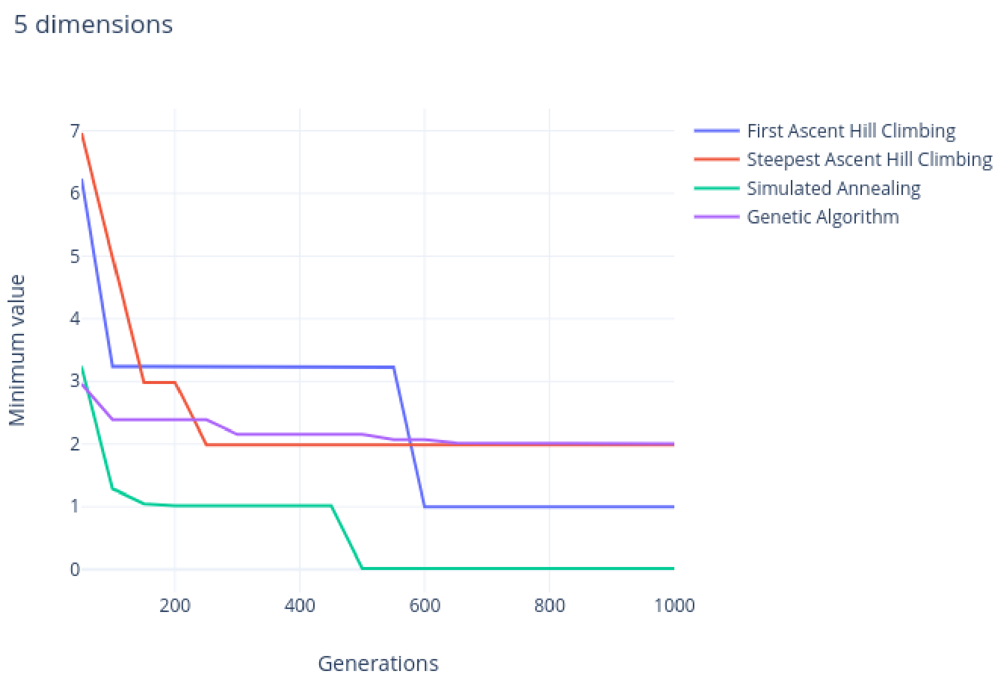
Rastrigin Function

Rastrigin's function produces many local minima. Thus, the test function is highly multimodal. However, the location of the minima are regularly distributed.

- Number of variables: n variables
- Definition: $f(\mathbf{x}) = f(x_1, ..., x_n) = 10n + \sum_{i=1}^{n}(x_i^2 - 10cos(2\pi x_i))$
- Search domain: $-5.12 \leqslant x_i \leqslant 5.12,\ i = 1, 2, ..., n$
- Global minimum: $x^* = (0, 0, ..., 0),\ f(x^*) = 0$
- Function graph: for $n = 2$

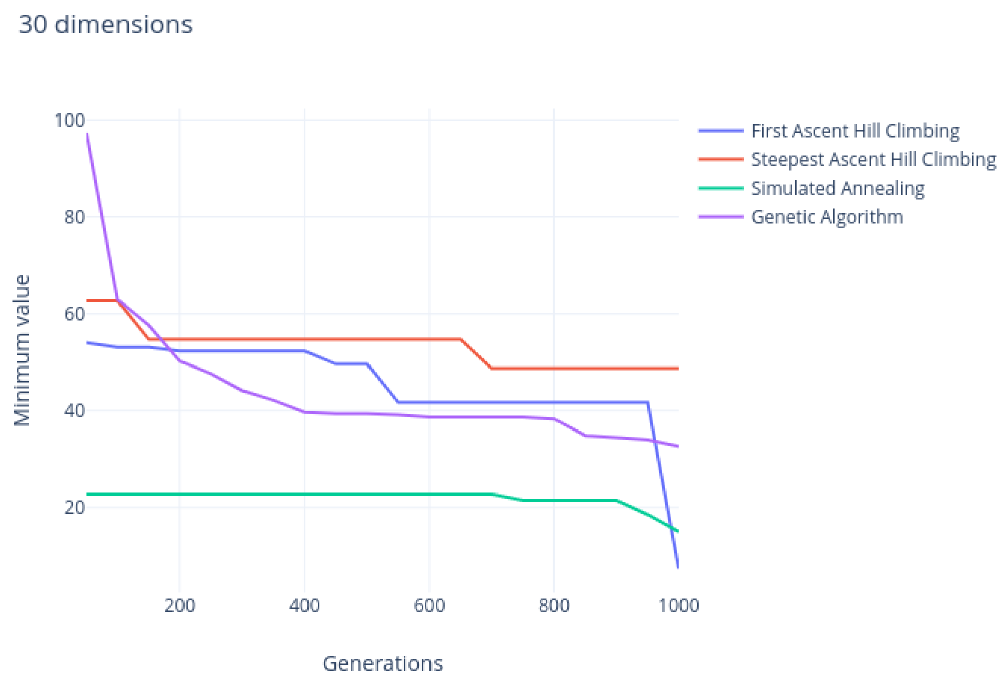


5 dimensions *Minimum value*	**Next Ascent Hill Climbing**	**Steepest Ascent Hill Climbing**	**Simulated Annealing**	**Genetic Algorithm**
Lowest value	0.994959	0.994959	0.005161	2.004868
Highest value	3.230728	3.979836	1.271012	3.3166680
Mean	2.443212	2.300843	0.460844	2.539986
Median	2.235768	1.989918	0.041462	2.311426
Standard deviation	0.486063	0.641207	0.511944	0.476530



10 dimensions Minimum value	**Next Ascent Hill Climbing**	**Steepest Ascent Hill Climbing**	**Simulated Annealing**	**Genetic Algorithm**
Lowest value	8.461455	6.964708	1.047186	5.152758
Highest value	12.369662	11.939504	4.995281	8.671650
Mean	11.156503	9.141182	2.992019	7.139389
Median	11.907784	8.954626	3.167510	7.617865
Standard deviation	1.179549	1.483046	0.878797	1.447348

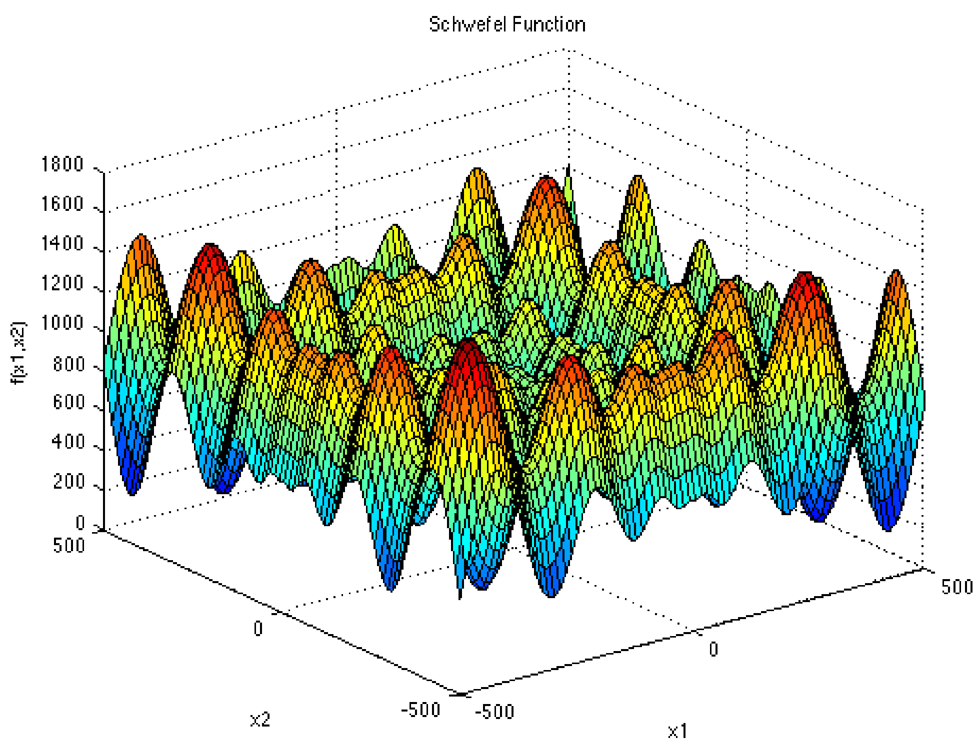
30 dimensions *Minimum value*	**Next Ascent Hill Climbing**	**Steepest Ascent Hill Climbing**	**Simulated Annealing**	**Genetic Algorithm**
Lowest value	34.528340	35.818481	12.940115	32.608660
Highest value	43.143851	49.747852	22.568770	47.659110
Mean	38.320229	42.378966	17.798400	39.260720
Median	38.496617	41.788210	17.809668	37.573380
Standard deviation	2.328852	4.484311	2.402766	5.696160



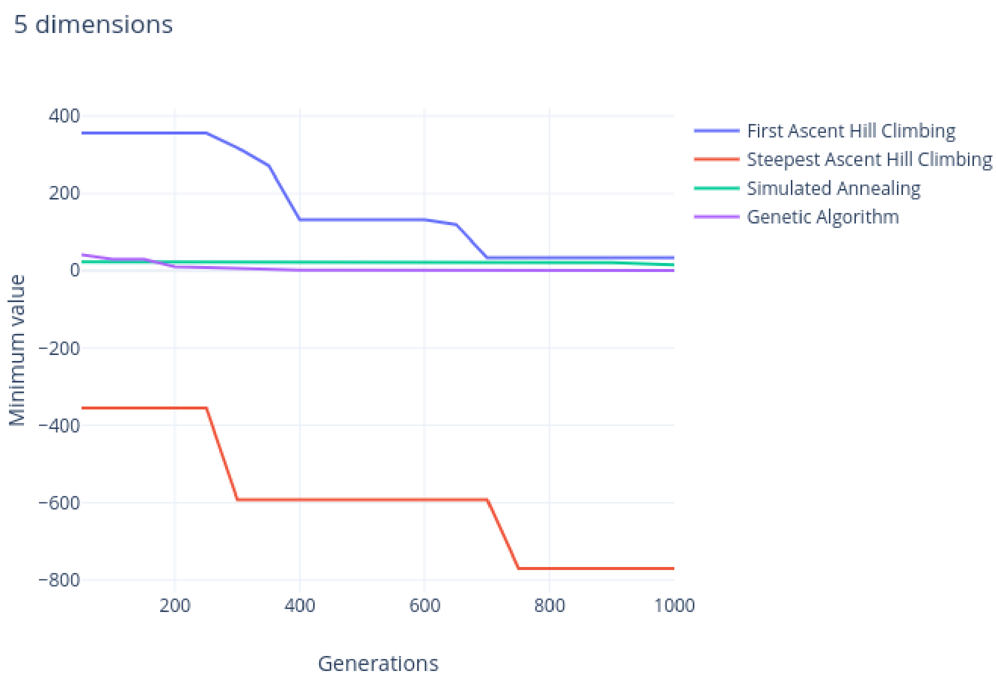
Schwefel Function

Schwefel's function is deceptive in that the global minimum is geometrically distant, over the parameter space, from the next best local minimum. Therefore, the search algorithms are potentially prone to convergence in the wrong direction.

- Number of variables: n variables
- Definition: $f(\mathbf{x}) = f(x_1, ..., x_n) = 418.9829n - \sum_{i=1}^{n} x_i \sin(\sqrt{|x_i|})$
- Search domain: $-500 \leqslant x_i \leqslant 500,\ i = 1, 2, ..., n$
- Global minimum: $x^* = (420.9687, 420.9687, ..., 420.9687),\ f(x^*) = 0$
- Function graph: for $n = 2$

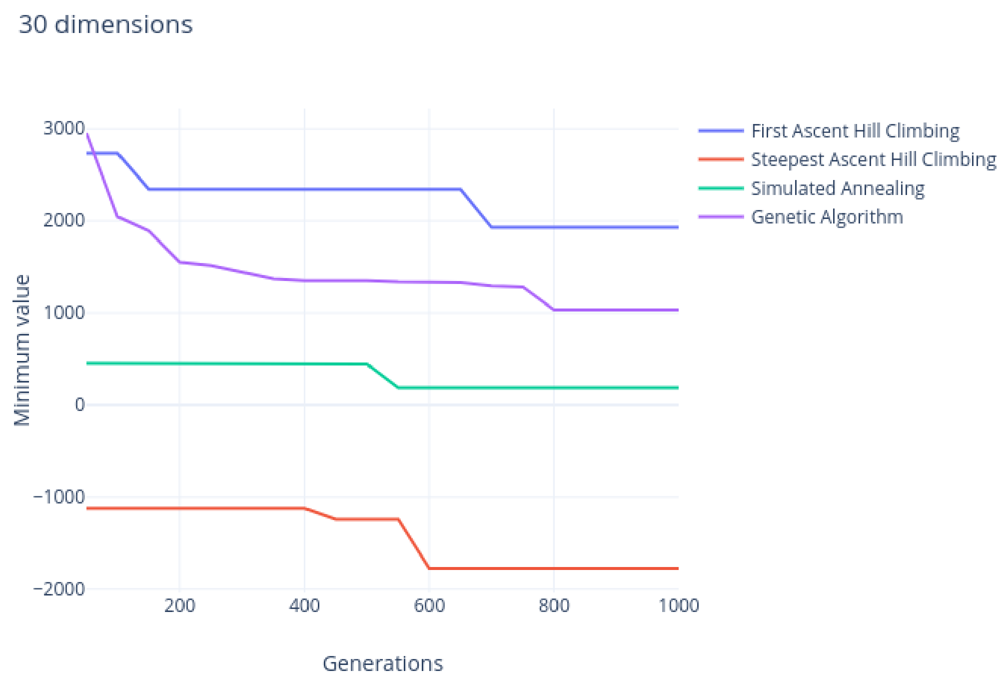


5 dimensions *Minimum value*	**Next Ascent Hill Climbing**	**Steepest Ascent Hill Climbing**	**Simulated Annealing**	**Genetic Algorithm**
Lowest value	0.104988	-2546.373939	0.005364	0.280626
Highest value	187.019248	-769.835768	118.457059	172.860100
Mean	113.511944	-1534.111869	3.803725	69.210550
Median	124.466178	-1539.646112	0.119324	51.890990
Standard deviation	50.981583	460.507656	20.922003	67.686830



10 dimensions *Minimum value*	**Next Ascent Hill Climbing**	**Steepest Ascent Hill Climbing**	**Simulated Annealing**	**Genetic Algorithm**
Lowest value	318.245493	-2191.058871	0.139940	154.773500
Highest value	590.017889	-1065.927094	68.828602	354.522100
Mean	450.352956	-1584.881793	6.827592	284.524200
Median	434.069567	-1539.662008	0.462836	306.180100
Standard deviation	88.861298	288.008396	16.083109	61.859350

30 dimensions *Minimum value*	**Next Ascent Hill Climbing**	**Steepest Ascent Hill Climbing**	**Simulated Annealing**	**Genetic Algorithm**
Lowest value	1419.395233	-4145.235784	119.700726	1033.183000
Highest value	2254.105152	-1538.136432	425.293153	1813.277000
Mean	1975.709800	-2926.939982	295.731647	1525.735000
Median	2002.961724	-2842.416569	311.421029	1548.453000
Standard deviation	204.189145	959.004882	80.957382	246.820000



5 Interpretation

Simulated Annealing gives better results than the Genetic Algorithm I chose, but the second method finds the minimum value in less generations than the first two trajectory methods it was compared with.

6 Conclusion

This paper presents the basic functionality of Genetic Algorithms, along with the techniques that provided the best results, and a comparison between a Genetic Algorithm and three trajectory methods, namely Next Ascent Hill Climbing, Steepest Ascent Hill Climbing and Simulated Annealing.

References

- https://www.researchgate.net/publication/309770246_A_Study_on_Genetic_Algorithm_and_its_Applications
- https://www.researchgate.net/publication/268010749_Genetic_Algorithm_-_Survey_Paper
- https://www.sfu.ca/~ssurjano/ackley.html
- https://www.sfu.ca/~ssurjano/michal.html
- https://www.sfu.ca/~ssurjano/rastr.html
- https://www.sfu.ca/~ssurjano/schwef.html