

A Study of Genetic Algorithms in relation to Hill Climbing and Simulated Annealing

Florea Robert-Andrei

1 December 2024

1 Abstract

This paper illustrates the application of Genetic Algorithms (GA) which has the purpose of optimizing the following functions: De Jong 1, Schwefel, Rastrigin, and Michalewicz. These were applied within the 5, 10 and 30 dimensions. The investigation focuses on the contrast between the GA's performance and the variants' known as Hill Climbing and Simulated Annealing. Supported by an iterative process of mutation, crossover, selection and fitness evaluations, GA demonstrates the fact that it is suitable in regard with low-dimensional spaces. However, its performance is significantly decreased in high-dimensional spaces. In this context, through a lense of differentiation, it is showed that its capacity is diminished in comparison to Hill Climbing and Simulated Annealing performance.

2 Introduction

This study focuses on GA's performances on finding the minimum value of the functions mentioned above: De Jong 1, Schwefel's, Rastrigin's, and Michalewicz's. These functions represent the base of the evaluation, showing the efficiency and the workability of GA used on a variety of problems. The starting point is the previous paperwork and using the method of comparison, the differences in performance are highlighted according to the same framework. Therefore, this study presents the strengths and the drawbacks of using these methodologies.

3 Methods

Before diving into the details of the manner in which the algorithm functions, there should be a clear delimitation of what this GA stands for.

Genetic Algorithms are a family of computational models inspired by evolution. These algorithms encode a potential solution to a specific problem on a simple chromosome-like data structure and apply recombination operators to these structures as as to preserve critical information. Genetic algorithms are often viewed as function optimizer, although the range of problems to which genetic algorithms have been applied are quite broad.[1]

An implementation of genetic algorithm begins with a population of (typically random) chromosomes. One then evaluates these structures and allocated reproductive opportunities in such a way that these chromosomes which represent a better solution to the target problem are given more chances to ‘re-produce’ than those chromosomes which are poorer solutions.

The parameters used are the following: the size of the population is 100, the probability of mutation set to 0.001, the probability of crossover set to 0.7 and a maximum generation number of 1000. In terms of precision, the value is 10^{-5} while the performance will run on the 5, 10 and 30 dimension. In order to obtain multiple results the program is ought to run 30 times. According to the values emerged the minimum, maximum, mean and standard deviation will be selected. Firstly, the algorithm will generate a population of 100 dimension, each one being the bool type. Considering the number of dimensions and the function’s interim, each population will be randomly generated using the number of bytes which is required. The algorithm is going to use the GA’s functions: mutation, crossover as well as selection. Besides those, another evaluation function will generate the values of each population based on the chosen function. By applying the mutation step to the whole population at the each generation’s outset, random changes are introduces in the individuals’ genetic makeup. It is mandatory to go through this process in order to inject new genetic characteristics into the population. This way, its diversity and aiding is enhanced within the exploration of new areas in the solution space. This process is done by the action of flipping the bits in the binary strings while setting the probability in the initial phase. By selecting two parents from the population the crossover process starts. Two crossover points are randomly determined by the implemented algorithm and placed in the genetic sequence of the parents. Furthermore, there is a zigzag pattern of genetic exchange which is implemented. In this context, the parent bit strings’ segments are interchanged between the two identified crossover points. This method generates two offspring, each developing a mix of genetic data from both parents. The next phase is ‘selection’. This phase presents how individuals are chosen: there is a combination of elitism strategy and roulette wheel selection. The individuals which fit best and constitute the top 4 ones of the population are directly transfered to the next generation, ensuring that the ones from the current population are also carried over the new one. In the context of the remaining ones within; the population, a probabilistic approach is used to maintain genetic diversity and avoiding premature convergence. By doing that, the approach presents a chance to be selected to each individual according to their fitness. Moreover, a random number is generated for each remaining slot in the new population. Also an individual corresponding to the range in which the random number falls according to cumulative probabilities is selected.

The execution is based on the following mathematical functions:

De Jong Function[5]:

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

Schwefel Function[5]:

$$f(x) = 418.9829 \times n - \sum_{i=1}^n x_i \times \sin(\sqrt{|x_i|}), x_i \in [-500, 500], i = 1..n \quad (2)$$

Rastrigin Function[5]:

$$f(x) = A \times n + \sum_{i=1}^n [x_i^2 - A \times \cos(2\pi x_i)], A = 10, x_i \in [-5.12, 5.12], i = 1..n \quad (3)$$

Michalewicz Function[5]:

$$f(x) = - \sum_{i=1}^n \sin(x_i) \times [\sin(\frac{i \times x_i^2}{\pi})]^{2m}, m = 10, x_i \in [0, \pi], i = 1..n \quad (4)$$

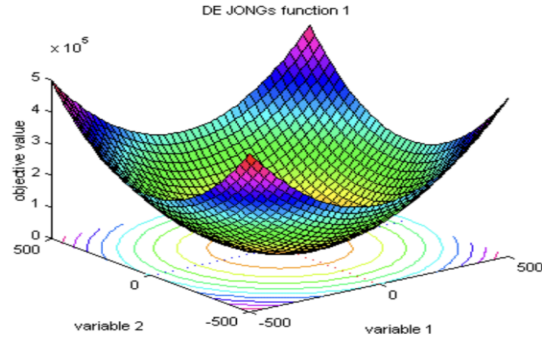


Figure 1: De Jong Function [5]

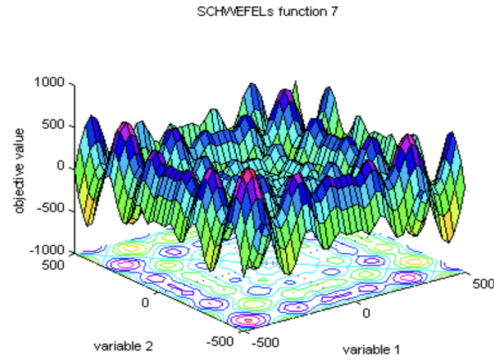


Figure 2: Schwefel Function [5]

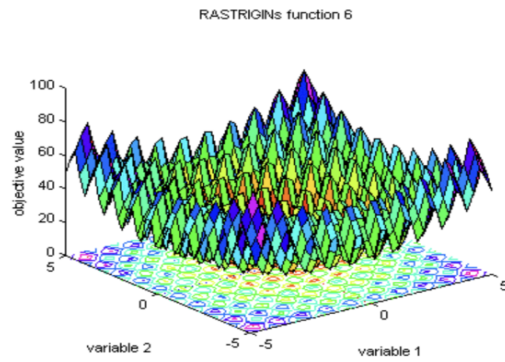


Figure 3: Rastrigin Function [5]

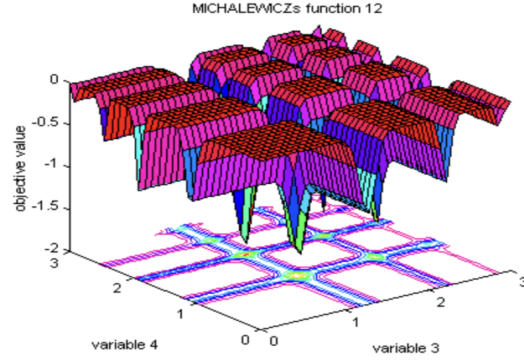


Figure 4: Michalewicz Function [5]

4 Results and comparisons

4.1 De Jong

GA	n=5	n=10	n=30
Min. value	0.00000	0.00000	0.00106
Max. value	0.00000	0.00000	0.01690
Mean	0.00000	0.00000	0.00607
Standard Deviation	0.00000	0.00000	0.00393

n=5

	GA	HC-Best	HC-First	HC-Worst	SA
Min. value	0.00000	0.00000	0.00000	0.31097	0.00000
Max. value	0.00000	0.00000	0.00000	5.46014	0.00000
Mean	0.00000	0.00000	0.00000	2.89522	0.00000
Standard Deviation	0.00000	0.00000	0.00000	1.26908	0.00000

Results for HC-Best, HC-First, HC-Worst and SA [2]

n=10

	GA	HC-Best	HC-First	HC-Worst	SA
Min. value	0.00000	0.00000	0.00000	10.72072	0.00000
Max. value	0.00000	0.00000	0.00000	26.91588	0.00000
Mean	0.00000	0.00000	0.00000	19.88266	0.00000
Standard Deviation	0.00000	0.00000	0.00000	4.67432	0.00000

Results for HC-Best, HC-First, HC-Worst and SA [2]

n=30

	GA	HC-Best	HC-First	HC-Worst	SA
Min. value	0.00106	0.00000	0.00000	111.93527	0.00000
Max. value	0.01690	0.00000	0.00000	156.39117	0.00040
Mean	0.00607	0.00000	0.00000	135.67476	0.00002
Standard Deviation	0.00393	0.00000	0.00000	12.01665	0.00007

Results for HC-Best, HC-First, HC-Worst and SA [2]

4.2 Schwefel

GA	n=5	n=10	n=30
Min. value	0.20866	28.79078	356.26457
Max. value	302.63667	563.99510	1412.73961
Mean	63.51824	203.56509	716.63961
Standard Deviation	54.35799	165.55734	273.23644

n=5

	GA	HC-Best	HC-First	HC-Worst	SA
Min. value	0.20866	0.00011	0.10375	379.19581	0.00069
Max. value	302.63667	0.20865	102.70665	1945.27857	34.44355
Mean	63.51824	0.09165	27.87243	724.84412	1.01453
Standard Deviation	54.35799	0.06684	23.72457	134.63654	5.42187

Results for HC-Best, HC-First, HC-Worst and SA [2]

n=10

	GA	HC-Best	HC-First	HC-Worst	SA
Min. value	28.79078	0.62312	153.29677	1772.89931	0.00201
Max. value	563.99510	248.34030	425.61503	2623.49310	34.96195
Mean	203.56509	107.34842	288.57985	2255.27826	5.45540
Standard Deviation	165.55734	53.76364	66.15256	208.63557	12.41442

Results for HC-Best, HC-First, HC-Worst and SA [2]

n=30

	GA	HC-Best	HC-First	HC-Worst	SA
Min. value	356.26457	888.09205	1313.39166	7881.37057	0.72923
Max. value	1412.73961	1411.22948	2038.94606	9653.71196	119.89815
Mean	716.63961	1226.33757	1762.37164	9005.03426	30.82801
Standard Deviation	273.23644	144.79832	138.93934	449.61408	38.17938

Results for HC-Best, HC-First, HC-Worst and SA [2]

4.3 Rastrigin

GA	n=5	n=10	n=30
Min. value	1.99497	6.93826	31.55983
Max. value	16.91429	31.47747	74.35898
Mean	8.20896	18.90277	53.72954
Standard Deviation	4.29973	6.23960	10.02593

n=5

	GA	HC-Best	HC-First	HC-Worst	SA
Min. value	1.99497	0.00000	0.00000	5.87731	0.00000
Max. value	16.91429	1.23582	2.23078	33.83907	8.14915
Mean	8.20896	0.40413	0.93721	23.72123	2.67206
Standard Deviation	4.29973	0.50265	0.59600	5.72894	1.68146

Results for HC-Best, HC-First, HC-Worst and SA [2]

n=10

	GA	HC-Best	HC-First	HC-Worst	SA
Min. value	6.93826	0.99496	3.22574	58.70354	1.98992
Max. value	31.47747	5.45652	7.93322	101.13403	12.37491
Mean	18.90277	3.61979	5.92097	85.79486	5.03176
Standard Deviation	6.23960	0.97000	1.03978	8.94824	2.55480

Results for HC-Best, HC-First, HC-Worst and SA [2]

n=30

	GA	HC-Best	HC-First	HC-Worst	SA
Min. value	31.55983	18.85637	31.48343	342.51404	11.20057
Max. value	74.35898	32.95540	41.46681	407.72442	28.22147
Mean	53.72954	27.70595	36.37522	382.71412	18.15091
Standard Deviation	10.02593	2.99251	2.57392	11.87121	4.01798

Results for HC-Best, HC-First, HC-Worst and SA [2]

4.4 Michalewicz

GA	n=5	n=10	n=30
Min. value	4.68512	9.20740	27.00472
Max. value	3.22824	7.02734	22.36519
Mean	4.21019	8.34958	25.19241
Standard Deviation	0.57110	6.23960	1.04261

n=5

	GA	HC-Best	HC-First	HC-Worst	SA
Min. value	-4.68512	-4.68766	-4.69765	-3.42603	-4.68593
Max. value	-3.22824	-4.68153	-4.66927	-2.33583	-3.54586
Mean	-4.21019	-4.68681	-4.68372	-2.81155	-4.23580
Standard Deviation	0.31202	0.00140	0.00507	0.25297	0.31140

Results for HC-Best, HC-First, HC-Worst and SA [2]
n=10

	GA	HC-Best	HC-First	HC-Worst	SA
Min. value	-9.20740	-9.57161	-9.53408	-4.56525	-9.24567
Max. value	-7.02734	-9.25546	-9.08584	-3.38553	-7.37724
Mean	-8.34958	-9.40824	-9.28123	-4.03756	-8.36363
Standard Deviation	0.57110	0.07306	0.10925	0.31096	0.49831

Results for HC-Best, HC-First, HC-Worst and SA [2]
n=30

	GA	HC-Best	HC-First	HC-Worst	SA
Min. value	-27.00472	-27.54316	-27.01624	-10.66123	-27.14902
Max. value	-22.36519	-26.58271	-25.65992	-7.07555	-23.28145
Mean	-25.19241	-27.00167	-26.30104	-8.03017	-25.11001
Standard Deviation	1.04261	0.21404	0.32959	0.69609	0.25

Results for HC-Best, HC-First, HC-Worst and SA [2]

5 Conclusion

In conclusion, GA can be classified as a competitive optimization tool in the context of lower-dimensional problems. It archives results which are closer to the global minimum on the De Jong 1, Rastrigin and Michalewicz functions. Nonetheless, once the dimensionality of the benchmark functions is getting complex, the GA's performance starts to decrease, considerably within optimization methods in comparison to Hill Climbing and Stimulated Annealing which are significantly more efficient and consistent. The source of the increased standard deviation which appears in the the context of a higher dimension and indicating a variability might be the algorithm's inherent exploration mechanisms and also the parameter settings. These could be undertuned for such complex landscapes. This fact is visible in the Rastrigin and Michalewicz functions' results, where GA has difficulties in finding the global minimum and is not as effective as the other methods. This study proves that GAs are suitable for lower-dimensional search spaces. Therefore, their default strategies may require improvements. For instance, fine-tuning of parameters, incorporation of local search techniques as well as hybridization by using other optimization methods in order to combine the complexities of higher-dimensional and multimodal function optimization.

References

- [1] Mathew, Tom V. "Genetic algorithm." Report submitted at IIT Bombay 53 (2012). [https://datajobstest.com/data-science-repo/Genetic-Algorithm-Guide-\[Tom-Mathew\].pdf](https://datajobstest.com/data-science-repo/Genetic-Algorithm-Guide-[Tom-Mathew].pdf).
- [2] Antoniev Valeriu-Gabriel. Hill Climbing and Simulated Annealing results are taken from "Homework 1". Unpublished manuscript 2024.
- [3] An, J., W. Ding, and C. Lin. "ChatGPT." tackle the growing carbon footprint of generative AI 615 (2023): 586. <https://chatgpt.com/>.

- [4] Lambora, Annu, Kunal Gupta, and Kriti Chopra. "Genetic algorithm-A literature review." 2019 international conference on machine learning, big data, cloud and parallel computing (COMIT-Con). IEEE, 2019. <https://ieeexplore.ieee.org/abstract/document/8862255>.
- [5] Hartmut Pohlheim, *Examples of Objective Functions. GEATbx Examples*, November 2005, http://www.geatbx.com/download/GEATbx_ObjFunExpl_v37.pdf.
- [6] Genetic Algorithms Explained By Example, Kie Codes, 13 July 2020. https://www.youtube.com/watch?v=uQj5UNhCPuo&t=526s&ab_channel=KieCodes.