Report - Function Minimization: Comparing Genetic Algorithm, Hill Climbing, and Simulated Annealing

Serban Robert Stefan and Borcan Petru-Razvan

01.12.2024

Abstract

This study explores the application of optimization algorithms to minimize four well-known benchmark functions: *De Jong*, *Schwefel*, *Rastrigin*, and *Michalewicz*. Genetic Algorithms (GA), Hill Climbing (HC), and Simulated Annealing (SA) were implemented and tested across 5, 10, and 30-dimensional configurations. The GA demonstrated superior performance, consistently achieving closer approximations to the global minima across all test cases. Its population-based approach, balanced exploration and exploitation, and use of elitism make it particularly effective for highly multimodal and complex optimization problems. Performance was analyzed using statistical metrics such as mean and standard deviation across 30 independent runs, highlighting GA's robustness and reliability compared to HC and SA. This study underscores the importance of selecting appropriate optimization algorithms for complex, high-dimensional problems.

1 Introduction

Optimization is a fundamental task in science and engineering, particularly for problems involving functions with multiple local minima. This study investigates the minimization of four benchmark functions: *De Jong 1*, *Schwefel's*, *Rastrigin's*, and *Michalewicz's*, which are widely used to evaluate optimization algorithms due to their complexity and multimodal nature.We employ a Genetic Algorithm (GA), an evolutionary optimization technique inspired by natural selection, to find the global minima for 5, 10, and 30-dimensional configurations of these functions. The algorithm uses selection, crossover, and mutation to explore the search space effectively.The algorithm is executed 30 times for statistical reliability. Results are analyzed using metrics such as mean and standard deviation to evaluate the GA's performance across different dimensions.This study aims to demonstrate the strengths and limitations of Genetic Algorithms in solving high-dimensional, complex optimization problems.

2 Problem Description

The primary objective of the program is to minimize four well-known mathematical functions used in optimization: De Jong, Schwefel, Rastrigin, and Michalewicz. Each of these functions is defined within a multidimensional search space, where the dimensions can vary between 5, 10, and 30 to simulate different levels of difficulty. These functions are widely used in the field of optimization due to their challenging nature, as they exhibit multiple local minima and varying levels of complexity.The multidimensional nature of these functions reflects real-world optimization problems, where algorithms must navigate intricate landscapes to identify the global minimum. By adjusting the dimensionality, the study aims to evaluate the efficiency and robustness of optimization techniques in scenarios ranging from simpler low-dimensional cases to highly complex high-dimensional spaces.

2.1 Optimization Functions

1.De Jong Function

The De Jong Function: The simplest test function is the De Jong function, also known as the sphere model. It is continuous, convex, and unimodal. It is defined as follows:

$$f(x) = \sum_{i=1}^{n} x_i^2$$

2.Schwefel Function

The Schwefel function is deceptive because the global minimum is geometrically distant, across the parameter space, from the next best local minima. Therefore, search algorithms are potentially prone to converging in the wrong direction. It is defined as follows:

$$f(x) = -\sum_{i=1}^{n} x_i \sin(\sqrt{|x_i|})$$

3.Rastrigin's Function

Rastrigin's function is based on function 1 with the addition of cosine modulation to produce many local minima. Thus, the test function is highly multimodal. However, the location of the minima is regularly distributed. It is defined as follows:

$$f(x) = 10n + \sum_{i=1}^{n} \left(x_i^2 - 10\cos(2\pi x_i)\right)$$

4.Michalewicz Function

The Michalewicz function is a multimodal test function. The parameter m defines the "steepness" of the valleys or edges. Larger M leads to a more difficult search. For very large m the function behaves like a needle in a haystack (values of the function for points in space outside the narrow peaks give very little information about the location of the global optimum). It is defined as follows:

$$f(x) = -\sum_{i=1}^{n} \sin(x_i) \left(\sin\left(\frac{(i+1)x_i^2}{\pi}\right)\right)^{2m}$$

3 Algorithms Used and Explanation

3.1 Initialization

The algorithm begins by generating an initial population of candidate solutions, where each solution is represented as a vector of values. For this implementation, the population size is set to 100, meaning the algorithm starts with 100 distinct candidate solutions.The algorithm runs for a fixed number of 10,000 generations . During each generation, the population evolves through crossover, mutation and selection to produce a new generation of solutions.

3.2 Fitness Evaluation

The fitness of each candidate solution is calculated using a fitness function tailored to the optimization problem. For `De Jong` and `Rastrigin` functions, fitness is inversely proportional to the objective function value, encouraging smaller values. For `Schwefel's` and `Michalewicz's` functions, the fitness is based directly on the objective value, with transformations applied to handle the negative scaling. This step determines the quality of each individual, influencing its likelihood of selection for reproduction.

3.3 Selection

The algorithm employs a selection process to identify parent solutions from the current population based on their fitness values. Fitter individuals have a higher probability of contributing to the next generation, ensuring that successful traits are preserved and propagated. To maintain high-quality solutions across generations, the algorithm incorporates elitism by directly transferring the top 10 elite individuals with the highest fitness values to the next generation without modification. This combined approach ensures that the best solutions found so far are retained, while the rest of the population evolves through crossover and mutation. By preserving elite individuals and emphasizing fitness in the selection process, the algorithm achieves incremental improvement over generations while maintaining diversity within the population.

3.4 Crossover

The crossover operation is applied to selected parent pairs with a probability defined by `CROSSOVER_RATE` (0.8). The program uses uniform crossover, where each gene has a 50% chance of being swapped between the two parents. This results in two offspring solutions that inherit traits from both parents. The advantage of having each gene with a 50% chance of being swapped between the two parents in uniform crossover lies in the balance it creates between exploration and exploitation in the Genetic Algorithm.

3.5 Mutation

Mutation is applied to each gene of an individual with a probability defined by `MUTATION_RATE` (0.1). If mutation occurs, the gene is perturbed by a small random value, sampled uniformly within a fraction of the domain size. To ensure feasibility, the mutated gene is constrained within the bounds. This process prevents the population from converging prematurely by introducing variability into the population.

3.6 Statistical Analysis

To ensure robust results, the Genetic Algorithm is executed `RUNS` (30) times for each function and dimensional configuration. After each run, the best solution's objective value is recorded. The program calculates the mean and standard deviation of these values, providing insights into the algorithm's reliability and consistency. The final results summarize the algorithm's performance across different scenarios.

4 Experimental Results

4.1 De Jong Function

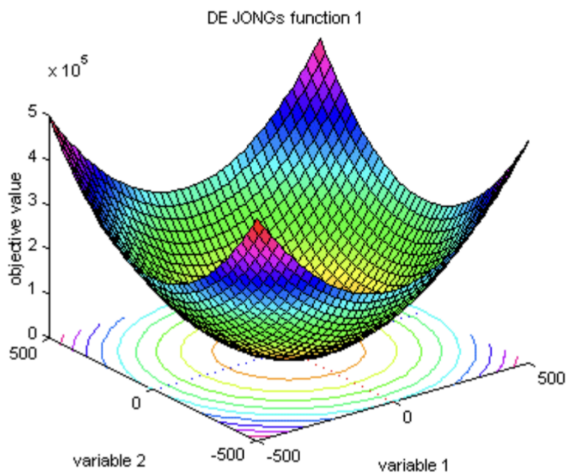

Figure 1: Visualisation of De Jong's function where the global minima are located at $f(x) = 0$; $x(i) = 0$, $i = 1 : n$. [4]

	HC - Best	HC - First	HC - Worst	Simulated Annealing	AG
$n = 5$					
Min. value	0.00000	0.00000	0.31097	0.00000	0.00000
Max. value	0.00000	0.00000	5.46014	0.00000	0.00007
Mean	0.00000	0.00000	2.89522	0.00000	0.00001
Standard Deviation	0.00000	0.00000	1.26908	0.00000	0.00001
$n = 10$					
Min. value	0.00000	0.00000	10.72072	0.00000	0.00001
Max. value	0.00000	0.00000	26.91588	0.00000	0.00018
Mean	0.00000	0.00000	19.88266	0.00000	0.00004
Standard Deviation	0.00000	0.00000	4.67432	0.00000	0.00003
$n = 30$					
Min. value	0.00000	0.00000	111.93527	0.00000	0.00044
Max. value	0.00000	0.00000	156.39117	0.00040	0.00250
Mean	0.00000	0.00000	135.67476	0.00002	0.00106
Standard Deviation	0.00000	0.00000	12.01665	0.00007	0.00044

Table 1: The results of the algorithms for the De Jong function at different dimensions [6]

4.2 Schwefel Function

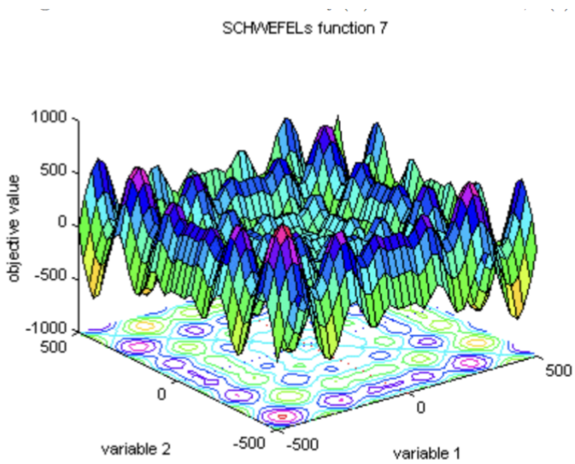

Figure 2: The global minima are located at f(x) =n·418.9829; x(i) = 420.9687, i= 1 : n [4]

	HC - Best	HC - First	HC - Worst	Simulated Annealing	AG
$n = 5$					
Min. value	0.00011	0.10375	379.19581	0.00069	-2094.91
Max. value	0.20865	102.70665	945.27857	34.44355	-2094.76
Mean	0.09165	27.87243	724.84412	1.01453	-2094.9
Standard Deviation	0.06684	23.72457	134.63654	5.42187	0.0285637
$n = 10$					
Min. value	0.62312	153.29677	1772.89931	0.00201	-4189.82
Max. value	248.34030	425.61503	2623.49310	34.96195	-4071.36
Mean	107.34842	288.57985	2255.27826	5.45540	-4185.81
Standard Deviation	53.76364	66.15256	208.63557	12.41442	21.2526
$n = 30$					
Min. value	888.09205	1313.39166	7881.37057	0.72923	-12568
Max. value	1411.22948	2038.94606	9653.71196	119.89815	-11739.17
Mean	1226.33757	1762.37164	9005.03426	30.88201	-12291
Standard Deviation	144.79832	138.93934	449.61408	38.17398	180.475

Table 2: The results of the algorithms for the Schwefel function at different dimensions [6]

4.3 Rastrigin's Function

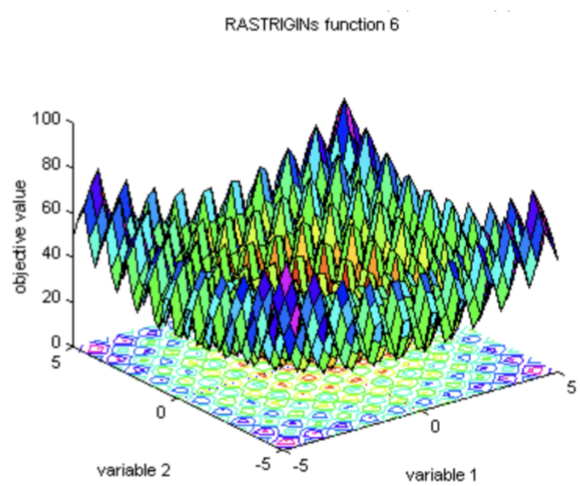

Figure 3: The global minima are located at f(x) = 0; x(i) = 0, i= 1 :n [4]

	HC - Best	HC - First	HC - Worst	Simulated Annealing	AG
			$n = 5$		
Min. value	0.00000	0.00000	5.87731	0.00000	0.00014
Max. value	1.23582	2.23078	33.83907	0.00007	0.00425
Mean	0.40413	0.93721	23.72123	2.67206	0.00183401
Standard Deviation	0.50265	0.59600	5.72894	1.68146	0.00122646
			$n = 10$		
Min. value	0.99496	3.22574	58.70354	1.98992	0.00279487
Max. value	5.45652	7.93322	101.13403	12.37491	0.0259587
Mean	3.61979	5.92097	85.79486	5.03176	0.00922885
Standard Deviation	0.97000	1.03978	8.94824	2.55480	0.00511625
			$n = 30$		
Min. value	18.85637	31.48343	342.51404	11.20057	0.0598776
Max. value	32.95540	41.46681	407.72442	28.22147	0.335986
Mean	27.70595	36.37522	382.71412	18.15091	0.176523
Standard Deviation	2.99251	2.57392	11.87121	4.01798	0.0685945

Table 3: The results of the algorithms for the Rastrigin function at different dimensions [6]

4.4 Michalewicz Function

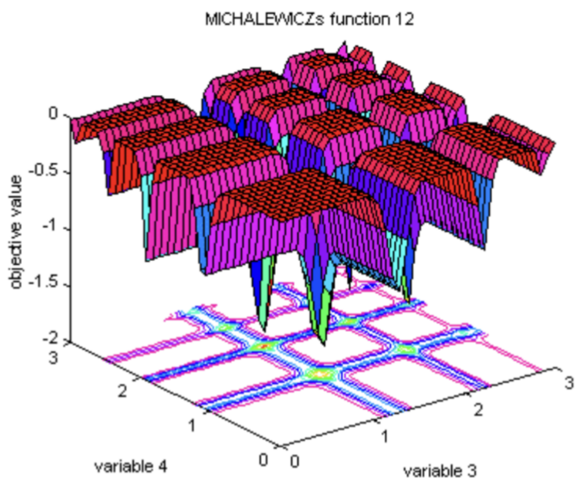

Figure 4: The global minima are located at f(x) =4.687(n= 5);f(x) =9.66(n= 10) [4]

	HC - Best	HC - First	HC - Worst	Simulated Annealing	AG
$n = 5$					
Min. value	-4.68766	-4.68765	-3.42603	-4.68593	-4.68765
Max. value	-4.68153	-4.66927	-2.33583	-3.54586	-4.68744
Mean	-4.68681	-4.68372	-2.81155	-4.23580	-4.68758
Standard Deviation	0.00140	0.00507	0.25297	0.31140	0.00007
$n = 10$					
Min. value	-9.57161	-9.53408	-4.56525	-9.24567	-9.64512
Max. value	-9.25546	-9.08584	-3.38553	-7.37724	-9.65991
Mean	-9.40824	-9.28123	-4.03756	-8.33663	-9.65839
Standard Deviation	0.07306	0.10925	0.31096	0.49831	0.00259
$n = 30$					
Min. value	-27.54316	-27.01624	-10.66123	-27.14902	-29.5769
Max. value	-26.58271	-25.65922	-7.07555	-23.28145	-29.4603
Mean	-27.00167	-26.30104	-8.03017	-25.11001	-29.5277
Standard Deviation	0.21404	0.32959	0.69609	0.99451	0.02628

Table 4: The results of the algorithms for the Michalewicz function at different dimensions [6]

5 Comparison of Results and Analysis

The experimental results demonstrate clear differences in the performance of Genetic Algorithms (GA), Hill Climbing (HC), and Simulated Annealing (SA) across the four benchmark functions (*De Jong*, *Schwefel*, *Rastrigin*, and *Michalewicz*) and across different dimensionalities ($n = 5$, $n = 10$, and $n = 30$).

5.1 Advantages of GA over HC and SA

The key advantages of the Genetic Algorithm over Hill Climbing and Simulated Annealing can be summarized as follows:

- **Population-Based Search:** Unlike HC and SA, GA operates on a population of solutions, enabling it to explore multiple regions of the search space simultaneously. This approach significantly reduces the likelihood of getting trapped in local minima.
- **Exploration and Exploitation:** The combination of crossover and mutation in GA enables a balance between exploration (searching new areas) and exploitation (refining existing solutions). This balance is particularly effective for highly multimodal functions.
- **Consistency Across Dimensionalities:** GA exhibits robust performance across different dimensional configurations, maintaining effectiveness even as the search space becomes more complex, unlike HC and SA, which degrade significantly with higher dimensions.
- **Preservation of High-Quality Solutions:** Through the use of elitism, GA ensures that the best-performing solutions are retained throughout the optimization process, guaranteeing steady progress toward the global minimum.

5.2 Observations on Performance Metrics

The results also highlight differences in statistical metrics such as the mean and standard deviation of the objective values:

- For *De Jong*, all algorithms perform well, but GA consistently achieves the global minimum with minimal variability, unlike HC and SA.
- For *Schwefel*, GA outperforms both HC and SA by a significant margin, particularly for higher dimensions, where HC and SA fail to approach the global minimum.

- For *Rastrigin*, the multimodal nature of the function makes HC's performance erratic, while GA demonstrates both precision and reliability.
- For *Michalewicz*, GA's ability to handle steep valleys and sharp edges in the landscape is evident from its superior results, while HC and SA struggle significantly.

Overall, the Genetic Algorithm emerges as the most robust and versatile optimization technique for the tested functions, particularly in scenarios involving high dimensionality and complex landscapes.

6 Conclusion

The comparative analysis between Genetic Algorithms, Hill Climbing, and Simulated Annealing revealed clear distinctions in their performance across the benchmark functions. GA consistently outperformed HC and SA, particularly for functions with higher dimensions and more complex landscapes. Its population-based search allowed simultaneous exploration of multiple regions in the search space, reducing the risk of getting trapped in local minima. The incorporation of crossover and mutation enabled a dynamic balance between exploration and exploitation, while elitism ensured the preservation of high-quality solutions throughout the optimization process. Overall, the Genetic Algorithm proved to be the most robust and versatile optimization technique among the three, maintaining consistent performance across all tested scenarios. These findings emphasize the advantages of population-based algorithms for solving high-dimensional, complex optimization problems.

References

[1] An, J., W. Ding, and C. Lin. "ChatGPT." tackle the growing carbon footprint of generative AI 615 (2023): 586. `https://chatgpt.com`

[2] YouTube, L. L. C. "YouTube." Retrieved 27 (2011): 2011. `https://www.youtube.com/watch?v=XP2sFzp2Rig` ;https://www.youtube.com/watch?v=uQj5UNhCPuo `https://www.youtube.com/watch?v=nhT56blfRpE` `https://www.youtube.com/watch?v=SWi-4IHFf1c`

[3] "C++ Reference." *cplusplus.com*, The C++ Resources Network `https://cplusplus.com`

[4] Pohlheim, Hartmut. "Geatbx: Genetic and evolutionary algorithm toolbox for use with matlab." IEE Colloquium on Applied Control Techniques Using MATLAB. Vol. 14. No. 1. 1998. `http://www.geatbx.com/docu/fcnindex-01.html#P89_3085`

[5] Croitoru, Eugen. "Genetic Algorithms Teaching Material." *Faculty of Computer Science, Alexandru Ioan Cuza University of Iași* `https://profs.info.uaic.ro/eugen.croitoru/teaching/ga/`

[6] Antoniev Valeriu-Gabriel. Hill Climbing And Simulated Annealing results are taken from "Homework 1". Unpublished manuscript, 2024.