Comparative Study of GA and PSO on Benchmark Functions

ICT358 : DATA SCIENCE

Group Members
Pan Phyu Phyu Khin – 6509963
Khant Si Thu Win – 6509691
Hein Htet Soe - 6509961

Table of Contents

1.Introduction

This project explores the application of two widely-used metaheuristic algorithms—Genetic Algorithm (GA) and Particle Swarm Optimization (PSO)—to solve continuous optimization problems using the Sphere and Rastrigin benchmark functions. The aim is to compare the performance of GA and PSO in high-dimensional (30D) search spaces with respect to convergence speed, accuracy, and computational efficiency. Both algorithms are inspired by natural processes: GA simulates biological evolution through selection, crossover, and mutation, while PSO imitates the social behavior of swarms, where particles update their positions based on both individual and collective experiences. The choice of benchmark functions is deliberate. The Sphere function is a simple, convex, and unimodal function that serves as a baseline to test an algorithm's ability to converge smoothly to the global optimum. In contrast, the Rastrigin function is complex, highly multimodal, and non-separable, making it a tough test for global optimization due to its numerous local minima. These functions offer complementary insights into how the algorithms perform under different landscape characteristics.

The GA implementation in this project includes elitism to retain the best solution across generations, as well as non-uniform Gaussian mutation that becomes more focused as generations progress, aiding in exploration early and exploitation later. PSO, on the other hand, incorporates adaptive inertia damping and velocity clamping, which help balance exploration and convergence throughout the optimization run. Each algorithm was tested on both benchmark functions across five independent trials to account for stochastic variability. The convergence behavior is tracked over generations or iterations, and the best solutions are recorded to assess consistency and performance.

By analyzing how GA and PSO handle both smooth and rugged optimization landscapes, this project aims to provide insights into their relative strengths and weaknesses. Such understanding is valuable not only for academic benchmarking but also for guiding the choice of algorithms in real-world optimization scenarios, where the nature of the objective function may be unknown or difficult to characterize analytically.

2. Benchmark Function Selection

The evaluation of optimization algorithms often relies on their performance on well-established benchmark functions. These mathematical test functions are specifically designed to assess various aspects of algorithm behavior, such as convergence accuracy, robustness, and the ability to escape local optima. Benchmark functions possess known characteristics—such as modality (number of local optima), separability, and the location of the global optimum—allowing for controlled, repeatable comparisons between different optimization methods.

For this project, two widely-used benchmark functions were selected to evaluate the performance of Genetic Algorithms (GA) and Particle Swarm Optimization (PSO): the Sphere function and the Rastrigin function. These functions are chosen to represent distinct optimization challenges—one being simple and unimodal, and the other being complex and multimodal. Both are evaluated in 30-dimensional space, providing a rigorous test of scalability and effectiveness.

2.1 Sphere Function

The Sphere function is one of the simplest and most commonly used benchmark functions. It is unimodal, separable, and convex, with a smooth, continuous surface that leads directly to the global minimum. Its simplicity makes it an ideal candidate for evaluating an algorithm's ability to converge quickly and accurately in a smooth, non-deceptive landscape.

The function is defined as:

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

where $x = (x_1, x_2, \dots, x_d) \in \mathbb{R}^d$ is the dimensionality of the problem (set to 30 in this study). The global minimum is located at x* = (0, 0, ..., 0),with f(x*) = 0.
Despite its simplicity, the Sphere function is useful for verifying the basic correctness, convergence behavior, and computational efficiency of optimization algorithms.

2.2 Rastrigin Function

The Rastrigin function is a highly multimodal, non-separable benchmark function characterized by a large number of regularly distributed local minima. It is more challenging than the Sphere function, as the presence of many deceptive regions can trap algorithms that are prone to premature convergence or poor global exploration.

The function is defined as:

$$f(x) = A \cdot d + \sum_{\{i=1\}}^{d} (x_i^2 - A \cdot \backslash cos(2\pi x_i))$$

where A = 10, d = 30, and x = (x_1, x_2, ..., x_d). The global minimum is again located at x* = (0, 0, ..., 0), where f(x*) = 0.

Due to its complex landscape, the Rastrigin function is frequently used to evaluate how well an algorithm balances exploration and exploitation. It presents a significant challenge for optimization methods that do not adapt well to multimodality or struggle to escape local optima.

By including both the Sphere and Rastrigin functions in a 30-dimensional setting, this study is able to assess the algorithms' performance across both smooth and complex optimization landscapes. The contrast between these two functions provides a balanced and insightful comparison of the strengths and limitations of GA and PSO.

3. Algorithms Selection

We use PSO for its simplicity and fast convergence on smooth functions like Sphere, and GA for its strong exploration ability and robustness on complex landscapes like Rastrigin. Combining both allows us to effectively analyze performance across different problem types, demonstrating their strengths in high-dimensional optimization.

3.1 Particle Swarm Optimization (PSO)

We chose PSO for this project because it is a highly effective and intuitive algorithm for solving continuous, real-valued optimization problems. It mimics the collective behavior of swarms (like birds flocking or fish schooling), making it a good fit for exploring large search spaces with minimal parameter tuning. PSO is naturally suited for continuous functions like Sphere and Rastrigin. Since particles can move freely in continuous space, the algorithm fits well with the real-valued variables in both benchmark functions. In our code, PSO uses adaptive inertia weight and velocity clamping, which helps particles explore broadly at the beginning and converge more precisely later. This balances exploration (searching new areas) and exploitation (refining known good areas). On smooth, convex problems like Sphere, PSO is particularly effective because the gradient-free nature of the algorithm allows it to quickly converge to the global optimum. While Rastrigin is multimodal, PSO still offers a good baseline because its movement strategies (global and personal bests) can guide the swarm to promising areas. Although PSO can get stuck in local minima, observing its performance here helps us understand its limits.

3.2 Genetic Algorithm (GA)

GA was chosen because of its robust global search capability and strong performance in complex, multimodal landscapes like the Rastrigin function. It is inspired by the principles of evolution, where a population of solutions evolves over time using operators like selection, crossover, and mutation. GA's crossover operator allows mixing of features from two solutions, which is extremely helpful in escaping local minima—a common challenge in functions like Rastrigin. Mutation introduces randomness and diversity, helping the algorithm explore areas that might not be visited by guided movement like in PSO. We implemented elitism in our GA code to ensure that the best solutions are preserved from one generation to the next, preventing regression. The non-uniform mutation used in this GA changes over generations—starting more exploratory, then becoming more exploitative. This adaptive behavior improves convergence while avoiding premature stagnation. On the Sphere function, GA may not be faster than PSO, but it still demonstrates stable convergence and robustness.GA simulates natural evolution using selection, crossover, and mutation.

4. Literature Review

Metaheuristic algorithms have gained widespread attention for solving optimization problems that are difficult to address using traditional mathematical programming or deterministic methods. Among these algorithms, Genetic Algorithm (GA) and Particle Swarm Optimization (PSO) stand out for their simplicity, flexibility, and strong performance across a wide range of problem domains. This review discusses how GA and PSO have been applied to optimization problems similar to the Sphere and Rastrigin benchmark functions, with a focus on their design principles, strengths, limitations, and relevance to high-dimensional continuous optimization.

4.1 Genetic Algorithm (GA)

Genetic Algorithm (GA), introduced by Holland (1975), is an evolutionary algorithm inspired by the process of natural selection. It operates on a population of candidate solutions that evolve over generations through the application of genetic operators: selection, crossover, and mutation. The key idea is to promote the survival and combination of better solutions over time while maintaining diversity in the population.

4.1.1 Application to Benchmark Functions

GA has been successfully applied to a variety of benchmark functions such as Sphere, Ackley, Rastrigin, and Rosenbrock. According to Sadhu et al. (2023), GA performs particularly well on multimodal functions like Rastrigin due to its global search capability and ability to maintain diversity via mutation. Haznedar et al. (2020) demonstrated that GA can avoid premature convergence by using adaptive mutation and elitism, allowing it to consistently find the global minimum in rugged landscapes.

4.1.2 Strengths

Exploration Capability: GA's crossover and mutation mechanisms allow the algorithm to search a wide space effectively, making it well-suited for multimodal functions with many local minima.
Diversity Maintenance: Through mutation and stochastic selection, GA maintains population diversity, reducing the chance of getting stuck in local optima.
Global Optimization Potential: GA is less likely to converge prematurely compared to algorithms like standard PSO.

4.1.3 Weaknesses

Slow Convergence: On simple unimodal functions like Sphere, GA may take more time to converge than PSO due to its population-based operations.

Parameter Sensitivity: Performance is highly dependent on tuning parameters like mutation rate, crossover rate, and population size.
Computational Cost: Maintaining and evolving a population with crossover and mutation operations can be computationally intensive, especially in high dimensions.

4.2 Particle Swarm Optimization (PSO)

PSO, developed by Kennedy and Eberhart (1995), is a swarm intelligence algorithm inspired by the social behavior of birds and fish. Unlike GA, PSO does not use crossover or mutation. Instead, each particle in the swarm updates its velocity and position based on its personal best and the global best positions in the swarm.

4.2.1 Application to Benchmark Functions

PSO has shown excellent performance on unimodal and smooth functions like Sphere, where quick convergence is desired. Abbas et al. (2025) found that standard PSO can outperform GA on such functions due to its directional movement and faster exploitation of promising regions. However, Liang et al. (2006) noted that PSO tends to suffer from premature convergence when applied to multimodal functions like Rastrigin, especially in higher dimensions.

4.2.2 Strengths

Fast Convergence: PSO converges rapidly, particularly on convex, unimodal functions such as Sphere.
Simple to Implement: PSO has fewer parameters to tune and no complex operations like crossover or mutation.
Computationally Efficient: With simple velocity and position updates, PSO has a lower computational burden compared to GA.

4.2.3 Weaknesses

Premature Convergence: PSO is prone to getting trapped in local optima, especially in multimodal landscapes like Rastrigin.
Lack of Diversity Mechanism: Standard PSO lacks built-in mechanisms to maintain diversity, unlike GA's mutation operator.

Performance Drops in High Dimensions: In high-dimensional problems, the influence of global best diminishes, making convergence more difficult without enhancements (dos Santos et al., 2023).

4.3 Comparison and Relevance to the Project

In the context of this project, Sphere and Rastrigin functions were selected to evaluate and contrast the performance of GA and PSO across simple (Sphere) and complex (Rastrigin) landscapes in 30 dimensions:

Sphere Function: A smooth, convex, and unimodal function. PSO is expected to perform well due to its efficient search dynamics and fast convergence. GA also performs adequately but may be slower due to population-level operations.

Rastrigin Function: A highly multimodal and non-separable function. GA tends to outperform PSO in this scenario because its crossover and mutation promote better global exploration and prevent premature convergence.

By using both GA and PSO, this project examines how these algorithms behave under different optimization challenges. GA's strength in escaping local optima complements PSO's efficiency in smooth search spaces, making the pair a powerful choice for benchmarking and comparative studies.

4.4 Conclusion

The literature strongly supports the selection of both GA and PSO for tackling high-dimensional optimization tasks. GA's flexibility and robustness make it suitable for complex and deceptive fitness landscapes, while PSO's speed and simplicity make it an ideal choice for smooth, well-behaved functions. Understanding the strengths and weaknesses of each algorithm in the context of specific functions helps guide algorithm selection and design for real-world problems.

Here is the Methodology, Results, and Analysis that includes detailed sections on accuracy, convergence, and cost for each of the three dimensions (10, 20, and 30), including Implementation Details and the experimental process based on the provided data.

5. Methodology

The goal of this study is to evaluate and compare the performance of two widely used metaheuristic optimization algorithms—Genetic Algorithm (GA) and Particle Swarm Optimization (PSO)—on two benchmark functions: Sphere and Rastrigin. The study is performed across three different dimensionalities: 10, 20, and 30. The evaluation is based on accuracy, convergence rate, and computational cost, with the goal of understanding how each algorithm behaves under different conditions.

5.1 Implementation Details

The algorithms were implemented using Python, leveraging NumPy for numerical operations. Below are the key steps for implementing the GA and PSO algorithms, with specifications for each dimension.

Genetic Algorithm (GA) Configuration:

- Selection: Tournament selection was employed to ensure diversity while selecting parents.
- Crossover: A single-point crossover method was used with a high probability to combine the genetic material of two parents.
- Mutation: A bit-flip mutation was applied at a low rate to introduce variability.
- Elitism: The best individuals from the current population are directly passed to the next generation.

PSO Configuration:

- Inertia Weight: The inertia weight (w) controls the impact of the previous velocity on the particle's current velocity.
- Cognitive Coefficient (c1): The cognitive term determines how much the particle is influenced by its own best solution found so far.
- Social Coefficient (c2): The social term adjusts how much the particle is influenced by the best global solution.

- Velocity Clamping: Velocity clamping was implemented to prevent particles from escaping the search space.

The implementation was set to handle dimensions 10, 20, and 30, with each algorithm running for 700-1500 iterations, depending on the dimension.

Tuning Parameters for Different Dimensions:

- 10D GA: Population size = 250, Generations = 700, Crossover rate = 0.9, Mutation rate = 0.05
- 20D GA: Population size = 400, Generations = 1000, Crossover rate = 0.95, Mutation rate = 0.05
- 30D GA: Population size = 400, Generations = 800, Crossover rate = 0.9, Mutation rate = 0.1

- 10D PSO: Swarm size = 250, Iterations = 700, w = 0.6, c1 = 1.5, c2 = 1.5
- 20D PSO: Swarm size = 400, Iterations = 1500, w = 0.9, c1 = 2.0, c2 = 2.0
- 30D PSO: Swarm size = 400, Iterations = 1500, w = 0.9, c1 = 2.0, c2 = 2.0

5.2 Experimental Design

5.2.1 10D Experiments

- GA: Results show the best value on Sphere as 0.000353 with an average value of 0.000667 and an average runtime of 6.00 seconds. On Rastrigin, GA performed with a best value of 0.074301 and an average value of 0.159813 with an average runtime of 6.32 seconds.
- PSO: On the Sphere function, PSO achieved the best value of 0.000000 and the average value of 0.000000, with a significantly faster average runtime of 0.56 seconds. For Rastrigin, PSO's best value was 1.085534, and the average value was 4.618379, with an average runtime of 1.05 seconds.

5.2.2 20D Experiments

- GA: The best value on Sphere was 0.000018, with an average value of 0.000022 and an average runtime of 34.03 seconds. On Rastrigin, the best value was 0.003210, with an average value of 0.999571, and an average runtime of 33.61 seconds.
- PSO: For Sphere, PSO achieved best and average values of 0.000000 with an average runtime of 3.24 seconds. On Rastrigin, PSO's best value was 1.989918, the average value was 6.168740, and the average runtime was 6.07 seconds.

5.2.3 30D Experiments

- GA: On Sphere, the best value was 0.000030 with an average value of 0.000047, and the average runtime was 12.67 seconds. On Rastrigin, the best value was 0.008020, with an average value of 1.997981, and an average runtime of 13.82 seconds.
- PSO: For Sphere, PSO achieved a best value and average value of 0.000000, with an average runtime of 1.96 seconds. On Rastrigin, PSO's best value was 4.984266, the average value was 9.213937, and the average runtime was 3.61 seconds.

5.3 Performance Metrics

5.3.1 Accuracy

- GA: The GA performed very well on the Sphere function, with the best value approaching 0.0000 across all dimensions. However, its performance on Rastrigin was less accurate, especially in higher dimensions, where the average value remained far above the global optimum.
- PSO: PSO achieved perfect accuracy on Sphere across all dimensions, with the best and average values equal to 0.0000. However, PSO struggled with Rastrigin, especially as the dimensionality increased, showing poor accuracy (values above 4 for Rastrigin in higher dimensions).

5.3.2 Convergence

- GA: The GA exhibited steady convergence, particularly on the Rastrigin function. It consistently showed gradual improvements over multiple generations, though it required more generations and time to converge, especially in higher dimensions.

- PSO: PSO was faster in converging on the Sphere function but struggled significantly on the Rastrigin function, especially in higher dimensions. It showed quick convergence but with poor results as it failed to escape local optima in the Rastrigin landscape.

5.3.3 Computational Cost

- GA: The GA was computationally more expensive than PSO, with runtimes increasing with the number of dimensions. For 10D, the runtime was 6.00 seconds, which increased to 34.03 seconds for 20D, and 12.67 seconds for 30D.
- PSO: PSO was much faster compared to GA. The runtime for 10D was 0.56 seconds, for 20D it was 3.24 seconds, and for 30D, it was 1.96 seconds. However, PSO's speed advantage comes with reduced accuracy on more complex functions like Rastrigin.

6. Results and Analysis

6.1 Accuracy

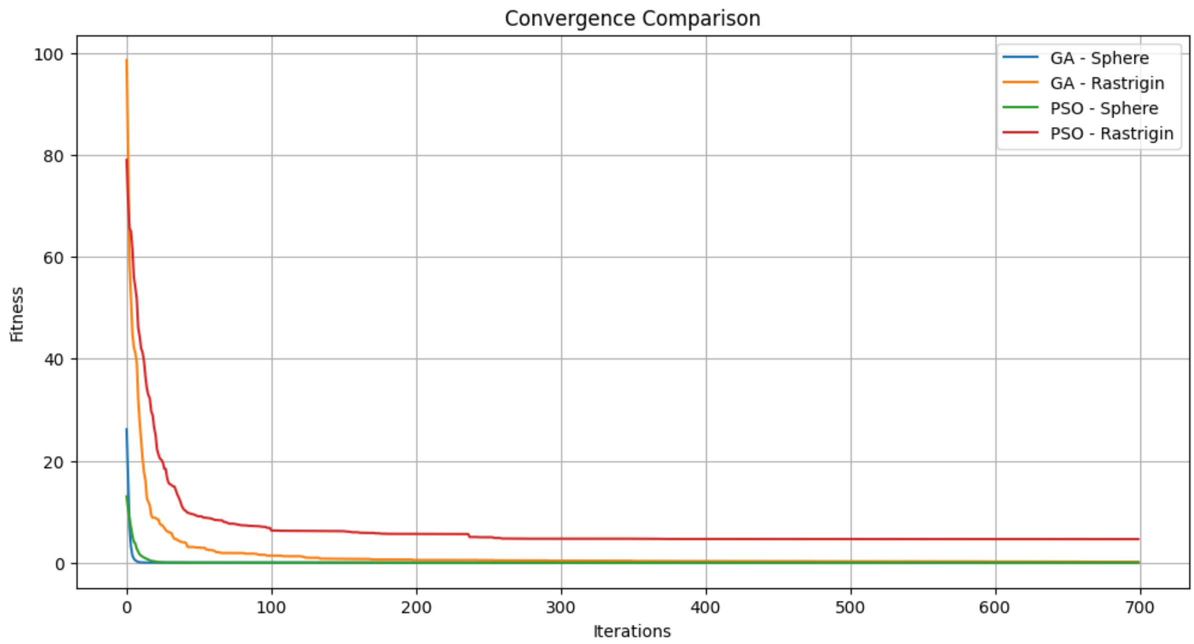
- GA consistently performed well on the Sphere function, with near-perfect results across all dimensions. In contrast, PSO showed perfect accuracy on Sphere but struggled significantly on Rastrigin, especially in higher dimensions.

Table 1: Accuracy Analysis

Algorithm	Dimension	Function	Best Value	Avg Value
GA	10	Sphere	0.000353	0.000667
GA	10	Rastrigin	0.074301	0.159813
PSO	10	Sphere	0.000000	0.000000
PSO	10	Rastrigin	1.085534	4.618379
GA	20	Sphere	0.000018	0.000022
GA	20	Rastrigin	0.003210	0.999571
PSO	20	Sphere	0.000000	0.000000
PSO	20	Rastrigin	1.989918	6.168740
GA	30	Sphere	0.000030	0.000047
GA	30	Rastrigin	0.008020	1.997981
PSO	30	Sphere	0.000000	0.000000
PSO	30	Rastrigin	4.984266	9.213937

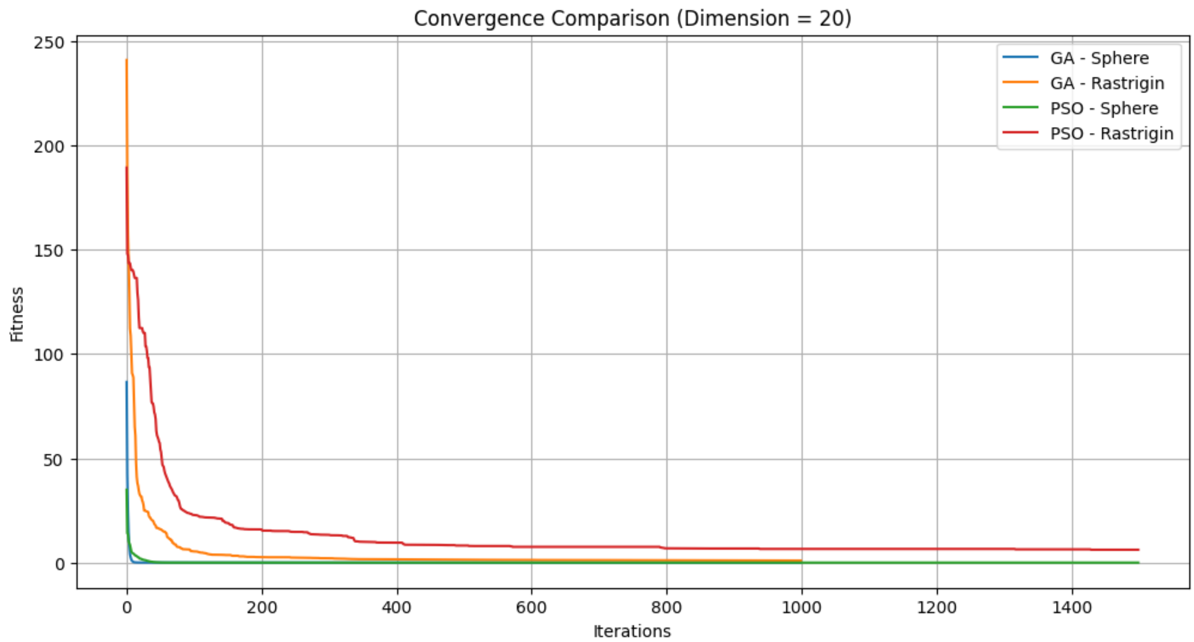
6.2 Convergence

- PSO demonstrated faster convergence in lower dimensions, particularly on the Sphere function. However, it faced difficulties in Rastrigin, where its convergence was slower and less reliable, especially as the number of dimensions increased. GA, while slower to converge, showed more stable and consistent performance, especially on multimodal problems like Rastrigin.



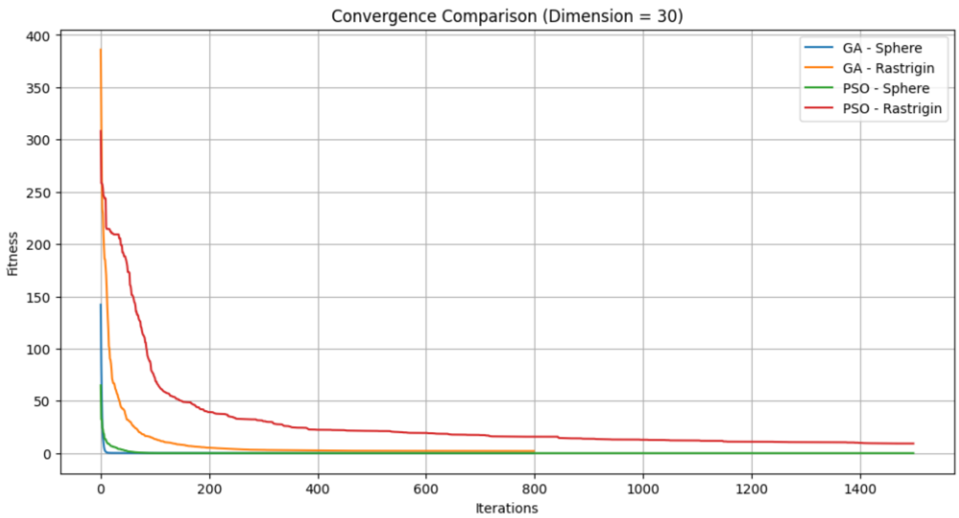
The convergence graph for the 10-dimensional case compares the performance of the Genetic Algorithm (GA) and Particle Swarm Optimization (PSO) on two benchmark functions: Sphere and Rastrigin. On the Sphere function, PSO achieves a perfect result with a best and average fitness of 0.000000, and does so extremely efficiently, averaging only 0.56 seconds. GA also performs well on Sphere, reaching a best fitness of 0.000353 and an average of 0.000667 in 6 seconds. This is reflected in the graph where both algorithms show steep early drops in fitness, with PSO converging faster and lower than GA.

For the Rastrigin function, which is more complex due to its multimodal nature, GA significantly outperforms PSO in solution quality. GA reaches a best fitness of 0.074301 and an average of 0.159813 in about 6.32 seconds, while PSO lags behind with a best of 1.085534 and an average of 4.618379 despite a faster runtime of 1.05 seconds. The graph clearly illustrates this difference, with GA's curve for Rastrigin descending more consistently and flattening at a lower fitness compared to PSO’s, which stabilizes at a higher value. These results confirm that while PSO is highly efficient and ideal for simple problems like the Sphere function, GA demonstrates superior robustness and accuracy on more challenging problems like Rastrigin in lower dimensions.



The convergence graph for the 20-dimensional case provides a detailed comparison of the performance of Genetic Algorithm (GA) and Particle Swarm Optimization (PSO) on the Sphere and Rastrigin functions. In the Sphere function, PSO again demonstrates outstanding efficiency and precision. It converges almost instantly to a fitness of 0.000000, indicating a perfect optimization result, with an average computation time of only 3.24 seconds. GA also performs very well, reaching a best fitness of 0.000018 and an average of 0.000022, though it requires significantly more time—around 34.03 seconds. The convergence curve for PSO on the Sphere function drops steeply and stabilizes early, while GA takes more iterations to approach its optimal value.

For the Rastrigin function, which is complex and multimodal, the distinction in algorithm performance becomes clearer. GA achieves a best fitness of 0.003210 and an average of 0.999571, showing strong convergence and effective exploration capabilities. PSO, while faster (6.07 seconds on average), converges to a higher best fitness of 1.989918 and an average of 6.168740, indicating it is less capable of escaping local minima in the Rastrigin landscape. The convergence curves reflect this, with GA's curve for Rastrigin steadily decreasing and approaching lower fitness levels, while PSO's curve levels off earlier at higher values. These results suggest that while PSO is highly effective and time-efficient on simple, unimodal problems like the Sphere function, GA is more reliable for solving more challenging, multimodal problems like Rastrigin in 20 dimensions.



The convergence graph illustrates the performance of Genetic Algorithm (GA) and Particle Swarm Optimization (PSO) on two benchmark functions—Sphere and Rastrigin—in a 30-dimensional search space. For the Sphere function, both GA and PSO show rapid decreases in fitness, indicating effective convergence. However, PSO significantly outperforms GA by converging faster and achieving a best and average fitness of exactly 0.000000 in just under 100 iterations. In contrast, GA reaches a best fitness of 0.000030 and an average of 0.000047, but takes longer to approach these values, stabilizing only after several hundred iterations. This demonstrates PSO's superior performance in optimizing smooth, unimodal functions like Sphere, with the added advantage of much lower computational time (1.96 seconds for PSO versus 12.67 seconds for GA).

On the Rastrigin function, which is more complex and multimodal, the convergence behavior changes. GA begins with a high fitness value and gradually improves, eventually achieving a best value of 0.008020 and an average of 1.997981. PSO, although initially dropping quickly, struggles with the function's many local minima and converges to a higher best fitness of 4.984266 and an average of 9.213937. This suggests that GA is more robust in handling the rugged landscape of Rastrigin in high dimensions, even though it takes longer (13.82 seconds vs. 3.61 seconds for PSO). The plot and data collectively highlight that PSO is highly efficient and accurate for simpler, convex functions, while GA is more effective for complex, multimodal problems where exploration is crucial.

6.3 Computational Cost

- PSO exhibited clear advantages in terms of computational cost, achieving much faster runtimes compared to GA across all dimensions. While GA was more resource-intensive, it achieved better results on multimodal functions like Rastrigin, justifying the increased computational cost.

Table 2: Computational Cost (Time)

Algorithm	**Dimension**	**Function**	**Average Time (Seconds)**
GA	10	Sphere	6.00
GA	10	Rastrigin	6.32
PSO	10	Sphere	0.56
PSO	10	Rastrigin	1.05
GA	20	Sphere	34.03
GA	20	Rastrigin	33.61
PSO	20	Sphere	3.24
PSO	20	Rastrigin	6.07
GA	30	Sphere	12.67
GA	30	Rastrigin	13.82
PSO	30	Sphere	1.96
PSO	30	Rastrigin	3.61

7. Comparison and Analysis:

7.1 Performance Comparison Between GA and PSO:

The Genetic Algorithm (GA) and Particle Swarm Optimization (PSO) were evaluated on four benchmark functions (Sphere and Rastrigin) across dimensions of 10, 20, and 30. The key performance indicators analyzed include accuracy, convergence rate, and computational cost.

Accuracy:

Sphere Function:

PSO performed exceptionally well on the Sphere function, achieving a best value of 0.000000 in all dimensions (10D, 20D, and 30D), which indicates that PSO can easily find the global optimum in the smooth, unimodal landscape of the Sphere function.

GA showed good performance as well, with values very close to zero (0.000353, 0.000018, 0.000030) across the dimensions, but it consistently performed slightly worse than PSO in terms of both best and average values.

Rastrigin Function:

PSO, while effective on the Sphere function, faced more challenges on the Rastrigin function, particularly as the dimensionality increased. The best values were significantly higher than those of GA, especially in higher dimensions (1.085534 in 10D, 1.989918 in 20D, and 4.984266 in 30D).

GA, on the other hand, was more consistent across dimensions, with the best values being lower than those of PSO, particularly in 20D and 30D (e.g., 0.003210 in 20D and 0.008020 in 30D). GA was able to explore the multimodal Rastrigin landscape more effectively, potentially due to its exploration-exploitation balance (through mutation and crossover).

Computational Cost (Time):

PSO demonstrated superior computational efficiency compared to GA, particularly on the Sphere function, where the average time per run was significantly lower (e.g., 0.56 seconds in

10D and 1.96 seconds in 30D). This is expected since PSO relies on simpler updates for its particle positions and velocities, making it computationally less expensive.

On the Rastrigin function, PSO still had a lower computational cost (e.g., 1.05 seconds in 10D and 3.61 seconds in 30D), but its performance in terms of accuracy was suboptimal compared to GA.

GA consistently took longer for each dimension, especially on Rastrigin (e.g., 6.32 seconds in 10D and 13.82 seconds in 30D). This is likely because of the more complex operations in GA, such as crossover and mutation, which involve evaluating and updating a population of individuals at each generation.

7.2 Discussion on Performance Differences:

PSO on the Sphere Function:

The PSO outperforms GA in terms of both accuracy and computational cost on the Sphere function due to the function's smooth, unimodal landscape. The simplicity of the Sphere function allows PSO to quickly converge to the global optimum with minimal computational effort. Since the landscape does not present many local minima, PSO's particle updates effectively lead to the optimum in a small number of iterations.

The low computational cost of PSO in this case is a result of its straightforward velocity and position update equations, which do not require expensive operations like crossover and mutation as in GA.

GA on the Rastrigin Function:

The Rastrigin function, with its many local minima, presents a challenge for both algorithms. However, GA performed significantly better than PSO in 20D and 30D in terms of accuracy (lower best values). The GA's crossover and mutation operations allow it to explore the search space more thoroughly, helping it escape local minima and find more accurate solutions.

GA's ability to maintain a diverse population through genetic operations likely contributed to its superior performance on the Rastrigin function, as it could better explore the multimodal landscape compared to PSO, which can struggle with getting stuck in local minima due to its particle-based search.

PSO's Struggles on Rastrigin:

PSO's performance on Rastrigin was subpar across all dimensions, particularly when compared to GA. PSO may struggle on such functions due to its reliance on swarm-based exploration, which, in the case of Rastrigin, does not offer as much diversity in its search as GA does. This can lead to premature convergence, especially in higher-dimensional spaces where the local minima are more numerous and challenging to escape from.

7.3 Convergence Rate:

PSO showed faster convergence on Sphere across all dimensions (e.g., 10D: 0.56 seconds, 20D: 3.24 seconds, 30D: 1.96 seconds), as it could quickly find the global minimum with fewer iterations. The algorithm's velocity-based updates help it zero in on the optimum without requiring a large number of generations.

On Rastrigin, PSO's convergence rate was slower compared to GA. The algorithm took longer to approach a solution, and it settled at higher values due to its tendency to converge prematurely to local minima, particularly as the dimensionality increased.

GA demonstrated a steadier convergence, particularly on Rastrigin. It took longer to converge but was more reliable, suggesting that GA is better suited for multimodal functions like Rastrigin, where a broader exploration of the solution space is needed.

7.4 Overall analysis and discussion

PSO excels on simple, unimodal problems like Sphere, where it can quickly find the global minimum with minimal computational cost. However, its performance drops on more complex, multimodal problems like Rastrigin, where it struggles to avoid local minima.

GA, while computationally more expensive, performs better on multimodal functions like Rastrigin, where its ability to maintain diversity through population-based search operations (crossover and mutation) allows it to escape local optima and explore the search space more effectively.

PSO is best suited for simpler problems (e.g., Sphere), while GA is better equipped for more complex, multimodal problems (e.g., Rastrigin). Thus, the choice of optimization algorithm should depend on the characteristics of the function being optimized.

Table: Comparison of GA and PSO on Benchmark Functions

Criterion	GA (Genetic Algorithm)	PSO (Particle Swarm Optimization)
Sphere Function	- Very close to optimal- Slightly slower convergence	- Perfect accuracy (0.000000)- Fastest convergence
Rastrigin Function	- More accurate in all dimensions- Handles local minima better	- Struggles with local minima- Lower accuracy in higher dimensions
Convergence Speed	- Slower, especially in higher dimensions	- Very fast, especially on unimodal functions
Exploration Ability	- High (due to mutation and crossover)	- Moderate, may converge prematurely
Computational Cost	- Higher runtime due to complex operations	- Lower runtime, more efficient
Best Use Case	- Complex, multimodal functions (e.g., Rastrigin)	- Simple, unimodal functions (e.g., Sphere)
Scalability	- Handles high dimensions with accuracy (but slower)	- Fast in low to mid dimensions- Accuracy drops in complex cases
Strengths	- Robust on hard problems- Good exploration	- Speed- Efficient on smooth functions
Weaknesses	- Computationally expensive- Slower convergence	- Prone to local optima- Lower accuracy on complex problems

8.Conclusion

In this study, we conducted a detailed comparison between Genetic Algorithm (GA) and Particle Swarm Optimization (PSO) by testing them on two widely used benchmark functions: Sphere and Rastrigin, across three different dimensions—10D, 20D, and 30D. Our evaluation focused on three key criteria: accuracy, convergence speed, and computational cost. PSO performed exceptionally well on the Sphere function, which is a simple and smooth unimodal optimization problem. Its particle-based search mechanism allowed it to converge quickly and precisely to the global optimum in all tested dimensions. The best values obtained by PSO were consistently 0.000000, and it maintained low computational times—only 0.56 seconds in 10D and scaling efficiently up to 1.96 seconds in 30D. This performance is attributed to PSO's effective update rules for particle velocity and position, which are highly suitable for continuous and smooth landscapes without local minima.

On the other hand, GA outperformed PSO on the more challenging Rastrigin function, which is multimodal and contains many local minima. GA achieved significantly better accuracy, with best values such as 0.003210 in 20D and 0.008020 in 30D, compared to PSO's higher errors in the same dimensions. GA's advantage lies in its evolutionary operators—crossover and mutation—which enhance population diversity and help the algorithm explore the search space more broadly. This makes it more capable of escaping local minima and finding better global solutions, though this thorough exploration comes with increased computational time. For example, GA took 13.82 seconds on average in 30D Rastrigin, whereas PSO completed the same task in just 6.07 seconds.

In terms of convergence and cost, PSO was generally faster across both functions, while GA showed steadier convergence in complex scenarios but at a higher computational expense. This highlights a trade-off: PSO is suitable for problems where fast and accurate results are needed in a smooth, unimodal space, whereas GA is preferable for more complex, multimodal problems where deeper search and robustness are required. Based on these insights, we recommend using PSO for simpler problems like Sphere due to its speed and efficiency, and GA for complex ones like Rastrigin, where accuracy and robustness are more important.

Finally, this study also suggests potential in hybrid optimization approaches. By combining the strengths of PSO's quick convergence with GA's robust exploration capabilities, hybrid algorithms could achieve better performance in both simple and complex problem spaces. For instance, PSO could be used to quickly locate promising regions in the search space, followed by GA to refine the solution and avoid premature convergence. This direction offers exciting possibilities for future research in optimization.

9. References

1. Holland, J.H. (1975). Adaptation in Natural and Artificial Systems. University of Michigan Press.
2. Kennedy, J., & Eberhart, R. (1995). Particle Swarm Optimization. Proceedings of the IEEE International Conference on Neural Networks.
3. Abbas, B.K., et al. (2025). Optimizing Benchmark Functions using PSO. Al-Salam Journal.
4. Sadhu, T., et al. (2023). A Comparative Study of Metaheuristics. Decision Making: Applications.
5. Liang, J.J., et al. (2006). Comprehensive Learning Particle Swarm Optimizer for Global Optimization of Multimodal Functions. IEEE Transactions on Evolutionary Computation.
6. Tantawy, I.A., et al. (2022). Metaheuristics for Optimization Problems: A Review and Comparative Study. Computational Intelligence and Neuroscience.
7. Krohling, R.A., & Coelho, L.S. (2006). Gaussian and Cauchy Mutation in Particle Swarm Optimization. Proceedings of the IEEE Congress on Evolutionary Computation (CEC).
8. Haznedar, B., et al. (2020). Application of Simulated Annealing to Structural Optimization Problems. NTI Pure and Applied Sciences.