Evaluation of Gradient Descent, Genetic Algorithm, and Differential Evolution for Global Optimization in Multi-Dimensional Search Spaces

University of Johannesburg

Tshepiso Clearance Mahoko – 220015607

1. Introduction

Optimization plays a critical role in science and engineering, where the goal is to find the best solution to a problem under given constraints. Traditional optimization methods, such as Gradient Descent (GD), rely on function derivatives to iteratively move toward a local minimum. While effective for convex functions, they struggle in complex, multi-modal landscapes, especially for functions with many local minima or narrow curved valleys.

Evolutionary Algorithms (EAs), inspired by principles of natural selection, offer a population-based alternative that explores the search space globally, without requiring gradient information. In this study, we explore three optimization methods: Gradient Descent, Genetic Algorithm (GA), and Differential Evolution (DE) and compare their performance on benchmark functions in an n-dimensional search space.

2. Problem

Finding the global minimum in high-dimensional, multi-modal functions are challenging since gradient-based methods like Gradient Descent often get trapped in local minima. While Evolutionary Algorithms (GA and DE) perform global, population-based search, their comparative performance on representative benchmark functions, including Rastrigin, Ackley, Schwefel, and Six-Hump Camel, remains unclear. This study aims to systematically evaluate GD, GA, and DE to determine which method most reliably finds global minima across diverse, complex landscapes.

3. Literature

3.1 Gradient Descent (GD)

Gradient Descent is a first-order iterative optimization algorithm widely used in numerical optimization and machine learning. It works by moving iteratively in the direction of the steepest descent of the objective function to find local minima. Formally, the update rule is expressed as:

$$x_{k+1} = x_k - \eta \nabla f(x_k)$$

where $x_k$represents the candidate solution at iteration k, η is the learning rate controlling the step size, and $\nabla f(x_k)$is the gradient of the function at x_k[1]. GD is particularly effective for convex functions where the gradient consistently points toward the global minimum. However, in non-convex, multi-modal, or high-dimensional landscapes, GD may converge to a local minimum rather than the global optimum. The algorithm is highly sensitive to the initialization of the solution, the choice of learning rate, and the curvature of the function. Poor initialization or an inappropriate learning rate can lead to oscillations around narrow valleys, slow convergence, or divergence altogether.

Despite its limitations, GD is computationally efficient because it relies solely on gradient information, and its simplicity makes it a natural baseline for comparison with more complex, population-based optimization methods [1].

3.2 Genetic Algorithm (GA)

Genetic Algorithms are stochastic, population-based optimization methods inspired by the principles of natural evolution, first proposed by John Holland in the 1970s. Unlike gradient-based methods, GA does not require derivative information, making it suitable for non-differentiable, noisy, or multi-modal functions [2].

The main steps of GA include:

1. Initialization: A population of candidate solutions is randomly generated within the bounds of the n-dimensional search space [2].
2. Selection: Individuals are evaluated based on their fitness, and a subset is selected to propagate their genetic material to the next generation. Common strategies include roulette-wheel selection, tournament selection, and rank-based selection [3].
3. Crossover (Recombination): Selected parents are combined to produce offspring, promoting the combination of high-quality traits. Typical crossover operators include single-point, two-point, and uniform crossover [4].

4. Mutation: Small random perturbations are introduced to offspring to maintain genetic diversity and avoid premature convergence. Mutation helps the population escape local minima and explore the search space more thoroughly [3].

GA is highly flexible and can be applied to continuous or discrete optimization problems in high-dimensional spaces. Its performance is influenced by population size, mutation rate, crossover probability, and number of generations. Larger populations improve exploration at the cost of higher computational effort, while higher mutation rates increase diversity but may slow convergence. The stochastic nature of GA allows broad exploration of the search space, reducing the likelihood of getting trapped in local optima [2][4].

3.3 Differential Evolution (DE)

Differential Evolution is another population-based evolutionary algorithm, introduced by Storn and Price in 1997, designed specifically for continuous global optimization [5]. DE evolves candidate solutions by perturbing existing individuals based on the weighted differences between randomly chosen members of the population:

$$v_i = x_r + F \cdot (x_s - x_t)$$

where x_r, x_s, x_t are randomly selected population members, and F is the mutation factor controlling the amplification of the differential variation. DE combines mutation with crossover and selection to balance global exploration and local exploitation [6].

Compared to GA, DE typically exhibits faster convergence and higher reliability in finding global optima due to its differential mutation mechanism, which generates diverse candidate solutions without complex recombination. DE's performance is less sensitive to initialization than gradient descent, as population differences guide the search rather than gradients. This makes DE suitable for functions with narrow valleys, oscillatory landscapes, or deceptive minima [5][6]. Additionally, DE requires fewer control parameters than GA, simplifying practical implementation while maintaining robust performance [6].

4. Benchmark Functions

To evaluate the algorithms, a set of multi-modal and non-convex benchmark functions was selected. These functions provide representative examples of diverse optimization landscapes some featuring many local minima, others containing deceptive valleys or flat plateaus that pose challenges to gradient-based methods [7].

Evolutionary Algorithms (EAs) are particularly effective on such problems because they perform global, population-based exploration without depending on gradient information. This allows them to handle discontinuities, non-smooth regions, and deceptive search spaces that often hinder deterministic optimizers [2][5].

The chosen functions Rastrigin, Ackley, Schwefel, and Six-Hump Camel jointly assess each algorithm's ability to escape local minima and flat regions, navigate deceptive and rugged landscapes, and maintain convergence efficiency in multi-dimensional spaces [7].

Table 1. *Characteristics of Benchmark Functions*

Function	Characteristics
Rastrigin	Highly multi-modal with numerous evenly distributed local minima
Ackley	Flat outer regions with a central basin and vanishing gradients
Schwefel	Deep deceptive local minima
Six-Hump Camel	Multiple local minima with two global minima in a bounded 2D region

All selected benchmark functions can be extended to $n$dimensions ($n \geq 2$), making them suitable for studying scalability and algorithmic robustness. As dimensionality increases, the search space expands exponentially, amplifying the need for effective exploration mechanisms [7].

Random initialization plays a crucial role:

- For Gradient Descent (GD), convergence heavily depends on the starting point, often leading to local minima [1].
- For EAs, population diversity mitigates this sensitivity, enabling exploration of multiple regions concurrently [2][5].

Testing algorithms on these diverse landscapes provides a balanced and generalizable evaluation framework, illustrating their ability to manage exploration–exploitation trade-offs. This reflects real-world optimization contexts such as engineering design, machine learning model tuning, and control parameter optimization, where objective functions are often complex, noisy, or high-dimensional [1][5].

5. Methodology

1. Initialization: Candidate solutions were randomly generated within the bounds of each function. Random initialization strongly affected the performance of GD, while GA and DE were less sensitive.

2. Parameters:

    - GD: learning rate, max iterations, epsilon
    - GA: population size, generations, mutation rate
    - DE: population size, generations, mutation factor F, crossover probability CR

3. Evaluation: Each algorithm was applied to the n-dimensional functions, and the best fitness, convergence rate, and number of function evaluations were recorded.

4. Visualization: Convergence curves and 3D function surfaces were plotted to observe algorithm behaviour.

6. Experimental Results

The experiments were conducted across the four benchmark functions Rastrigin, Ackley, Schwefel, and Six-Hump Camel evaluating Gradient Descent (GD), Genetic Algorithm (GA), and Differential Evolution (DE). The focus was on convergence behavior, robustness, and consistency in navigating multi-dimensional search spaces.

6.1 Rastrigin Function

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

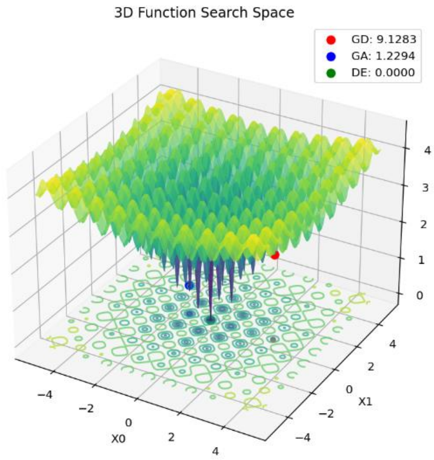

Fig. 1. *Rastrigin search space*

Observations from Fig. 1 and Fig. 2 show that GD frequently converged to local minima due to the highly multi-modal landscape, with noticeable oscillations. GA demonstrated consistent convergence, although slower than DE, while DE efficiently explored the search space and successfully located the global minimum.

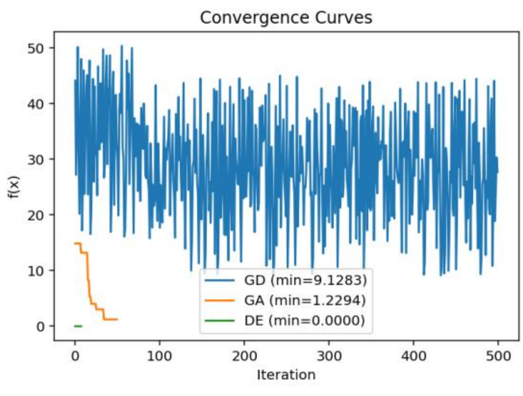

Figure 2. *GD, GA, and DE convergence on Rastrigin.*

6.2 Ackley Function

$$f(x) = -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^{n} \cos(2\pi x_i)\right) + 20 + e, x_i \in [-5,5]$$

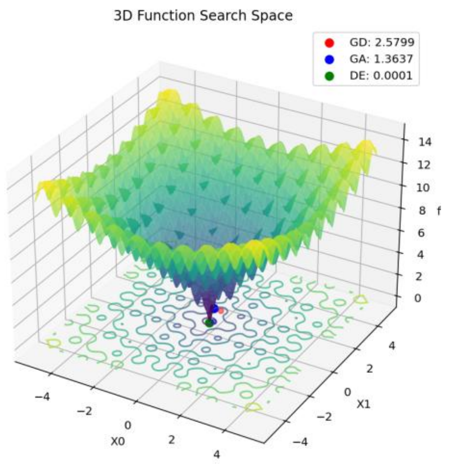

Fig. 3. Ackley search space

Observations from Fig. 3 and Fig. 4 indicate that GD struggled in flat outer regions and showed slow convergence near the central basin, while GA avoided some flat regions due to population diversity. DE consistently reached the global minimum, demonstrating the advantage of mutation-based exploration.

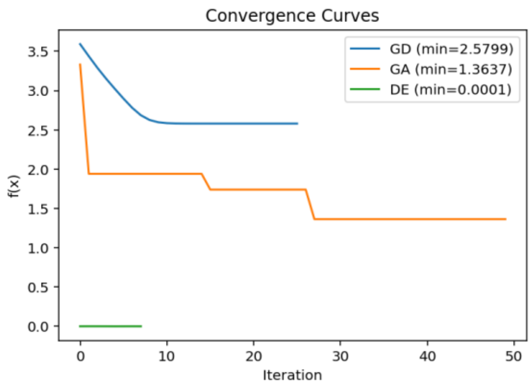

Fig. 4. *GD, GA, and DE convergence on Ackley*

6.3 Schwefel Function

$$f(x) = 418.9829 \cdot n - \sum_{i=1}^{n} x_i \sin\left(\sqrt{|x_i|}\right), x_i \in [-500,500]$$

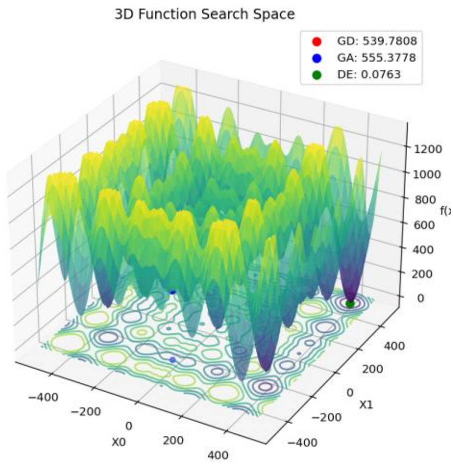

Fig. 5. *Schwefel search space*

Observations from Fig. 5 and Fig. 6 indicate that GD was frequently trapped in deceptive local minima. GA improved convergence through crossover and mutation but required more iterations, while DE's randomized differential mutation enabled a robust global search and rapid convergence to the true minimum.

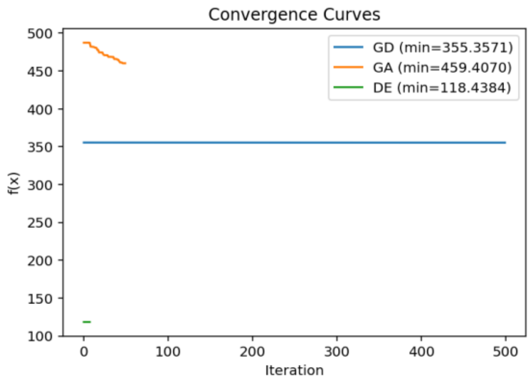

Fig. 6. *GD, GA, and DE convergence on Schwefel*

6.4 Six-Hump Camel Function

$$f(x_1, x_2) = (4 - 2.1x_1^2 + \frac{x_1^4}{3})x_1^2 + x_1x_2 + (-4 + 4x_2^2)x_2^2, x_1 \in [-3,3], x_2 \in [-2,2]$$

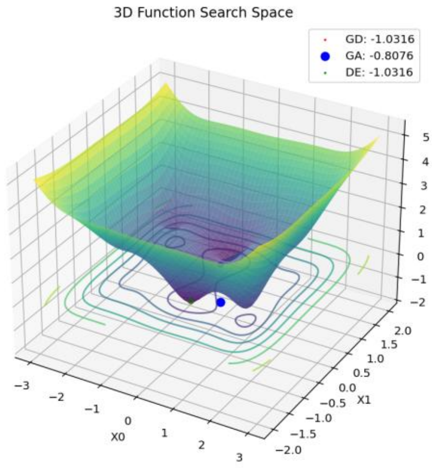

Fig. 7. Six-Hump Camel search space

Observations based on Fig. 7 and Fig. 8 show that GD can converge to one of the local minima depending on initialization. GA explores multiple minima and consistently finds one of the two global minima, while DE efficiently identifies both global minima and demonstrates superior exploration.

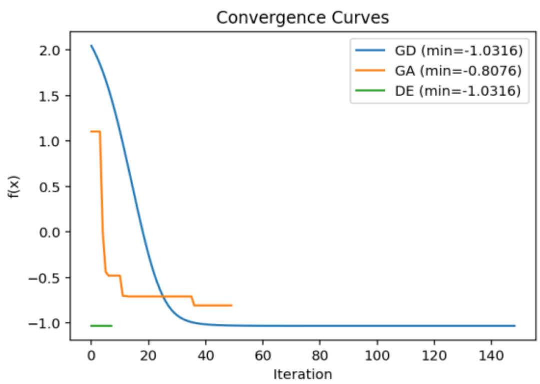

Fig. 8. *GD, GA, and DE convergence on Six-Hump Camel*

6.4. Empirical Analysis

Due to the stochastic nature of each optimization algorithm, all methods were executed 50 times per benchmark function to allow a fair and robust comparison of performance, as shown in Table 2.

Table 2. *Statistical Algorithm performance*

Function	Metric	Gradient Descent	Genetic Algorithm	Differential Evolution
Rastrigin	Mean	14.42016	2.07599	0.48737
	Std Dev	4.93312	2.84473	0.73336
	Median	17.20869	1.28588	0.00000
	Success Rate (Wins %)	0.00000	14.00000	86.00000
Ackley	Mean	8.49945	0.19698	0.00000
	Std Dev	2.43610	0.24368	0.00000
	Median	9.00109	0.13745	0.00000
	Success Rate (Wins %)	0.00000	0.00000	100.00000
Schwefel	Mean	482.71558	344.38652	101.22924
	Std Dev	263.20280	177.34905	94.91248
	Median	455.88006	321.14837	118.43836
	Success Rate (Wins %)	6.00000	8.00000	86.00000
Six-Hump Camel	Mean Final Value	0.13509	-0.85776	-1.03163
	Std Dev	1.38521	0.32473	0.00001
	Median	-0.21546	-1.02317	-1.03163
	Success Rate (Wins %)	2.00000	0.00000	98.00000

Across all functions, Differential Evolution (DE) consistently demonstrated superior performance, achieving the lowest mean and median values and the highest success rates, ranging from 86% to 100%. On the Rastrigin function, DE reached a mean final value of 0.48737 with an 86% success rate, while on Ackley it reached 0.00000 with a 100% success rate. Similar trends were observed for Schwefel and Six-Hump Camel functions, where DE consistently outperformed both Gradient Descent (GD) and Genetic Algorithm (GA).

Gradient Descent (GD) exhibited the highest variability and low success rates across all functions, reflecting its sensitivity to local minima and limited capability in navigating complex, multi-modal landscapes. Genetic Algorithm (GA) performed moderately well on Rastrigin and Schwefel but never matched DE's robustness, achieving moderate success rates and higher variability compared to DE.

Overall, these empirical results highlight that population-based stochastic methods, particularly DE, are more effective and reliable for global optimization in multi-dimensional search spaces. The consistent performance of DE across all four benchmark functions establishes it as the best-performing optimization method overall.

7. Discussion

The comparative evaluation of Gradient Descent (GD), Genetic Algorithm (GA), and Differential Evolution (DE) across multiple benchmark functions highlights the impact of algorithm design, search space dimensionality, and initialization on optimization performance. As the number of dimensions increases, the search space expands exponentially, introducing more local minima and complex curvature patterns. GD, being a local, gradient-based method, is highly sensitive to these complexities and frequently converges to suboptimal local minima, particularly in non-convex and multi-modal landscapes. Oscillatory or slow convergence was observed in narrow valleys, limiting GD's reliability in complex functions.

In contrast, population-based evolutionary algorithms (EAs), such as GA and DE, demonstrated superior robustness. GA, through population diversity, crossover, and mutation, better avoided local minima than GD, exploring regions that gradient methods could not reach, although its convergence was slower than DE. DE consistently outperformed both GD and GA, frequently locating global minima even in highly deceptive landscapes like Schwefel, with faster convergence and minimal sensitivity to initialization due to its differential mutation strategy. By maintaining multiple candidate solutions, GA and DE enable broader exploration across the n-dimensional search space, reducing the likelihood of premature convergence.

Empirical results reinforce these conclusions. GD showed high variability and consistently low success rates, GA performed moderately, and DE achieved the highest success rates (86–100%) across all functions. The combination of stochastic mutation, recombination, and selection allows EAs to navigate complex, high-dimensional search spaces effectively, whereas GD's reliance on gradient information limits its performance in multi-modal, non-convex landscapes.

Overall, Differential Evolution demonstrates the best balance between exploration and convergence speed, GA offers moderate reliability, and GD is effective only in simpler convex regions. These observations underscore the superiority of evolutionary algorithms for global optimization, emphasizing their robustness, adaptability, and consistent ability to locate global optima across diverse and challenging problem landscapes.

8. Conclusion

This study demonstrates that Evolutionary Algorithms, particularly Differential Evolution (DE), provide a robust approach to global optimization in n-dimensional, multi-modal functions. While Gradient Descent (GD) can be effective in simple convex problems, EAs consistently outperform it in complex landscapes due to their population-based exploration, mutation, and crossover mechanisms. DE achieved the best balance between global exploration and convergence speed, showing high success rates and robustness across all tested benchmark functions. These results reinforce the effectiveness and adaptability of Evolutionary Algorithms in navigating complex, high-dimensional optimization problems.

Further studies could explore more sophisticated evolutionary strategies, such as island models or hybrid approaches, for problems with very large search spaces and bounds.

References

[1] Ruder, S. (2016). *An overview of gradient descent optimization algorithms.* arXiv:1609.04747.
[2] Katoch, S., Chauhan, S. S., & Kumar, V. (2021). *A review on genetic algorithm: past, present, and future.* Multimedia Tools and Applications, 80, 8091–8126.
[3] Ghosh, A., et al. (2022). *Mutation strategies in evolutionary optimization: A survey.* Information Sciences, 606, 1055–1079.
[4] Zhang, X., et al. (2023). *Adaptive genetic algorithms for multi-modal optimization.* Expert Systems with Applications, 220, 119731.
[5] Storn, R., & Price, K. (1997). *Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces.* Journal of Global Optimization, 11, 341–359.
[6] Zhang, S. X., et al. (2024). *Differential evolution with evolutionary scale adaptation.* Computers & Industrial Engineering, 189, 108704.
[7] Liu, Y., et al. (2019). *Benchmark functions for evaluating evolutionary algorithms: A comprehensive survey.* Applied Soft Computing, 80, 690–710.